LNCS 6218

Edwin R. Hancock Richard C. Wilson
Terry Windeatt Ilkay Ulusoy
Francisco Escolano (Eds.)

Structural, Syntactic, and Statistical Pattern Recognition

Joint IAPR International Workshop, SSPR & SPR 2010
Cesme, Izmir, Turkey, August 2010
Proceedings

S+SSPR 2010

IAPR

Springer

Edwin R. Hancock Richard C. Wilson
Terry Windeatt Ilkay Ulusoy
Francisco Escolano (Eds.)

Structural, Syntactic, and Statistical Pattern Recognition

Joint IAPR International Workshop, SSPR & SPR 2010
Cesme, Izmir, Turkey, August 18-20, 2010
Proceedings

Springer

Volume Editors

Edwin R. Hancock
University of York
York, United Kingdom
E-mail: erh@cs.york.ac.uk

Richard C. Wilson
University of York
York, United Kingdom
E-mail: wilson@cs.york.ac.uk

Terry Windeatt
University of Surrey
Guildford, Surrey, United Kingdom
E-mail: t.windeatt@surrey.ac.uk

Ilkay Ulusoy
Middle East Technical University
Ankara, Turkey
E-mail: ilkay@metu.edu.tr

Francisco Escolano
University of Alicante
Alicante, Spain
E-mail: sco@dccia.ua.es

Library of Congress Control Number: 2010931854

CR Subject Classification (1998): I.2, H.3, H.4, I.5, I.4, H.2.8

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

ISSN	0302-9743
ISBN-10	3-642-14979-0 Springer Berlin Heidelberg New York
ISBN-13	978-3-642-14979-5 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

This volume in the Springer *Lecture Notes in Computer Science* (LNCS) series contains the papers presented at the S+SSPR 2010 Workshops, which was the seventh occasion that SPR and SSPR workshops have been held jointly. S+SSPR 2010 was organized by TC1 and TC2, Technical Committees of the International Association for Pattern Recognition (IAPR), and held in Cesme, Izmir, which is a seaside resort on the Aegean coast of Turkey. The conference took place during August 18–20, 2010, only a few days before the 20th International Conference on Pattern Recognition (ICPR) which was held in Istanbul. The aim of the series of workshops is to create an international forum for the presentation of the latest results and exchange of ideas between researchers in the fields of statistical and structural pattern recognition.

SPR 2010 and SSPR 2010 received a total of 99 paper submissions from many different countries around the world, giving it a truly international perspective, as has been the case for previous S+SSPR workshops. This volume contains 70 accepted papers, 39 for oral and 31 for poster presentation. In addition to parallel oral sessions for SPR and SSPR, there were two joint oral sessions of interest to both SPR and SSPR communities. Furthermore, to enhance the workshop experience, there were two joint panel sessions on "Structural Learning" and "Clustering," in which short author presentations were followed by discussion. Another innovation this year was the filming of the proceedings by Videolectures. The workshop program was enriched by invited talks from four prominent speakers: Narendra Ahuja (University of Illinois at Urbana-Champaign), Ernesto Estrada (University of Strathclyde), Fatih Porikli (Mitsubishi Electric Research Laboratories) and Luc Devroye (McGill University), who was winner of the 2010 Pierre Devijver award.

S+SSPR 2010 was sponsored by the IAPR and PASCAL2. We gratefully acknowledge generous financial support from the PASCAL2 network to cover the costs of filming the proceedings of the joint workshop by Videolectures, and thank IAPR for their support. We would like to take this opportunity to express our gratitude to all those who helped to organize S+SSPR 2010. First of all, thanks are due to the members of the SPR and SSPR Scientific Committees and additional reviewers, who selected the best papers from a large number of submissions to create excellent technical content. Special thanks are due to the members of the Organizing Committee for their efforts and to our host Ilkay Ulusoy for running the event smoothly. We would also like to thank Miguel-Angelo Lozano who set up the website, and Lin Han and Weiping Xu for their help in paper management. We also appreciate the help of the editorial staff at Springer in producing this book, and for supporting the event through publication in the LNCS series. Finally, we thank all the authors and the invited speakers for helping to make this event a success, and for producing a high-quality publication to document the event.

August 2010

Edwin Hancock
Richard Wilson
Terry Windeatt
Ilkay Ulusoy
Francisco Escolano

SPR and SSPR 2010

General Chair

Edwin R. Hancock

Dept. of Computer Science
University of York, UK
erh@cs.york.ac.uk

General Co-chair

Ilkay Ulusoy

Electrical and Electronics Engineering Department
Middle East Technical University, Ankara, Turkey
ilkay@metu.edu.tr

SSPR Programme Chair

Richard C. Wilson

Dept. of Computer Science
University of York, UK
wilson@cs.york.ac.uk

SPR Programme Chair

Terry Windeatt

Dept. of Electrical and Electronic Engineering
University of Surrey, UK
T.Windeatt@surrey.ac.uk

Publicity Chair

Francisco Escolano

Dept. of Computer Science and AI
University of Alicante, Spain
sco@dccia.ua.es

Program Committee

Andrea Torsello (Italy)
Bai Xiao (UK)
Antonio Robles-Kelly (Australia)
Marcello Pelillo (Italy)
Anand Rangarajan (USA)
Gavin Brown (UK)
Oleg Okun (Sweden)
Giorgio Valentini (Italy)
Marco Loog (The Netherlands)
Michael Haindl (Czech Republic)
Larry Hall (USA)
Nikunj Oza (USA)
Konstantinos Sirlantzis (UK)
Gady Agam (USA)
Mayer Aladjem (Israel)
Juan Andrade Cetto (Spain)
Fransesc J. Ferri (Spain)
Georgy Gimelfarb (New Zealand)
Colin de La Higuera (France)
Tin Kam Ho (USA)
Jose Manuel Inesta (Spain)
Francois Jacquenet (France)
Xiaoyi Jiang (Germany)
Jean-Michel Jolion (France)
Mineichi Kudo (Japan)
Walter G. Kropatsch (Austria)
Ventzeslav Valev (Bulgaria)
Elzbieta Pekalska (UK)
Philip Jackson (UK)
Chih-Jen Lin (Taiwan)
John Oommen (Canada)
Sarunas Raudys (Lithuania)
Carlo Sansone (Italy)
Francesco Tortorella (Italy)
Changshui Zhang (China)
Ana Fred (Portugal)
Venu Govindaraju (USA)
Adam Krzyzak (Canada)
Longin J. Latecki (USA)
Punam Kumar Saha (USA)
Zhi Hua Zhou (China)
Hirobumi Nishida (Japan)
Alberto Sanfeliu (Spain)
Luc Brun (France)
Horst Bunke (Sweden)
Sudeep Sarkar (USA)
Jairo Rocha (Spain)
Francesc Serratosa (Spain)
Sargur Srihari (USA)
Mario Vento (Italy)
Sergios Theodoridis (Greece)
Terry Caelli (Australia)
David Windridge (UK)
James Kwok (Hong Kong)
Robert Duin (The Netherlands)
Tibério Caetano (Australia)
Wenwu Wang (UK)
Adrien Bartoli (France)
David Tax (The Netherlands)

Sponsoring Institutions

IAPR

PASCAL2
Pattern Analysis, Statistical Modelling and
Computational Learning

Table of Contents

Invited Talks

Structural Descriptions

Machine Learning

Structural Learning

Poster Session

Geometric Methods

Structural Methods for Vision

Clustering

Poster Session

Dissimilarity-Based Methods

Language

Multiple Classifiers

Graphs

Statistical Pattern Recognition

Structural Methods for OCR

From Region Based Image Representation to Object Discovery and Recognition

Narendra Ahuja[1] and Sinisa Todorovic[2]

[1] Department of Electrical and Computer Engineering,
Coordinated Science Lab, and Beckman Institute,
University of Illinois Urbana-Champaign

[2] School of Electrical Engineering and Computer Science,
Oregon State University

Abstract. This paper presents an overview of the work we have done over the last several years on object recognition in images from region-based image representation. The overview focuses on the following related problems: (1) discovery of a single 2D object category frequently occurring in a given image set; (2) learning a model of the discovered category in terms of its photometric, geometric, and structural properties; and (3) detection and segmentation of objects from the category in new images. Images in the given set are segmented, and then each image is represented by a region graph that captures hierarchy and neighbor relations among image regions. The region graphs are matched to extract the maximally matching subgraphs, which are interpreted as instances of the discovered category. A graph-union of the matching subgraphs is taken as a model of the category. Matching the category model to the region graph of a new image yields joint object detection and segmentation. The paper argues that using a hierarchy of image regions and their neighbor relations offers a number of advantages in solving (1)-(3), over the more commonly used point and edge features. Experimental results, also reviewed in this paper, support the above claims. Details of our methods as well of comparisons with other methods are omitted here, and can be found in the indicated references.

1 Introduction

This paper presents an overview of the region based approach to object recognition and related problems that we have developed over the last several years, and briefly explains its advantages over the more commonly used methods based on point and edge features (e.g., [1, 20, 21, 32, 39, 52, 59, 65]). We briefly describe the major components of our work; details can be found in [6, 54–56].

As a way of addressing recognition-related issues, we consider the following problem. Suppose we are given a set of arbitrary, unlabeled images that contains frequent occurrences of 2D objects from an unknown category. Whether, and where, any objects from the category occur in a specific image from the set is unknown. We are interested in extracting instances of the category from the image set, and obtaining a compact category model in terms of photometric, and geometric and other structural properties. A model derived from such training can then be used to determine whether a new test

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 1–19, 2010.

image contains objects from the learned category, and when it does, to segment all occurrences of the object.

This problem brings together most recognition related issues of interest here, and serves well to highlight the strengths and shortcomings of different approaches. Our region based formulation of this problem, originally presented in [6, 54–56], offers a general framework, subsumes most existing region-based methods, and achieves best performance on challenging benchmark datasets, including Caltech-101 and Caltech-256 [20], and Weizmann Horses [9]. We have shown that our approach:

1. Facilitates access to important object properties that are frequently used as recognition cues, including
   (a) Photometric (e.g., color, brightness),
   (b) Geometric (e.g., size, shape), and
   (c) Structural properties (e.g., layout and recursive embedding of object parts), and
2. Allows simultaneous detection and segmentation, of the target objects and their parts;
3. Simplifies object representation, e.g., for use as statistical models for object classification;
4. Allows efficient and robust learning and inference of object models; and
5. Enables object modeling under various degrees of supervision, including no supervision.

In this paper, we review the part of our work related to objects belonging to a single category [6, 54–56]. Our approach therein consists of four major steps. Given an arbitrary image set, in step 1, each image is segmented using a multiscale segmentation algorithm, and then represented by a region graph capturing the hierarchical and neighbor relations among image regions. Nodes of this graph correspond to regions, ascendant-descendant edges capture their recursive embedding, and lateral edges represent neighbor relations with sibling regions, i.e., those other regions that are embedded within the same parent region. The root of the graph represents the entire image. Step 2 discovers frequent occurrences of an object category in the images by searching for their similar subimages. This is done by matching the corresponding region graphs, and finding their common subgraphs. The set of maximally matching subgraphs is interpreted as occurrences of the discovered object category. In step 3, the matching subgraphs are fused into a single graph-union, which is taken to constitute the canonical model of the discovered object. The graph-union is defined as the smallest graph which contains every subgraph extracted in step 2. In step 4, a newly encountered image is also represented by the region graph that captures the hierarchical and neighbor relations among the image regions. This region graph is then matched with the graph-union model learned in step 3 to simultaneously detect and segment all occurrences of the category in the new image. This matching also identifies object parts along with their containment and neighbor relationships present, which can be used as an explanation of why each object is recognized.

We have also investigated the following other closely related recognition problems, the work on which we will not review in this paper. In [5], we presented a region-based method for extracting a taxonomy of categories from an arbitrary image set. The taxonomy captures hierarchical relations between the categories, such that layouts of frequently co-occurring categories (e.g., head, body, legs, and tail) define more complex, parent categories (e.g., horse). The taxonomy also encodes sharing of categories among different ascendant categories. In the rest of this paper, by “hierarchy” we will refer to both region embedding and their neighbor relations, or layout. As demonstrated in [5], the above hierarchical region-based image representation improves the efficiency of search for shared categories; the available inter-category taxonomy yields sublinear complexity of recognizing all categories that may be present in the image set. Also, in [55], we showed that a hierarchy of regions helps capturing contextual properties of an object (e.g., co-occurrence statistics, and layout of other objects in the vicinity). This is used for estimating the significance of detecting a category in pointing to the presence of other, co-occurring categories in the image. Finally, in [4, 57], we addressed two related problems, that of texture segmentation, and detecting and segmenting the texture elements, called texels. An image texture can be characterized by statistical variations of the photometric, geometric, and structural properties of texels, and relative orientations and displacements of the texels. Since regions facilitate direct capturing these texel properties, our region-based approach outperforms existing methods on benchmark datasets.

The remainder of this paper is organized as follows. Sec. 2 briefly reviews different image features frequently used for recognition. Extraction of a hierarchy of regions from an image is presented in Sec. 3. Sec. 4.1 explains how to discover frequent occurrences of an object category by matching the region hierarchies of a given set of images. Fusing the matching subgraphs into a graph-union, which constitutes the object model, is presented in Sec. 4.2. Finally, Sec. 5 presents some of our empirical results that demonstrate the advantages of using hierarchical region-based image representations for single-category discovery, modeling, and recognition.

2 Regions as Image Features

Recent work typically uses point-based features (e.g., corners, textured patches) and edges (e.g., Canny, Berkeley’s edge map) to represent images [16, 35–37, 48]. Interest points and edges have been shown to exhibit invariance to relatively small affine transforms of target objects across the images [35, 37, 48]. However, there are a number of unsatisfying aspects associated with point features and edges. They are usually defined only in terms of local, gray-level discontinuities (e.g., gradients of brightness), whereas target object occurrences in the image occupy regions. Therefore, the inherent locality of points and edges is dimensionally mismatched with the full 2D spatial extent of objects in the image. As a direct consequence, point-based object detection requires the use of scanning windows of pre-specified size and shape, and often result in multiple, overlapping, candidate detections that need to be resolved in a postprocessing step (e.g., non-maxima suppression). This postprocessing is usually based on heuristic assumptions about the numbers, sizes, and shapes of objects present. Since the final result of this is identification of the points associated with detected objects, it leads to only approximate object localization, not exact object segmentation . To obtain object segmentation, usually the probabilistic map is thresholded which provides likely object locations. This suffers from errors because both locations of local features and the threshold values depend on the particular scene and imaging conditions.

A number of approaches, including our previous work, use image regions as features [2, 6, 8, 11, 18, 27, 28, 31, 43, 55, 56, 64, 67–69]. These methods argue that regions are in general richer descriptors, more discriminative, and more noise-tolerant than interest points and edges. Regions are dimensionally matched with object occurrences in the image. Therefore, regions make various constraints, frequently used in object recognition—such as those dealing with continuation, smoothness, containment, and adjacency—implicit and easier to incorporate than points and edges. Region boundaries coincide with the boundaries of objects and their subparts. This allows simultaneous object detection and segmentation. Since there are fewer regions than local features, using regions often leads to great computational savings, and better performance because, e.g., the number of outliers is significantly reduced.

As always, it is worth noting that the impact of any shortcomings of an image segmentation algorithm should not be confused with the weaknesses of region based representation. For example, oversimplifying assumptions made by some segmentation algorithms about shape, curvature, size, gray-level contrast, and topological context of regions to be expected in an image [24, 38] may lead to segmentation errors of specific types. The same holds for algorithms that implement scale as input parameter which controls the degree of image blurring and subsampling for segmentation [10, 34], or pre-select the number of regions as input parameter [49]. In addition, most segmentation algorithms also use an oversimplified model of photometric profiles of image regions, as being homogeneous and surrounded by step discontinuities, instead of the more realistic ramp (non-step) discontinuities. Therefore, many regions in real images with small intensity gradients do not get segmented, thus adversely affecting object recognition. These limitations of specific segmentation algorithms aside, the use of regions as primitives well serves the objectives of object recognition.

To obtain good segmentation results, we use a multiscale segmentation algorithm presented in [3, 7, 53]. It partitions an image into homogeneous regions of a priori unknown shape, size, gray-level contrast, and topological context. A region is considered to be homogeneous if variations in intensity within the region are smaller than intensity change across its boundary, regardless of its absolute degree of variability. Image segmentation is performed at a range of homogeneity values, i.e., intensity contrasts. As the intensity contrast increases, regions with smaller contrasts strictly merge. A sweep of the contrast values thus results in the extraction of all the segments present in the image.

3 Segmentation Tree and Region Descriptors

After segmenting an image, the resulting regions and their spatial and structural relationships can be used for recognition. A number of approaches do not exploit region relationships, but account for region intrinsic properties, and treats the regions as a bag of visual words [28, 45]. Other methods additionally account for pairwise region relations [27], and the contextual information provided by larger ancestor regions within which smaller regions are embedded [33]. Our work [6, 55, 56], along with several other methods [25], generalizes previous approaches by additionally accounting for the spatial layout and recursive embedding of regions in a segmentation tree.

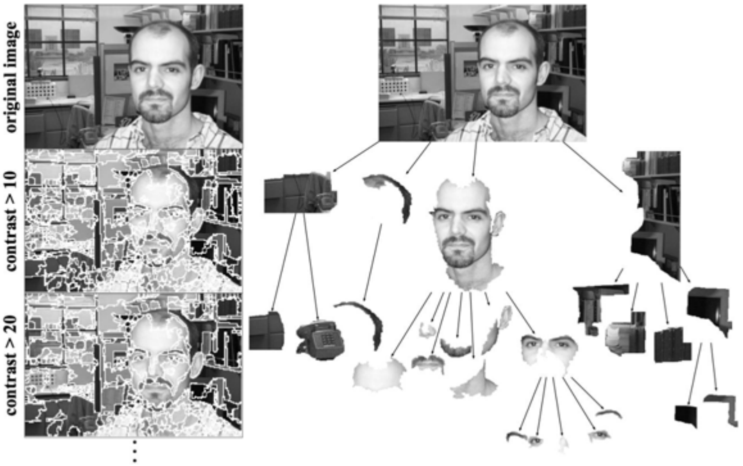


Fig. 1. Segmentation trees of sample Caltech-101 images [20]: (left) segmentations obtained for two sample intensity contrast values from the exhaustive range [1,255]; (right) sample nodes of the corresponding segmentation tree, where the root represents the whole image, nodes closer to the root represent large regions, while their children nodes capture smaller embedded details. The number of nodes (typically 50–100), branching factor (typically 0–10), and the number of levels (typically 7–10) in different parts of the segmentation tree are image dependent, and automatically determined.

In the segmentation tree, the root represents the whole image, nodes closer to the root represent large regions, while their children nodes capture smaller embedded details, as depicted in Fig. 1. The tree in general may not have regular structure (e.g., quad-tree). For example, the multiscale segmentation of [3, 7, 53] gives the number of nodes (typically 50–100), branching factor (typically 0–10), and the number of levels (typically 7–10) that are image dependent in different parts of the tree. Thus, the segmentation tree is a rich image representation that is capable of capturing object properties (a)–(d), mentioned in Sec. 1.

The segmentation tree (ST), however, cannot distinguish among many different ways in which the same set of subregions may be spatially distributed within the parent region. This may give rise to significantly different visual appearances, while the region-embedding properties remain the same. Consequently, STs for many visually distinct objects are identical. The ST can be extended by including the information about 2D spatial adjacency among the regions – while retaining the information about their recursive embedding. This new model augments ST with region adjacency graphs, one for the children of each ST node. A neighbor edge is added between two sibling nodes in ST if the corresponding two regions are neighbors in the image. This transforms ST into a graph, consisting of two distinct sets of edges – one representing the original, parent-child hierarchy, and the other, consisting of lateral links, representing the newly added neighbor relationships (Fig. 2). The neighbor relationships between any nonsibling nodes in CST can be easily retrieved by examining the neighbor relations of their ancestor nodes. To highlight the presence of the complementary, neighbor information modifying the segmentation tree, the new representation is referred to as connected

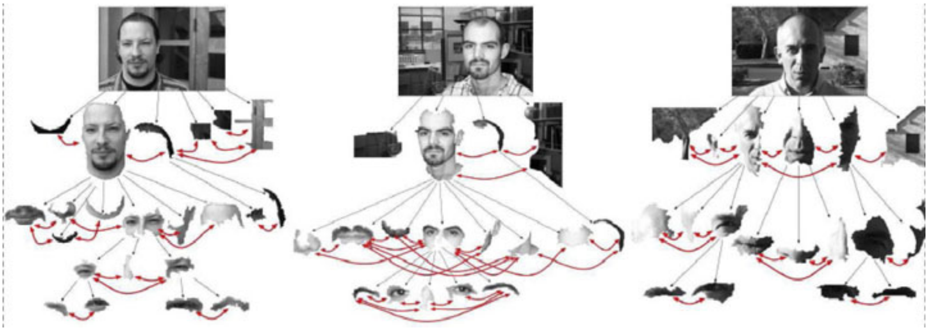


Fig. 2. Example Connected Segmentation Trees (CSTs): Lateral edges (red) that link neighboring image regions are added to the corresponding segmentation trees (black) of the images. CSTs reduce ambiguity about the region layout.

segmentation tree (CST), even though it is strictly a graph. Both nodes and edges of CST have attributes, i.e., they are weighted, where the node (edge) weight is defined in terms of properties of the corresponding region (spatial relationship between regions). Thus, CST generalizes ST to represent images as a hierarchy of region adjacency graphs. As multiscale regions may be viewed as a basic vocabulary of object categories, the CST may be seen as a basis for defining general purpose image syntax, which can serve as an intermediate stage to isolate and simplify inference of image semantics. In the following, we will interchangeably use CST and region hierarchy to denote the same image representation—namely, the hierarchical graph representation that captures recursive embedding of regions, as well as region layout at all levels.

Each node v in the region hierarchy can be characterized by a vector of properties of the corresponding region, denoted as $\boldsymbol{\psi}_v$. In our previous work, we use intrinsic photometric and geometric properties of the region, as well as relative inter-region properties describing the spatial layout of the region and its neighbors. In this way, $\boldsymbol{\psi}_v$ encodes the spatial layout of regions, while the CST structure itself captures their recursive containment. The properties are defined to allow scale and rotation-in-plane recognition invariance. In particular, elements of $\boldsymbol{\psi}_v$ are defined relative to the corresponding properties of v's parent-node u, and thus ultimately relative to the entire image.

Let w, v, and u denote regions forming a child-parent-grandparent triple. Then, the properties of each region v we use are as follows: (1) normalized gray-level contrast g_v, defined as a function of the mean region intensity G, $g_v \triangleq \frac{|G_u - G_v|}{|G_v - G_w|}$; (2) normalized area $a_v \triangleq A_v / A_u$, where A_v and A_u are the areas of v and u; (3) area dispersion AD_v of v over its children $w \in C(v)$, $\mathrm{AD}_v \triangleq \frac{1}{|C(v)|} \sum_{w \in C(v)} (a_w - \overline{a_{C(v)}})^2$, where $\overline{a_{C(v)}}$ is the mean of the normalized areas of v's children; (4) the first central moment μ_v^{11}; (5) squared perimeter over area $\mathrm{PA}_v \triangleq \frac{\text{perimeter}(v)^2}{A_v}$; (6) angle γ_v between the principal axes of v and u; the principal axis of a region is estimated as the eigenvector of matrix $\frac{1}{\mu^{00}} \begin{bmatrix} \mu^{20} & \mu^{11} \\ \mu^{11} & \mu^{02} \end{bmatrix}$ associated with the larger eigenvalue, where the μ's are the standard central moments; (7) normalized displacement $\overrightarrow{\Delta}_v \triangleq \frac{1}{\sqrt{A_u}} \overrightarrow{d}_v$, where $|\overrightarrow{d}_v|$ is the

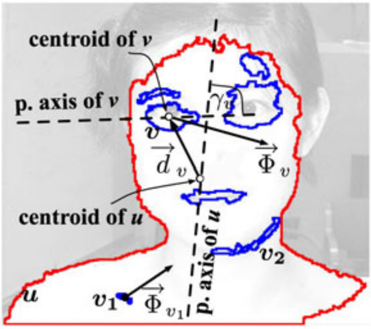


Fig. 3. Fig. 3. Properties of a region associated with the corresponding node in the segmentation tree: Region u (marked red) contains a number of embedded regions $v, v_1, v_2, \ldots$ (marked blue). The principal axes of u and v subtend angle γ_v, the displacement vector $\boldsymbol{d}_v$ connects the centroids of u and v, while the context vector $\boldsymbol{F}_v$ records the general direction in which the siblings $v_1, v_2, \ldots$ of v are spatially distributed.

distance between the centroids of u and v, and $\measuredangle \overrightarrow{d}_v$ is measured relative to the principle axis of parent node u, as illustrated in Fig. 3; $\sqrt{A_u}$ represents an estimate of the diameter of parent region u; and (8) context vector $\overrightarrow{F}_v \triangleq \sum_{s \in S(v)} \frac{A_s}{|\overrightarrow{d}_{vs}|^3} \overrightarrow{d}_{vs}$, where $S(v)$ is the set of v's sibling regions s, and $|\overrightarrow{d}_{vw}|$ is the distance between the centroids of v and s, and $\measuredangle \overrightarrow{d}_{vs}$ is measured relative to the principle axis of their parent node u; as illustrated in Fig. 3, the context vector records the general direction v sees its sibling regions and disallows matching of scrambled layouts of regions at a specific tree level. In summary, the vector of region properties associated with node v is $\boldsymbol{\psi}_v = [g_v, a_v, \mathrm{AD}_v, \mu_v^{11}, \mathrm{PA}_v, \gamma_v, \overrightarrow{\Delta}_v, \overrightarrow{F}_v]^{\mathrm{T}}$. Each element of $\boldsymbol{\psi}_v$ is normalized over all multiscale regions of all training images to take a value in the interval $[0, 1]$. This list of useful region properties, can be easily modified to reflect the needs of different applications.

The aforementioned hierarchical region-based image representation will allow recognition performance with the following desirable invariance characteristics with respect to: (i) Translation, in-plane rotation and object-articulation (changes in relative orientations of object parts): because the segmentation tree itself is invariant to these changes; (ii) Scale: because subtree matching is based on relative properties of nodes, not absolute values; (iii) Occlusion in the training set: because subtrees are registered and stitched together within the tree-union encoding the entire (unoccluded) category structure; (iv) Occlusion in the test set: because subtrees corresponding to visible object parts can still be matched with the model; (v) Small appearance changes (e.g. due to noise): because changed regions may still be the best matches; (vi) Region shape deformations (e.g., due to minor depth rotations of objects): because changes in geometric/topological properties of regions (e.g., splits/mergers) are accounted for during matching; and (vii) Clutter: because clutter regions, being non-category subimages, are not repetitive and therefore frequent.

Any object occurrences in images will correspond to subgraphs within the corresponding CSTs. The goal of learning is to identify these subgraphs and capture their canonical node and node-connectivity properties. The goal of inference is to use this graph model to identify, within the CST of a new image, subgraphs that represent instances of the learned class. In the following two sections, we explain object learning and recognition using the region hierarchy.

4 Learning Object Properties

This section argues that hierarchical region-based representations of images possess two major features—namely, that they: (a) facilitate learning under various degrees of supervision, and (b) relax the requirements for complex object models and classifiers.

4.1 Object Discovery as Graph Matching

To communicate the natural variations of objects to a recognition algorithm, typically, a set of training images has to be manually annotated. Supervision in training may involve the following: manually segmented object instances in training images, bounding boxes placed around the objects, or only object labels associated with the entire images. In case the bounding boxes are available in training, they immediately provide access to similar subgraphs of region hierarchies corresponding to instances of the target object class. If the bounding boxes are not available, the object occurrences can be discovered by matching the region hierarchies of images from the same class, and thus identifying their similar subgraphs. Below, we explain how to match CSTs, and thus obtain a set of their similar subgraphs, which will be used then to learn the object model or classifier.

Two images may have a number of similar regions, which may confuse the matching algorithm. However, if similar regions also have similar nesting and layout properties, then it is very likely that they represent meaningful image parts, e.g., instances of the same object class, which indeed should be matched. Our algorithm achieves robustness by pairing regions whose photometric, geometric, and structural properties match, and the same holds for their neighbors, and these two conditions recursively hold for their embedded subregions. Such region matching can be formalized using the graph matching techniques. In the following, we first briefly review graph-based image matching methods, and then present our approach.

Image matching using graph image representations may be performed by: (a) exploiting spectral properties of the graphs' adjacency matrices [44, 50, 51]; (b) minimizing the graph edit-distance [12, 47, 62]; (c) finding a maximum clique of the association graph [41]; (d) using energy minimization or expectation-maximization of a statistical model [23, 63]. All these formulations can be cast as a quadratic assignment problem, where a linear term in the objective function encodes node compatibility functions, and a quadratic term encodes edge compatibility functions. Therefore, approaches to graph matching mainly focus on: (i) finding suitable definitions of the compatibility functions; and (ii) developing efficient algorithms for approximately solving the quadratic assignment problem (since it is NP-hard), including a suitable reformulation of the quadratic into linear assignment problem. However, most popular approximation algorithms (e.g., relaxation labeling, and loopy belief propagation) critically depend on a good initialization and may be easily trapped in a local minimum, while some (e.g., deterministic annealing schemes) can be used only for graphs with a small number of nodes. Graduated nonconvexity schemes [26], and successive convexification methods [30] have been used to convexify the objective function of graph matching, and thus alleviate these problems. In our work, we use the replicator dynamics algorithm to solve the underlying convex problem, as explained in the sequel.

Let $H = (V, E, \psi, \phi)$ denote the region hierarchy, where $V = \{v\}$ and $E = \{(v, u)\} \subseteq V \times V$ are the sets of nodes and edges, and ψ and ϕ are functions that assign attributes to nodes, $\psi : V{\rightarrow}[0, 1]^d$, and to edges, $\phi : E{\rightarrow}[0, 1]$. Given two shapes, H and H', the goal of the matching algorithm is to find a subgraph isomorphism, $f{:}U{\rightarrow}U'$, where $U{\subseteq}V$ and $U'{\subseteq}V'$, which minimizes the cost, C, defined as

$$C = \min_f \left[\beta \sum_{(v,v')\in f} a_{vv'} + (1-\beta) \sum_{(v,v',u,u')\in f\times f} b_{vv'uu'} \right], \tag{1}$$

where the a's are non-negative costs of matching nodes v and $v' = f(v)$, and the b's are non-negative costs of matching edges $(v, u) \in E$ and $(v', u') \in E'$, and $\beta \in [0, 1]$ weights their relative significance to matching.

To minimize C, we introduce a confidence vector, X, indexed by all node pairs $(v, v'){\in}V{\times}V'$, whose each element $x_{vv'}{\in}[0, 1]$ encodes the confidence that node pair (v, v') should be matched. Matching can then be reformulated as estimating X so that C is minimized. That is, we relax the discrete problem of (1) to obtain the following quadratic program (QP):

$$\begin{array}{ll} \min\limits_X & \left[\beta A^{\mathrm{T}} X + (1-\beta) X^{\mathrm{T}} B X\right], \\ \text{s.t.} & \forall (v, v'){\in}V{\times}V', \quad x_{vv'}{\geq}0, \\ & \forall v'{\in}V', \quad \sum_{v\in V} x_{vv'}{=}1, \\ & \forall v{\in}V, \ \sum_{v'\in V'} x_{vv'}{=}1, \end{array} \tag{2}$$

where A is a vector of costs $a_{vv'}$, and B is a matrix of costs $b_{vv'uu'}$. We define $a_{vv'} = \|\psi(v) - \psi(v')\|_2$. Also, we define $b_{vv'uu'}$ so that matching edges of different types–namely, hierarchical and neighbor edges—is prohibited, and matches between edges of the same type with similar weights are favored in (2): $b_{vv'uu'} = \infty$ if edges (v, u) and (v', u') are not of the same type; and $b_{vv'uu'} = |\phi(v, v') - \phi(u, u')|$ if edges (v, u) and (v', u') are of the same type. Both the a's and b's are normalized to [0,1].

To satisfy the isomorphism constraints of matching, the algorithm matches regions with regions, and separately region relationships with corresponding relationships, while preserving the original node connectivity of H and H'. The constraints in (2) are typically too restrictive, because H and H' may have relatively large structural differences in terms of the number of nodes and their connectivity, even if H and H' represent two objects from the same class. These structural differences may, e.g., arise from different outputs of the segmentation algorithm on images of the same object class but captured under varying illumination. In this case, splitting or merging regions along their shared, low-contrast boundary may occur which affects the structure of H and H'. Therefore, a more general many-to-many matching formulation would be more appropriate for our purposes. The literature reports a number of heuristic approaches to many-to-many matching [19, 42, 58], which however are developed only for weighted graphs, and thus cannot be used for our region hierarchies that have attributes on both nodes and edges. To relax the constraints in (2), we first match H to H', which yields solution X_1. Then, we match H' to H, which yields solution X_2. The final solution, $\tilde{X}$, is estimated as an intersection of non-zero elements of X_1 and X_2. Formally, the constraints in (2) are relaxed as follows: (i) $\forall(v,v') \in V{\times}V', x_{vv'} \geq 0$; and (ii) $\forall v \in V, \sum_{v'\in V'} x_{vv'} = 1$ when matching H to H'; and $\forall v' \in V', \sum_{v\in V} x_{vv'} = 1$ when matching H' to H. Thus, by using an auxiliary matrix $W = \beta\text{diag}(A) + (1-\beta)B$, we reformulate (2) and arrive at the following one-to-many matching problem

$$\begin{array}{ll}\min\limits_{X} & X^{\mathrm{T}}WX,\\ \text{s.t.} & \forall(v,v'){\in}V{\times}V', \quad x_{vv'}{\geq}0,\\ & \forall v'{\in}V', \quad \sum_{v\in V} x_{vv'}{=}1,\end{array} \tag{3}$$

which can be efficiently solved by using the replicator dynamics update rule [40]:

$$X \leftarrow \frac{WX}{X^{\mathrm{T}}WX}. \tag{4}$$

The proof that the optimization of (3) results in the subgraph isomorphism follows from the well-known Motzkin-Strauss theorem, as shown in [40, 41].

Complexity of our matching is $O((|V|{+}|E|)^2)$. Our implementation in C takes about 1min on a 2.8GHz, 2GB RAM PC for two CSTs with approximately 50 nodes.

The matching subgraphs may represent complete object occurrences or their parts (e.g., due to partial occlusion, or changes in illumination, viewpoint, or scale variations across the images). Therefore, the extracted similar subgraphs provide for many observations of entire objects or their parts in the class. This allows robust estimation of the region-based object model. Note that as a result of matching region hierarchies, we immediately have access to correspondences between nodes and edges of all extracted subgraphs. These correspondences can be used to learn a canonical graph of the object class that subsumes all extracted instances, and thus represents the object model.

4.2 Region-Based Object Model

The region-based object model is aimed at capturing how image regions are recursively laid out to comprise an object, and what their geometric and photometric properties are. From a set of given or extracted similar CSTs, as explained in the previous section, our goal is to obtain a compact, canonical model of the target class. In our work we formulate this canonical graph as graph-union.

Graph-unions are well studied graph structures, the detailed treatment of which can be found, for example, in [13–15, 29, 60, 61]. The graph-union $\mathcal{T}$ is the smallest graph, which contains every graph from a given set $\mathbb{D}$. Ideally, $\mathcal{T}$ should be constructed by first finding the maximum common subgraph of $\mathbb{D}$, and then by adding to the common subgraph, and appropriately connecting, the remaining nodes from $\mathbb{D}$. However, finding this maximum common subgraph would entail prohibitive complexity if D is large. Therefore, we resort to a suboptimal sequential approach. In each iteration $\mathcal{T}$ is

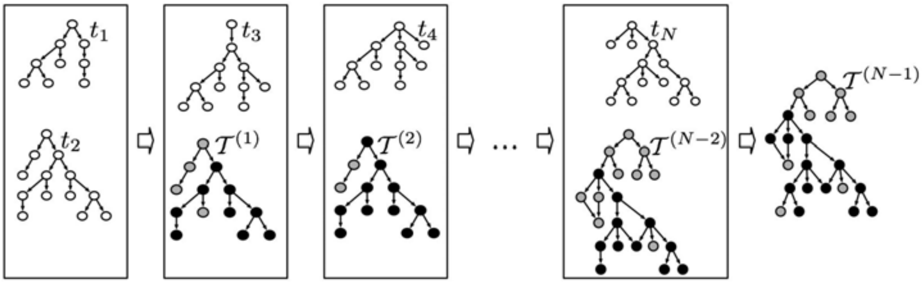


Fig. 4. Construction of graph-union $\mathcal{T}$ from the extracted set of similar CSTs $\mathbb{D}=\{t_1, t_2, \ldots, t_N\}$: In each iteration, a selected CST t from $\mathbb{D}$ is first matched against the current estimate $\mathcal{T}^{(n)}$, which yields their maximum common subgraph τ (marked black). Then the unmatched nodes from t are added and appropriately connected (marked gray), to form $\mathcal{T}^{(n+1)}$. The result is the graph-union.

extended by adding a new CST t from $\mathbb{D}$ until every CST from $\mathbb{D}$ has been added to the graph-union, as illustrated in Fig. 4. As can be seen, the selected t is first matched against the current estimate $\mathcal{T}^{(n)}$, which results in their common subtree τ, and then the unmatched nodes from t are added and appropriately connected to τ in order to form $\mathcal{T}^{(n+1)}$. For matching t and $\mathcal{T}^{(n)}$, we use the same algorithm presented in Sec. 4.1. After adding the unmatched nodes, the result is the graph-union, which preserves the node connectivity from $\mathbb{D}$.

5 Results

Region hierarchies, as our image representations, allow joint object detection, recognition and segmentation. This can be achieved by matching the learned graph-union model, presented in the previous section, with the region hierarchy of a new image. In our approach, the matching subgraphs whose similarity measure is larger than a specified threshold are taken as detected objects. This detection simultaneously delineates object boundaries, due to using regions as basic image features. This section reviews the empirical validation of our approach, presented in [6]. The experiments demonstrate advantages of using region-based image representations and object modeling for recognition versus alternative approaches.

We consider 14 categories from four datasets: 435 faces, 800 motorbikes, 800 airplanes, 526 cars (rear) from Caltech-101 [20]; 328 Weizmann horses [9]; 1554 images queried from LabelMe [46] to contain cars, trees, and buildings together; and 200 images with 715 occurrences of cows, horses, sheep, goats, camels, and deer from UIUC Hoofed Animals dataset [6]. Caltech-101 images contain only a single, prominently featured object from the category, except for images of cars (rear) containing multiple, partially occluded cars appearing at different scales, with low contrast against textured background. The Weizmann dataset contains sideviews of walking/galloping horses of different breeds, colors and textures, with different object articulations in their natural (cluttered) habitat. LabelMe is a more difficult collection of real-world images which contain many other object categories along with the queried ones, captured under different lighting conditions, and at varying scales. The Hoofed Animals dataset presents the mentioned challenges, and has higher complexity as it contains multiple instances of multiple very similar animal categories per image, requiring high inter-category resolvability.

The Caltech-101 and Weizmann categories are learned one category at a time on the training set that consists of M_p randomly selected examples showing the category, and $M_n{\geq}0$ images from the background category in Caltech-101 ($M{=}M_p{+}M_n$). The LabelMe and Hoofed Animals categories are all learned together by randomly selecting M images from the corresponding dataset. To recognize and segment any category occurrences in a test image, the learned category model is matched with CST of the image. The matched subtrees (i.e., detections) whose similarity measure is larger than a threshold are adjudged as detected objects. Results shown in tables and figures are obtained for the threshold that yields equal error rate. We use the following definitions of detection (DE), and segmentation (SE) errors. Let D denote the area that a detection covers in the test image, and G denote the ground-truth object area. Then, DE$\triangleq\frac{D\cap G}{D\cup G}$, and SE$\triangleq\frac{\text{XOR}(D,G)}{D\cup G}$. A detection is a false positive if DE<0.5, otherwise it is a true positive (TP). Recognition is evaluated only on TP's by visual inspection.

5.1 Qualitative Evaluation – Segmentation

Figs. 5–6 demonstrate high accuracy of simultaneous object detection and segmentation in images from LabelMe and Hoofed Animals datasets, using $M{=}50$ training images. Each TP in the figures is correctly recognized. CSTs outperform STs in both object detection and segmentation, especially in cases of partial occlusion (e.g., cars and cows in Fig. 6), and for objects defined rather as a region spatial layout than containment (e.g., spotted cows in Fig. 6). In these cases, modeling of the region adjacency by CSTs proves advantageous. Segmentation is good even in cases when object boundaries are jagged and blurred (e.g., trees in Fig. 5), and when objects from the same category occlude each other, forming a complex region topology with low-intensity contrasts

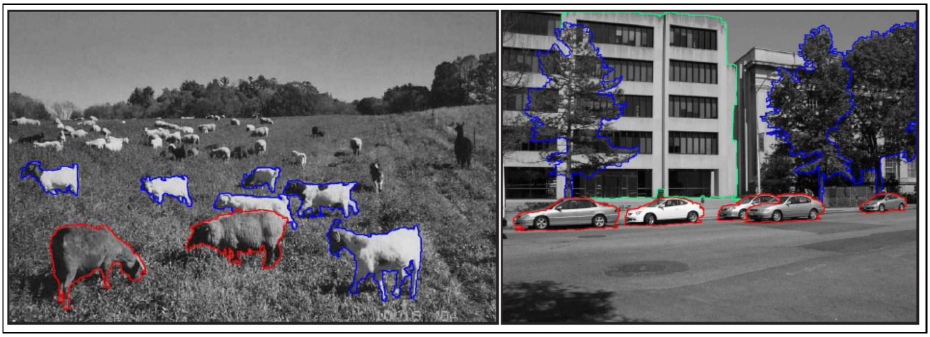
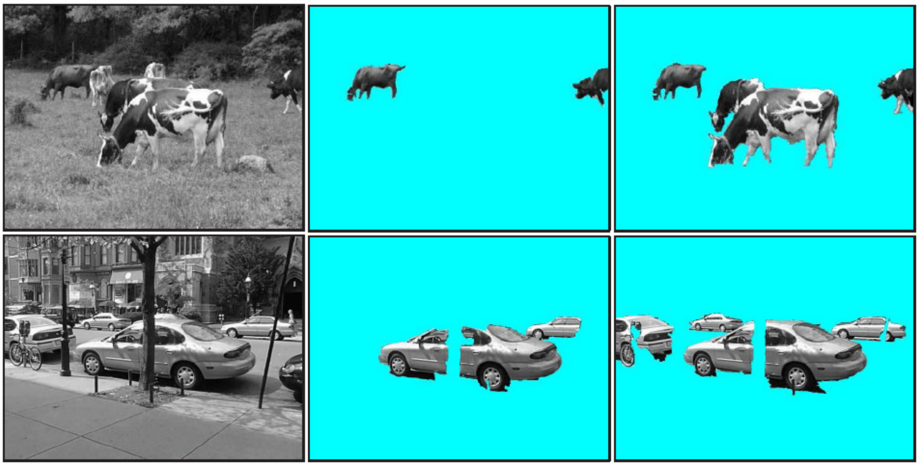


Fig. 5. Samples from Hoofed Animals (left) and LabelMe (right). Segmentation results of CST are overlaid on the original. Different colors denote recognized categories. CST successfully resolves small differences between the categories sheep and goats.



(a) original image (b) STs (c) CSTs

Fig. 6. CSTs outperform STs in both detection and segmentation on samples from Hoofed Animals (top) and LabelMe (bottom). Undetected image parts are masked out.

(e.g., cars in Fig. 5). Objects that are not detected, for the most part, have low intensity contrasts with the surround, and thus do not form category-characteristic subgraphs within CSTs that can be matched with the category model.

5.2 Qualitative Evaluation – Model

Fig. 7 illustrates the model $\mathcal{G}$ obtained for the category horses, learned on six, randomly selected images $\mathbb{D}$ from the Weizmann dataset. Nodes v in $\mathcal{G}$, depicted as rectangles, contain regions from $\mathbb{D}$ that got matched with v during learning. As can be seen, the structure of $\mathcal{G}$ correctly captures the recursive containment and neighbor relations of regions occupied by the horses in $\mathbb{D}$. For example, nodes *head*, *neck*, and *mane* are found to be children of node *head&neck*, and they are all identified as neighbors. Also, it is correct that *head&neck* and *tail* are not neighbors. Similar background regions that co-occur with horses in $\mathbb{D}$ may also be included in the model (e.g., nodes corresponding to *fence*). Typically, the percentage of background nodes out of the total number of model nodes is small (3-5%).

5.3 Quantitative Evaluation

Fig. 8 (left) presents the recall-precision curves (RPC) of detection for the Caltech-101 categories using CSTs and STs. Detection performance in the presence of occlusion is tested by masking out a randomly selected rectangular area in the image, and replacing this area with a patch from the background category of Caltech-101. CST increases the area under the RPC of ST by $6.5 \pm 0.3\%$, and by $3.1 \pm 0.2\%$ in the presence of the occluding patch covering 20% of the image. Invariance to in-plane rotation is tested by randomly rotating test images. Performance on these rotated images is the

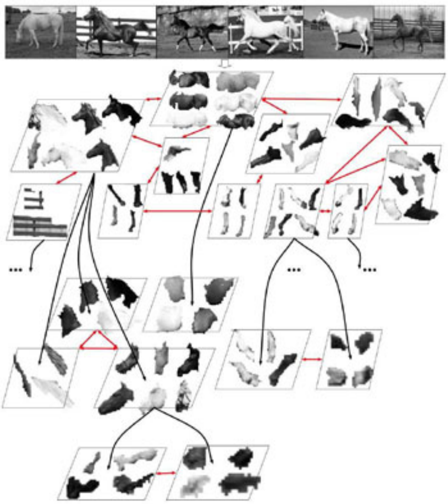


Fig. 7. CST-based model of Weizmann horses learned on the input images shown in the top row

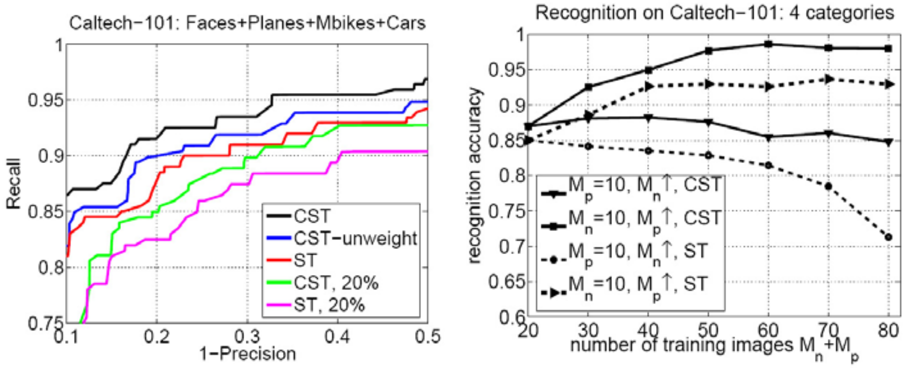


Fig. 8. (left) Detection recall-precision curves: "CST-unweight" means that edges in CST are not weighted. 20% is the size of a rectangular occlusion w.r.t. the image size. M_p=10, M_n=10. ST is the method of [54]. **(right)** Recognition accuracy of CST and ST for the varying ratio of M_p and M_n in the training set.

same as the one presented in Fig. 8. Measuring the strength of neighborliness using the generalized Voronoi diagram improves performance over the case when the weights of links in CST are set to take only values 1 or 0, referred to as CST-unweight. CST increases the area under the RPC of CST-unweight by $2.3 \pm 0.3\%$. Fig. 8 (right) shows recognition accuracy of CST and ST. A small increase in M_n does not downgrade the accuracy. As M_n becomes larger, objects belonging to other categories start appearing more frequently, and thus get learned, making the training set inappropriate. Increasing M_p yields smaller recognition error. CST outperforms ST in recognition, and longer maintains high accuracy with the increase of M_n. In general, the number of nodes in the

Table 1. Detection recall, segmentation and recognition errors (in %) on LabelMe and Weizmann Horses datasets, using the same number of training and test images as in [17, 45, 66]

	LabelMe Trees	LabelMe Buildings	LabelMe Cars	Weizmann Horses
Recall	47.6±6.9	92.6±6.9	67.6±6.9	91.9±5.2
Seg. error	41.6±7.9	34.6±13.4	32.5±8.2	7.2±2.5
Rec. error	19.7±3.8	11.6±2.9	12.9±4.8	7.9±4.1

Table 2. Detection recall, segmentation and recognition errors (in %) on UIUC Hoofed Animals dataset, using the same number of training and test images as in [17, 45, 66]

	Horses	Cows	Deer	Sheep	Goats	Camels
Recall	81.2±10.3	78.4±4.2	88.1±6.9	81.2±5.3	78.2±8.6	89.9±7.2
Seg. error	15.9±5.3	17.1±4.6	11.1±8.4	24.8±7.2	20.1±8.1	11.5±5.1
Rec. error	7.8±4.2	6.5±6.2	7.7±3.4	7.8±4.1	12.2±5.4	3.2±3.9

model quickly reaches saturation as new positive examples are added to the training set, and continues to very slowly increase, in part, due to chance repetitions of background regions.

Table 1 and Table 2 summarize detection recall, and segmentation and recognition errors obtained for the equal error rates on LabelMe, Weizmann, and Hoofed Animals datasets. For Hoofed Animals, CST outperforms ST in detection recall by 7.5%, segmentation by 10.7%, and recognition by 8.6%. For comparison, we obtained SE=6.5% on a relatively simple UIUC (multiscale) car dataset, using the same set-up as in [22], while their result is SE=7.9%. The other hierarchical approaches cited here use non-benchmark datasets, or report a single retrieval result for the entire Caltech-101, beyond the focus of this paper. Non-hierarchical approaches that model objects using image segments obtained at only one pre-selected scale, report the following state-of-the-art results: [45] – $SE{=}47\%$ for buildings, and $SE{=}79\%$ for cars of LabelMe; [66] – SE=7% for Weizmann horses; and [17] – SE=18.2% for Weizmann horses. In comparison with these approaches, Table 1 indicates that the CSTs yield better, or, in only a few cases, very similar performance. Regarding recognition accuracy, Fig. 8 shows that we outperform by 1.8 ± 0.3% the recognition rate of 94.6% of [17] on the four Caltech-101 categories. Other approaches cited here use a different, less demanding recognition evaluation based on classifying either the entire images or bounding boxes around objects.

The results demonstrate that our approach is invariant with respect to: (i) translation, in-plane rotation and object articulation, since CST itself is invariant to these changes; (ii) certain degree of scale changes, since matching is based on relative properties of regions; (iii) occlusion in the training and test sets, since graph-union registers the entire (unoccluded) category structure from partial views of occurrences in the training set, while subgraphs of visible object parts in the CST of a test image can still be matched with the model; (iv) minor depth rotations of objects causing their shape deformations, because structural instability of CSTs (e.g., due to region splits/mergers) is accounted for during matching; and (v) clutter, since clutter regions are not frequent and thus not learned.

6 Conclusions

We have argued in this paper that using multiscale regions as basic image features: (a) Facilitates capturing photometric, geometric, and structural properties of objects; (b) Allows simultaneous object discovery, recognition and segmentation; and (c) Enables efficient and robust learning and inference of region-based object representations. We have reviewed our region-based object recognition framework developed over the last several years. While the framework is capable of extracting a taxonomy of object categories from an arbitrary image set, and segmenting textures into texels, we have focused here on a compact subset of these problems. We have considered the related problems of single category discovery, detection, and segmentation. We have discussed how this set of problems poses many recognition related challenges, which are inadequately addressed by existing methods that use point and edge features. The summary of our experimental results that we have presented here shows that use of regions offers a number of advantages for object recognition over point and edge features.

References

1. Agarwal, S., Awan, A., Roth, D.: Learning to detect objects in images via a sparse, part-based representation. IEEE TPAMI 26(11), 1475–1490 (2004)
2. Ahmadyfard, A.R., Kittler, J.V.: Using relaxation technique for region-based object recognition. Image and Vision Computing 20(11), 769–781 (2002)
3. Ahuja, N.: A transform for multiscale image segmentation by integrated edge and region detection. IEEE TPAMI 18(12), 1211–1235 (1996)
4. Ahuja, N., Todorovic, S.: Extracting texels in 2.1D natural textures. In: ICCV (2007)
5. Ahuja, N., Todorovic, S.: Learning the taxonomy and models of categories present in arbitrary images. In: ICCV (2007)
6. Ahuja, N., Todorovic, S.: Connected segmentation tree – a joint representation of region layout and hierarchy. In: CVPR (2008)
7. Arora, H., Ahuja, N.: Analysis of ramp discontinuity model for multiscale image segmentation. In: ICPR, vol. 4, pp. 99–103 (2006)
8. Basri, R., Jacobs, D.: Recognition using region correspondences. IJCV 25(2), 145–166 (1997)
9. Borenstein, E., Ullman, S.: Class-specific, top-down segmentation. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2351, pp. 109–124. Springer, Heidelberg (2002)
10. Bouman, C.A., Shapiro, M.: A multiscale random field model for Bayesian image segmentation. IEEE Trans. Image Processing 3(2), 162–177 (1994)
11. Brice, C.R., Fennema, C.L.: Scene analysis using regions. Artificial Intelligence 1, 205–226 (1970)
12. Bunke, H., Allermann, G.: Inexact graph matching for structural pattern recognition. Pattern Rec. Letters 1(4), 245–253 (1983)
13. Bunke, H., Foggia, P., Guidobaldi, C., Vento, M.: Graph clustering using the weighted minimum common supergraph. In: Hancock, E.R., Vento, M. (eds.) GbRPR 2003. LNCS, vol. 2726, pp. 235–246. Springer, Heidelberg (2003)
14. Bunke, H., Jiang, X., Kandel, A.: On the minimum common supergraph of two graphs. Computing 65(1), 13–25 (2000)

15. Bunke, H., Kandel, A.: Mean and maximum common subgraph of two graphs. Pattern Rec. Letters 21(2), 163–168 (2000)
16. Canny, J.: A computational approach to edge detection. IEEE TPAMI 8(6), 679–698 (1986)
17. Cao, L., Fei-Fei, L.: Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In: ICCV (2007)
18. Darwish, A.M., Jain, A.K.: A rule based approach for visual pattern inspection. IEEE Trans. Pattern Analysis Machine Intelligence 10(1), 56–68 (1988)
19. Demirci, M.F., Shokoufandeh, A., Keselman, Y., Bretzner, L., Dickinson, S.J.: Object recognition as many-to-many feature matching. Int. J. Computer Vision 69(2), 203–222 (2006)
20. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE TPAMI 28(4), 594–611 (2006)
21. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR, vol. 2, pp. 264–271 (2003)
22. Fidler, S., Leonardis, A.: Towards scalable representations of object categories: Learning a hierarchy of parts. In: CVPR (2007)
23. Finch, A.M., Wilson, R.C., Hancock, E.R.: An energy function and continuous edit process for graph matching. Neural Computation 10(7) (1998)
24. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE TPAMI 6(6) (1984)
25. Glantz, R., Pelillo, M., Kropatsch, W.G.: Matching segmentation hierarchies. Int. J. Pattern Rec. Artificial Intelligence 18(3), 397–424 (2004)
26. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. IEEE TPAMI 18(4), 377–388 (1996)
27. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV (2009)
28. Gu, C., Lim, J.J., Arbelaez, P., Malik, J.: Recognition using regions. In: CVPR (2009)
29. Gupta, A., Nishimura, N.: Finding largest subtrees and smallest supertrees. Algorithmica 21(2), 183–210 (1998)
30. Jiang, H., Drew, M.S., Li, Z.N.: Matching by linear programming and successive convexification. IEEE TPAMI 29(6), 959–975 (2007)
31. Kittler, J., Hancock, E.R.: Contextual decision rule for region analysis. Image Vision Comput. 5(2), 145–153 (1987)
32. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: Workshop on Statistical Learning in Computer Vision, ECCV, pp. 17–32 (2004)
33. Lim, J.J., Arbelaez, P., Gu, C., Malik, J.: Context by region ancestry. In: ICCV (2009)
34. Lindeberg, T.: Scale-Space Theory in Computer Vision. Kluwer Academic Publishers, Norwell (1994)
35. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
36. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI 26, 530–549 (2004)
37. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. IJCV 65(1/2), 43–72 (2005)
38. Mumford, D., Shah, J.: Boundary detection by minimizing functionals. In: CVPR, pp. 22–26 (1985)
39. Opelt, A., Pinz, A., Zisserman, A.: Incremental learning of object detectors using a visual shape alphabet. In: CVPR, vol. 1, pp. 3–10 (2006)
40. Pelillo, M.: Matching free trees, maximal cliques, and monotone game dynamics. IEEE TPAMI 24(11), 1535–1541 (2002)

41. Pelillo, M., Siddiqi, K., Zucker, S.W.: Matching hierarchical structures using association graphs. IEEE TPAMI 21(11), 1105–1120 (1999)
42. Pelillo, M., Siddiqi, K., Zucker, S.W.: Many-to-many matching of attributed trees using association graphs and game dynamics. In: Arcelli, C., Cordella, L.P., Sanniti di Baja, G. (eds.) IWVF 2001. LNCS, vol. 2059, pp. 583–593. Springer, Heidelberg (2001)
43. Peng, J., Bhanu, B.: Closed-loop object recognition using reinforcement learning. IEEE Trans. Pattern Analysis Machine Intelligence 20(2), 139–154 (1998)
44. Qiu, H., Hancock, E.R.: Graph matching and clustering using spectral partitions. Pattern Recognition 39(1), 22–34 (2006)
45. Russell, B.C., Efros, A.A., Sivic, J., Freeman, W.T., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR, vol. 2, pp. 1605–1614 (2006)
46. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. IJCV 77(1-3), 157–173 (2008)
47. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing their shock graphs. IEEE Trans. Pattern Anal. Machine Intell. 26(5), 550–571 (2004)
48. Serre, T., Wolf, L., Poggio, T.: Object recognition with features inspired by visual cortex. In: CVPR, vol. 2, pp. 994–1000 (2005)
49. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE TPAMI 22(8), 888–905 (2000)
50. Shokoufandeh, A., Macrini, D., Dickinson, S., Siddiqi, K., Zucker, S.W.: Indexing hierarchical structures using graph spectra. IEEE TPAMI 27(7), 1125–1140 (2005)
51. Siddiqi, K., Shokoufandeh, A., Dickinson, S.J., Zucker, S.W.: Shock graphs and shape matching. IJCV 35(1), 13–32 (1999)
52. Sudderth, E., Torralba, A., Freeman, W., Willsky, A.: Learning hierarchical models of scenes, objects, and parts. In: ICCV, vol. 2, pp. 1331–1338 (2005)
53. Tabb, M., Ahuja, N.: Multiscale image segmentation by integrated edge and region detection. IEEE Trans. Image Processing 6(5), 642–655 (1997)
54. Todorovic, S., Ahuja, N.: Extracting subimages of an unknown category from a set of images. In: CVPR, vol. 1, pp. 927–934 (2006)
55. Todorovic, S., Ahuja, N.: Learning subcategory relevances to category recognition. In: CVPR (2008)
56. Todorovic, S., Ahuja, N.: Unsupervised category modeling, recognition, and segmentation in images. IEEE TPAMI 30(12), 1–17 (2008)
57. Todorovic, S., Ahuja, N.: Texel-based texture segmentation. In: ICCV (2009)
58. Todorovic, S., Ahuja, N.: Region-based hierarchical image matching. IJCV (to appear)
59. Torralba, A., Murphy, K., Freeman, W.: Sharing features: efficient boosting procedures for multiclass object detection. In: CVPR, vol. 2, pp. 762–769 (2004)
60. Torsello, A., Hancock, E.R.: Matching and embedding through edit-union of trees. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 822–836. Springer, Heidelberg (2002)
61. Torsello, A., Hancock, E.R.: Learning shape-classes using a mixture of tree-unions. IEEE Trans. PAMI 28(6), 954–967 (2006)
62. Torsello, A., Robles-Kelly, A., Hancock, E.R.: Discovering shape classes using tree edit-distance and pairwise clustering. IJCV 72(3), 259–285 (2007)
63. Tu, Z., Yuille, A.: Shape matching and recognition - using generative models and informative features. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 195–209. Springer, Heidelberg (2004)
64. Weiss, I., Ray, M.: Recognizing articulated objects using a region-based invariant transform. IEEE Trans. Pattern Analysis Machine Intelligence 27(10), 1660–1665 (2005)

65. Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. In: ICCV, vol. 2, pp. 1800–1807 (2005)
66. Winn, J., Jojic, N.: Locus: learning object classes with unsupervised segmentation. In: ICCV, pp. 756–763 (2005)
67. Worthington, P.L., Hancock, E.R.: Object recognition using shape-from-shading. IEEE TPAMI 23(5), 535–542 (2001)
68. Worthington, P.L., Hancock, E.R.: Region-based object recognition using shape-from-shading. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 455–471. Springer, Heidelberg (2000)
69. Zhang, R., Zhang, Z.: Hidden semantic concept discovery in region based image retrieval. In: CVPR, vol. 2, pp. 996–1001 (2004)

Learning on Manifolds[⋆]

Fatih Porikli

Mitsubishi Electric Research Labs,
Cambridge, MA, USA

Abstract. Mathematical formulation of certain natural phenomena exhibits group structure on topological spaces that resemble the Euclidean space only on a small enough scale, which prevents incorporation of conventional inference methods that require global vector norms. More specifically in computer vision, such underlying notions emerge in differentiable parameter spaces. Here, two Riemannian manifolds including the set of affine transformations and covariance matrices are elaborated and their favorable applications in distance computation, motion estimation, object detection and recognition problems are demonstrated after reviewing some of the fundamental preliminaries.

Keywords: Region Covariance, Riemannian Geometry, Detection, Tracking, Regression, Classification.

1 Topological Spaces

A group $\mathcal{G}$ is a set that is endowed with a binary operation and satisfies the closure, associativity, identity, and invertibility properties. A simple example of a group is the set of integers $\mathbb{Z}$ under addition where the identity is 0 and the inverse of any integer is its negative, which is still in $\mathbb{Z}$. Note that, if the binary operation is chosen to be multiplication the set of integers is no longer a group because the inverse may not be an integer. A subset of $\mathcal{G}$ is called as a subgroup if it satisfies all the group properties of being a group under the same binary operation. For example the set of positive rational numbers $\mathbb{Q}^+$ forms a subgroup of rational numbers under multiplication. Yet, the set of negative rational numbers $\mathbb{Q}^-$ is not a subgroup since it does not contain the identity and it is not closed under multiplication.

A topological space is a set $\mathcal{S}$ together with a family of subsets $\mathcal{T}$ if the empty set $\emptyset \in \mathcal{T}$ and $S \in \mathcal{T}$, the union of any family of sets in $\mathcal{T}$ also lies in $\mathcal{T}$, and the intersection of any finite number of sets in $\mathcal{T}$ belongs to $\mathcal{T}$. The family $\mathcal{T}$ is said to be the a topology of $\mathcal{S}$ and the sets in $\mathcal{T}$ are called open sets of the topological space. A given set may have many different topologies. Any open set $\mathcal{U} \in \mathcal{T}$ which contains point $X \in \mathcal{S}$ is called the neighborhood of the point. A Hausdorff space is a topological space in which distinct points have disjoint

[⋆] Throughout this paper, learning on manifolds refers to the family of supervised and unsupervised methods to search, cluster, classify, and recognize given observations on smooth manifolds without flattening, charting, or dimensionality reducing them.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 20–39, 2010.

neighborhoods, such that, $X, Y \in \mathcal{S}$ and there exists $\mathcal{U}_X, \mathcal{U}_Y \in \mathcal{T}$, $X \in \mathcal{U}_X$, $Y \in \mathcal{U}_Y$ and $\mathcal{U}_X \cap \mathcal{U}_Y = \emptyset$. For instance, the real numbers constitute a Hausdorff space.

For functions defined on Hausdorff spaces it is possible to introduce notions such as continuity by saying that as we move towards a point X, the value of the function gets closer to the value of the function at the point. The idea of being 'close' to a particular point is captured by its neighborhood and the continuity of a function is defined by how it maps open sets of the topology. A mapping between two topological spaces is called continuous if the inverse image of any open set with respect to the mapping is again an open set. A bijective (one-to-one and onto) mapping that is continuous in both directions is called a homeomorphism. Such mappings preserve the topological properties of a given space. Two spaces with a homeomorphism between them are called homeomorphic, and from a topological viewpoint, they are the same, e.g. a square and a circle are homeomorphic to each other, but a sphere and a torus are not.

A manifold $\mathcal{M}$ of dimension d is a connected Hausdorff space for which every point has a neighborhood that is homeomorphic to an open subset $\mathcal{U}$ of $\mathbb{R}^d$. In other words, a manifold corresponds to a topological space which is locally similar to an Euclidean space. For any point $X \in \mathcal{M}$, there exists an open neighborhood $\mathcal{U} \subset \mathcal{M}$ containing the point and homeomorphism ϕ mapping the neighborhood to an open set $\mathcal{V} \subset \mathbb{R}^d$, such that $\phi : \mathcal{U} \mapsto \mathcal{V}$. The pair $(\mathcal{U}, \phi)$ is called as a coordinate chart. An atlas is a family of charts for which the open sets constitute an open covering of the manifold. Every topological manifold has an atlas.

Let $(\mathcal{U}_X, \phi_X)$ and $(\mathcal{U}_Y, \phi_Y)$ be two coordinate charts, such that, $\mathcal{U}_X \cap \mathcal{U}_Y$ is nonempty. The transition map $\phi_X \circ \phi_Y^{-1}$ is a mapping between two open sets $\phi_X(\mathcal{U}_X \cap \mathcal{U}_Y)$ and $\phi_Y(\mathcal{U}_X \cap \mathcal{U}_Y)$. In other words, the transition maps relate the coordinates defined by the various charts to one another. A differentiable manifold C^k is a topological manifold equipped with an equivalence class of atlas whose transition maps are k-times continuously differentiable. In case all the transition maps of a differentiable manifold are smooth, i.e. all its partial derivatives exist, then it is a smooth manifold C^∞.

It is possible to define the derivatives of the curves on a differentiable manifold and attach to every point X a tangent space T_X, a real vector space that intuitively contains the possible directions in which one can tangentially pass through X. Suppose two curves with $\gamma_1(0) = \gamma_2(0) = X$ are equivalent, that is the ordinary derivatives of $\phi \circ \gamma_1$ and $\phi \circ \gamma_2$ at 0 coincide for all charts $(\mathcal{U}, \phi)$ where $X \in \mathcal{U}$. A tangent vector at X is defined by the equivalence class of the smooth curves $\gamma(0) = X$. Tangent vectors are the tangents to the smooth curves lying on the manifold. The tangent space T_X is the set of all tangent vectors at X. The tangent space is a vector space, thereby it is closed under addition and scalar multiplication.

A Riemannian manifold $(\mathcal{M}, g)$ is a differentiable manifold in which each tangent space has an inner product g metric, which varies smoothly from point to point. It is possible to define different metrics on the same manifold to obtain different Riemannian manifolds. In practice this metric is chosen by requiring it to be invariant to some class of geometric transformations. The inner product g induces a norm for the tangent vectors on the tangent space $\|X\|^2 =< X, X >= g(X)$. A detailed description of these concepts can be found in [1].

A Lie group is a group $\mathcal{G}$ with the structure of a differentiable manifold such that the group operations, multiplication and inverse, are differentiable maps. The tangent space to the identity element of the group forms a Lie algebra. The group operation provides Lie groups with additional algebraic structure. Let $X \in \mathcal{G}$. Left multiplication by the inverse of the group element $X^{-1} : \mathcal{G} \to \mathcal{G}$ maps the neighborhood of X to neighborhood of identity. The inverse mapping is defined by left multiplication by X.

2 Distance on Riemannian Manifolds

A geodesic is a smooth curve that locally joins their points along the shortest path. Suppose $\gamma(r) : [r_0, r_1] \mapsto \mathcal{M}$ be a smooth curve on $\mathcal{M}$. The length of the curve $L(\gamma)$ is defined as

$$L(\gamma) = \int_{r_0}^{r_1} \|\gamma'(r)\| dr. \tag{1}$$

A smooth curve is called geodesic if and only if its velocity vector is constant along the curve $\|\gamma'(r)\| = const$. Suppose X and Y be two points on $\mathcal{M}$. The distance between the points $d(X, Y)$, is the infimum of the length of the curves, such that, $\gamma(r_0) = X$ and $\gamma(r_1) = Y$. All the shortest length curves between the points are geodesics but not vice-versa. However, for nearby points the definition of geodesic and the shortest length curve coincide. For each tangent vector $x \in T_X$, there exists a unique geodesic γ starting at $\gamma(0) = X$ having initial velocity $\gamma'(0) = x$.

The exponential map, $\exp_X : T_X \mapsto \mathcal{M}$, maps the vector y in the tangent space to the point reached by the geodesic after unit time $\exp_X(y) = 1$. Since the velocity along the geodesic is constant, the length of the geodesic is given by the norm of the initial velocity $d(X, \exp_X(y)) = \|y\|_X$. An illustration is shown in Figure 1. Under the exponential map, the image of the zero tangent vector is the point itself $\exp_X(0) = X$. For each point on the manifold, the exponential map is a diffeomorphism (one-to-one, onto and continuously differentiable mapping

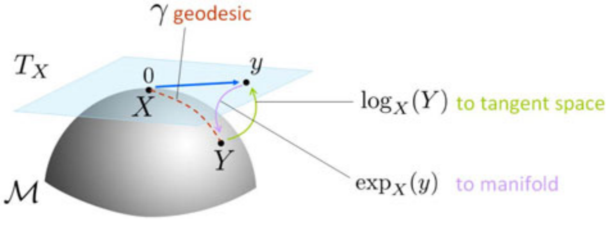


Fig. 1. Manifold and tangent space

in both directions) from a neighborhood of the origin of the tangent space T_X onto a neighborhood of the point X. In general, the exponential map $\exp_X$ is onto but only one-to-one in a neighborhood of X. Therefore, the inverse mapping $\log_X : \mathcal{M} \mapsto T_X$ is uniquely defined only around the neighborhood of the point X. If for any $Y \in \mathcal{M}$, there exists several $y \in T_X$ such that $Y = \exp_X(y)$, then $\log_X(Y)$ is given by the tangent vector with the smallest norm. Notice that both operators are point dependent. For certain manifolds the neighborhoods can be extended to the whole tangent space and manifold hence the exponential map is a global diffeomorphism. From the definition of geodesic and the exponential map, the distance between the points on manifold can be computed by

$$d(X,Y) = d(X, \exp_X(y)) =< \log_X(Y), \log_X(Y) >_X = \| \log_X(Y) \|_X = \|y\|_X. \tag{2}$$

For Riemannian manifolds endowing an inverse mapping, the geodesic distance between two group elements can be written as

$$d(X,Y) = \| \log(X^{-1}Y) \|. \tag{3}$$

The exponential identity $\exp(X)\exp(Y) = \exp(X+Y)$ does not hold for noncommutative matrix Lie groups. The identity is expressed through Baker-Campbell-Hausdorff formula [2] $\exp(X)\exp(Y) = \exp(\mathrm{BCH}(X,Y))$ as

$$\mathrm{BCH}(X,Y) = X + Y + \frac{1}{2}[X,Y] + O(|(X,Y)|^3). \tag{4}$$

where $[X,Y] = XY - YX$ is the Lie bracket operation for nonsingular matrix group.

2.1 Space of Nonsingular Covariance Matrices

The $d \times d$ dimensional symmetric positive definite matrices $\mathbb{S}_d^+$, can be formulated as a Riemannian manifold. Let points on this manifold are covariance matrices X, Y. An invariant Riemannian metric on the tangent space of $\mathbb{S}_d^+$ is given by [4]

$$< y, z >_{\mathrm{X}} = \mathrm{tr}\left(\mathrm{X}^{-\frac{1}{2}} y \mathrm{X}^{-1} z \mathrm{X}^{-\frac{1}{2}}\right). \tag{5}$$

The exponential map associated to the Riemannian metric

$$\exp_{\mathrm{X}}(y) = \mathrm{X}^{\frac{1}{2}} \exp\left(\mathrm{X}^{-\frac{1}{2}} y \mathrm{X}^{-\frac{1}{2}}\right) \mathrm{X}^{\frac{1}{2}} \tag{6}$$

is a global diffeomorphism. Therefore, the logarithm is uniquely defined at all the points on the manifold

$$\log_{\mathrm{X}}(\mathrm{Y}) = \mathrm{X}^{\frac{1}{2}} \log\left(\mathrm{X}^{-\frac{1}{2}} \mathrm{Y} \mathrm{X}^{-\frac{1}{2}}\right) \mathrm{X}^{\frac{1}{2}}. \tag{7}$$

Above,the exp and log are the ordinary matrix exponential and logarithm operators. Not to be confused, $\exp_{\mathrm{X}}$ and $\log_{\mathrm{X}}$ are manifold specific operators which are also point dependent, $X \in \mathbb{S}_d^+$. The tangent space of $\mathbb{S}_d^+$ is the space of $d \times d$ symmetric matrices and both the manifold and the tangent spaces are $d(d+1)/2$ dimensional.

For symmetric matrices, the ordinary matrix exponential and logarithm operators can be computed easily. Let $\Sigma = \mathrm{UDU}^T$ be the eigenvalue decomposition of a symmetric matrix. The exponential series is

$$\exp(\Sigma) = \sum_{k=0}^{\infty} \frac{\Sigma^k}{k!} = \mathrm{U} \exp(\mathrm{D}) \mathrm{U}^T \tag{8}$$

where exp(D) is the diagonal matrix of the eigenvalue exponentials. Similarly, the logarithm is given by

$$\log(\Sigma) = \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} (\Sigma - \mathrm{I})^k = \mathrm{U} \log(\mathrm{D}) \mathrm{U}^T. \tag{9}$$

The exponential operator is always defined, whereas the logarithms only exist for symmetric matrices with positive eigenvalues, $\mathbb{S}_d^+$. From the definition of the geodesic given in the previous section, the distance between two points on $\mathbb{S}_d^+$ is measured by substituting (7) into (5)

$$\begin{aligned} d^2(\mathrm{X}, \mathrm{Y}) &= < \log_{\mathrm{X}}(\mathrm{Y}), \log_{\mathrm{X}}(\mathrm{Y}) >_{\mathrm{X}} \\ &= \mathrm{tr}\left(\log^2(\mathrm{X}^{-\frac{1}{2}} \mathrm{Y} \mathrm{X}^{-\frac{1}{2}}) \right). \end{aligned} \tag{10}$$

An equivalent form of the affine invariant distance metric was first given in [3], in terms of joint eigenvalues of X and Y as

$$d(\mathrm{X}, \mathrm{Y}) = \left(\sum_{k=1}^{d} (\ln \lambda_k(\mathrm{X}, \mathrm{Y}))^2 \right)^{\frac{1}{2}} \tag{11}$$

where $\lambda_k(\mathrm{X}, \mathrm{Y})$ are the generalized eigenvalues of X and Y, computed from

$$\lambda_k \mathrm{X} \mathbf{v}_k - \mathrm{Y} \mathbf{v}_k = 0 \qquad k = 1 \ldots d \tag{12}$$

and $\mathbf{v}_k$ are the generalized eigenvectors. This distance measure satisfies the metric axioms, positivity, symmetry, triangle inequality, for positive definite symmetric matrices.

An orthogonal coordinate system on the tangent space can be defined by the vector operation. The orthogonal coordinates of a vector y on the tangent space at point X is given by

$$\mathrm{vec}_{\mathrm{X}}(y) = \mathrm{upper}(\mathrm{X}^{-\frac{1}{2}} y \mathrm{X}^{-\frac{1}{2}}) \tag{13}$$

where upper refers to the vector form of the upper triangular part of the matrix. The mapping $\mathrm{vec}_{\mathrm{X}}$, relates the Riemannian metric (5) on the tangent space to the canonical metric defined in $\mathbb{R}^d$.

2.2 Region Covariance Descriptor and Pattern Search

Suppose θ be a feature map extracted from a given image I comprising pixel coordinates, color values, pixel-wise derivatives, oriented gradients, filter responses, etc. of appearance and spatial attributes $\theta_{m,n} = [m, n, I, I_m, ...]^T_{m,n}$. Different functions of coordinates enables imposing of different spatial structures e.g. rotational invariance, symmetry, etc.

A region covariance matrix X for any image region is defined as

$$\mathrm{X} = \frac{1}{N} \sum_{m,n \in R}^{N} (\theta_{m,n} - \bar{\theta})(\theta_{m,n} - \bar{\theta})^T \tag{14}$$

where N is the number of pixels and $\bar{\theta}$ is the mean vector of the corresponding features within the region R. Note that, this is not the computation of the covariance of two image regions, but the covariance of image features of a region. Refer to [5] for more details. Such a descriptor provides a natural way of fusing multiple features without a weighted average. Instead of evaluating the first order statistics of feature distributions through histograms, it embodies the second order characteristics. The noise corrupting individual samples are largely filtered out by the multitude of pixels. It endows spatial scale and feature shift invariance. It is possible to compute covariance matrix from feature images in a very fast way using integral image representation [6]. After constructing $d(d+1)/2$ tensors of integral images corresponding to each feature dimension and multiplication of any two feature dimensions, the covariance matrix of any arbitrary rectangular region can be computed in $\mathcal{O}(d^2)$ time independent of the region size.

The space of region covariance descriptors is not a vector space. For example, it is not closed under multiplication with negative scalars. They constitute the space of positive semi-definite matrices $\mathbb{S}_d^{0,+}$. By adding a small diagonal matrix (or guaranteeing no features in the feature vectors would be exactly identical), they can be transformed into $\mathbb{S}_d^+$, which is a Riemannian manifold, in order to apply the Riemannian metrics ([10, 11]).

A first example using the covariance region descriptor is pattern search to locate a given object of interest in an arbitrary image. To find the most similar region in the image, distances between the descriptors of the object and candidate regions are computed. Each pixel of the image is converted to a 9-dimensional feature vector $\theta_{m,n} = \left[m, n, I^r, I^g, I^b, |I_m|, |I_n|, |I_{mm}|, |I_{nn}|\right]^T_{m,n}$ where $I^{r,g,b}$ are

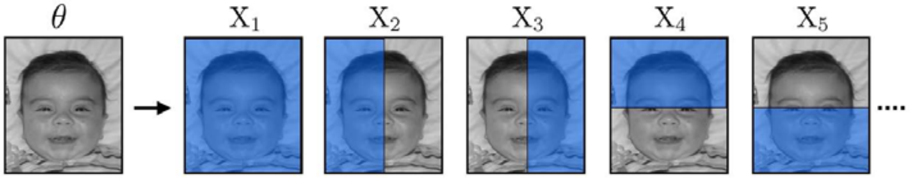


Fig. 2. Object representation by multiple covariance matrices of subregions

the RGB color values, and $I_{m,n}$ are spatial derivatives. An object is represented by a collection of partial region covariance matrices as shown in Figure 2.

At the first phase, only the covariance matrix of the whole region from the source image is computed. The target image is searched for a region having similar covariance matrix at all the locations and different scales. A brute force search can be performed since the covariance of an arbitrary region can be obtained efficiently. Instead of scaling the target image, the size of the search window is changes. Keeping the best matching locations and scales, the search for initial detections is repeated using the covariance matrices of N_R partially occluded subregions at the second phase. The distance of the object model O and a candidate region R is computed as

$$|O - R| = min_j \left[\sum_{i=0}^{N_R} d(\mathrm{X}_i^R, \mathrm{X}_i^O) - d(\mathrm{X}_j^R, \mathrm{X}_j^O) \right] \tag{15}$$

where the worst match is dismissed to provide robustness towards possible occlusions and changes. The region with the smallest distance is selected as the matching region. Sample matching results are presented in Figure 3 where the manifold search using the Riemannian metrics is compared to the histogram features using the Bhattacharyya distance.

Region covariance descriptor can be used for texture recognition within a k-NN framework. Each texture class in the training dataset is represented by a bag of region covariance descriptors of the randomly sampled subregions with random sizes between 16 × 16 and 128 × 128. Given a test image, a certain number of subregions are extracted and their descriptors are computed. For each covariance matrix, the distances from matrices in the training set are calculated. The label is predicted according to the majority voting among the k nearest neighbors. Votes are accumulated for all images in the dataset and the class having the maximum vote is assigned as the matching class.

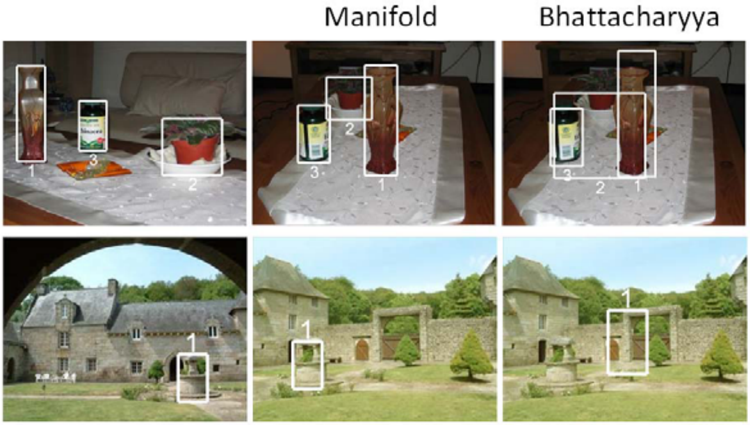


Fig. 3. Regions found via region covariance descriptor and feature histograms

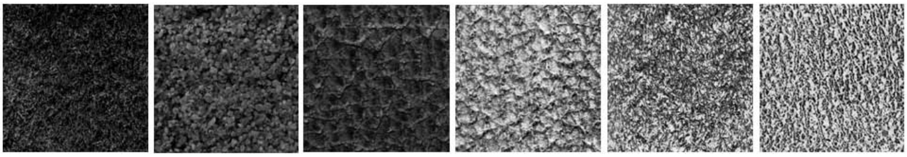


Fig. 4. Samples from 6 classes that are all correctly classified 109 classes out of 112

A multi-class classifier on Brodatz texture database that consists of 112 gray scale textures (Figure 4) is also tested. Image intensities and norms of first and second order derivatives in both x and y direction are incorporated into the pixel feature vector. Each pixel is mapped to a $d = 5$ dimensional feature space (only 15 independent coefficients). Due to the nonhomogeneous nature, recognition on this dataset is a challenging task. Each 640 × 640 texture image is divided into four 320 × 320 subimages and half of the images are used for training and half for testing. The k-NN on manifold is compared with the results reported in [7]. Even though the best performing conventional approach utilizes computationally very expensive texton histograms of 560 coefficients, its performance is limited to 97.32%. Experiments with 100 random covariances from each texture image, $k = 5$ for the k-NN algorithm shows 97.77% recognition with a fraction of the load.

3 Computing Mean on Riemannian Manifolds

Similar to Euclidean spaces, the Karcher mean [8] of points on Riemannian manifold, is the point on $\mathcal{M}$ which minimizes the sum of squared distances

$$\bar{X} = \arg \min_{X \in \mathcal{M}} \sum_{k=1}^{K} d^2(X_k, X) \tag{16}$$

where the distance metric is defined by (10,11).

Differentiating the error function with respect to X and setting it equal to zero gives the following gradient descent procedure [4]

$$\bar{X}^{j+1} = \exp_{\bar{X}^j} \left[\frac{1}{K} \sum_{k=1}^{K} \log_{\bar{X}^j}(X_k) \right] \tag{17}$$

which finds a local minimum of the error function. The method iterates by computing the first order approximations to the mean on the tangent space. The weighted mean computation is similar to arithmetic mean. Replacing the inside of the exponential, the mean of the tangent vectors with the weighted mean can be obtained as

$$\bar{X}^{j+1} = \exp_{\bar{X}^j} \left[\frac{1}{\sum w_k} \sum_{k=1}^{K} w_k \log_{\bar{X}^j}(X_k) \right]. \tag{18}$$

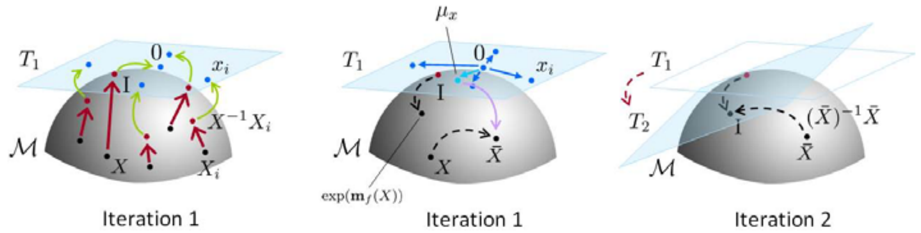


Fig. 5. Mean computation is achieved by transforming points on manifold to the neighborhood of I on the manifold by $X^{-1}X_i$, mapping them to the tangent space of X, finding the mean in the tangents space, back projecting the tangent space mean onto the manifold, and repeating these steps until the dislocation between the successive iterations becomes negligible

3.1 Object Model Update

Finding the correspondences of the previously detected objects in the current frame, called as tracking, is an essential task in many computer vision applications.

For a given object region, the covariance matrix of the features can be computed as the model of the object. within all possible locations of the current frame, the region that has the minimum covariance distance from the model can be searched and assigned as the estimated location. Note that such an exhaustive search is performed to highlight the discriminant power of the region descriptor and the distance metric on manifold. Often search is constrained by a predictive prior. In order to adapt to variations in object appearance, a set of previous covariance matrices are stored and a mean covariance matrix is computed on the manifold as the object representative. Sample tracking results are shown in Figure 6 below.

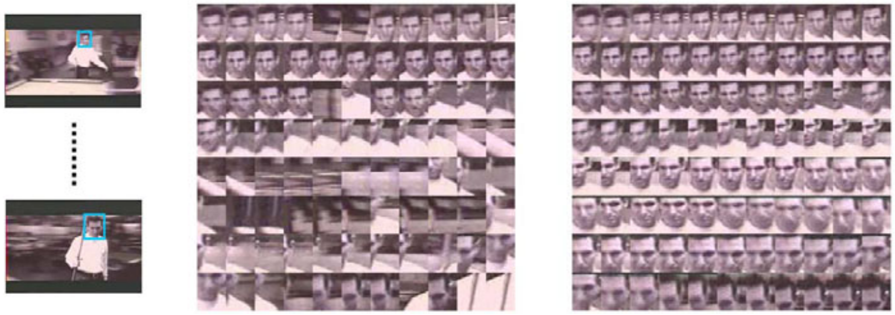


Fig. 6. Montages of the detection results (middle) without model update: detection rate is 47.7%, (right) with weighted mean based update mechanism on manifold: detection rate is 100%

4 Computing Kernel Density

The mean-shift is a nonparametric clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters. Data points are assumed to be originated from an unknown distribution which is approximated by kernel density estimation in vector spaces

$$f(x) = \frac{1}{N}\sum_{i=1}^{N} H\left(x - x_i\right) = \frac{\kappa}{N}\sum_{i=1}^{N} h\left(\|x - x_i\|^2\right) \tag{19}$$

where $H(x) = \kappa h(\|x\|^2)$ is a radially symmetric kernel with unit radius. The cluster centers are located by the mean-shift procedure and the data points associated with the same modes produce a partitioning of the feature space. By taking the gradient of the above equation, the stationary points of the density function can be found iteratively via

$$\bar{x} = \frac{\sum_i x_i \cdot k\left(\|x - x_i\|^2\right)}{\sum_i k\left(\|x - x_i\|^2\right)} \tag{20}$$

where $k(x) = -h'(x)$. At each step, a local weighted mean is computed, and the computation is repeated centered on the previous estimate. The difference between the current and the previous location estimates is called the mean-shift vector

$$m(x) = \bar{x} - x. \tag{21}$$

Starting at each data point, mean-shift iterations convergence to a local mode of the distribution, i.e. a basin of attraction.

A generalization of the mean-shift procedure for parameter spaces having matrix Lie group structure where the mean-shift algorithm runs on a Lie group by iteratively transforming points between the Lie group (on Riemannian manifold) and Lie algebra (on tangent space). Using the intrinsic distance, the multivariate kernel density estimate at X is given by

$$f(X) = \frac{\kappa}{N}\sum_{i=1}^{N} h\left(\|\log(X^{-1}X_i)\|^2\right) \tag{22}$$

where $x_i = \log(X^{-1}X_i)$.

The group operation maps the neighborhood of X to the neighborhood of I and the tangent space at X to the Lie algebra g. The approximation error can be expressed in terms of the higher order terms in BCH formula ([4]). The error is minimal around I and the mapping assures that the error is minimized. The point X is mapped to 0, thus the second term in the mean-shift vector does not exists. The mean-shift vector on the tangent space can be transferred to the Lie group as

$$m(X) = \exp\left(\frac{\sum_i \log(X^{-1}X_i) \cdot k\left(\|\log(X^{-1}X_i)\|^2\right)}{\sum_i k\left(\|\log(X^{-1}X_i)\|^2\right)}\right) \tag{23}$$

and the location of X can be updated as

$$\bar{X} = X \exp(m(X)). \tag{24}$$

An invariant estimator on the linear group of non-singular matrices with positive determinant can be found in [11].

4.1 Motion Detection

Several parameter spaces which commonly occur in computer vision problems do not form a vector space. For instance, the set of all affine transformations forms a matrix Lie group. Two-dimensional affine transformation $A(2)$ is given by the set of matrices in the following form

$$\mathrm{X} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ 0 & 1 \end{bmatrix}_{3\times 3} \tag{25}$$

where $\mathbf{A}$ is a nonsingular 2×2 matrix. By selecting each of the entries as an orthonormal basis, X constitutes a $d = 6$ dimensional manifold.

One application of the mean-shift on manifolds is multiple rigid motion estimation from noisy point correspondences in presence of large amount of outliers [9]. Given two images, local feature points such as corner points are found. These points are paired via a descriptor matching algorithm. Due to occlusions and errors in the point matching process most of the point correspondences are outliers. For each set of randomly selected 3-point correspondences a 2D rigid affine transformation $(\mathbf{A}, \mathbf{b})$ is estimated. These transformations constitute the set of X. Then the above mean-shift procedure is applied to find the local modes that represent rigid objects having distinct affine motions. A sample result is given in Figure 7.

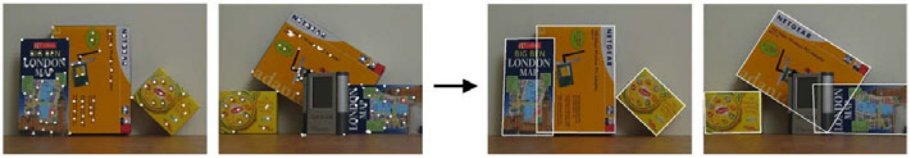


Fig. 7. (Left) 2D images with 83 points are detected via corner detection algorithm. Less than 50% of the point correspondences are accurate. (Right) The boundaries of the bodies and transformed boundaries with the estimated motion parameters. The estimation is almost perfect. *Courtesy O. Tuzel.*

5 Linear Regression on Riemannian Manifolds

Regression refers to understand the relationship between multiple variables. Linear regression assumes the relationship depends linearly on a model in which the conditional mean of a scalar variable given the other variables is an affine function of those variables. Numerous procedures have been developed for parameter estimation and inference in linear regression. Here a least squares estimator is described.

Suppose (α_i, X_i) are the pairs of observed data $\alpha \in \mathbb{R}^d$ in vector space and the corresponding points on the manifold $X \in \mathcal{M}$. The regression function φ maps the vector space data onto the manifold $\varphi : \mathbb{R}^d \mapsto \mathcal{M}$. An objective function is defined as the sum of the squared geodesic distances between the estimations $\varphi(\alpha_i)$ and the points X_i

$$J = \sum_i d^2 \left[\varphi(\alpha_i), X_i\right]. \tag{26}$$

Assuming a Lie algebra on the manifold can be defined, the objective function can be written using the Baker-Campbell-Hausdorff approximation (4) as

$$J = \sum_i \left\|\log\left[\varphi^{-1}(\alpha_i) X_i\right]\right\|^2 \approx \sum_i \left\|\log\left[\varphi(\alpha_i)\right] - \log\left[X_i\right]\right\|^2 \tag{27}$$

up to the first order terms. The regression function φ can be written as

$$\varphi(\alpha_i) = \exp\left(\alpha_i^T \varOmega\right) \tag{28}$$

to learn the function $\varOmega : \mathbb{R}^d \mapsto \mathbb{R}^n$ which estimates the tangent vectors $\log(X_i)$ on the Lie algebra where $\varOmega$ is the $d \times n$ matrix of regression coefficients. Thus, the objective function (27) becomes

$$J = \sum_i \left\|\alpha_i^T \varOmega - \log\left[X_i\right]\right\|^2 \tag{29}$$

Let $\mathbf{X}$ be the $k \times d$ matrix of initial observations and $\mathbf{Y}$ be the $k \times n$ matrix of mappings to the Lie algebra

$$\mathbf{X} = \begin{bmatrix} [\alpha_1]^T \\ \vdots \\ [\alpha_k]^T \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} [\log(X_1)]^T \\ \vdots \\ [\log(X_k)]^T \end{bmatrix} \tag{30}$$

Substituting (30) into (29), one can obtain

$$J = tr[(\mathbf{X}\varOmega - \mathbf{Y})^T(\mathbf{X}\varOmega - \mathbf{Y})] \tag{31}$$

where the trace replaces the summation in (27). Differentiating the objective function J with respect to $\varOmega$, the minimum is achieved at $\varOmega = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$. To avoid overfitting, additional constraints on the size of the regression coefficients can be introduced

$$J = tr[(\mathbf{X}\varOmega - \mathbf{Y})^T(\mathbf{X}\varOmega - \mathbf{Y})] + \beta\|\varOmega\|^2 \tag{32}$$

which is called the ridge regression [10]. The minimizer of the cost function J is given by $\varOmega = (\mathbf{X}^T\mathbf{X} + \beta\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}$ where $\mathbf{I}$ is an $d \times d$ identity matrix. The regularization coefficient β determines the degree of shrinkage on the regression coefficients.

5.1 Affine Motion Tracking

At the initialization of the object, the affine motion tracker estimates a regression function that maps the region feature vectors to the hypothesized affine motion vectors by first hypothesizing a set of random motion vectors within the given bounds, determining the transformed regions for these motions, and then computing the corresponding features within each warped region. In the tracking time, it extracts the feature vector only for the previous object region location and applies the learned regression function.

Let M transforms a unit square at the origin to the affine region enclosing the target object $[x \;\; y \;\; 1]_I^T = \mathrm{M}[x \;\; y \;\; 1]_O^T$ where the subscripts indicate the image and object coordinates respectively. The inverse M^{-1} is an affine motion matrix and transforms the image coordinates to the object coordinates. The aim of tracking is to estimate the transformation matrix M_i, given the previous images and the initial transformation M_0. The transformations are modeled incrementally

$$\mathrm{M}_i = \mathrm{M}_{i-1}.\Delta\mathrm{M}_i \tag{33}$$

and estimate the increments $\Delta\mathrm{M}_i$ at each time. The transformation $\Delta\mathrm{M}_i$ corresponds to motion of target from time $i-1$ to i in the object coordinates.

Suppose the target region is represented with orientation histograms computed at a regular grid inside the unit square in object coordinates, i.e with $\alpha(I(\mathrm{M}_i^{-1})) \in \mathbb{R}^d$ where d is the dimension of the descriptor. Given the previous location of the object M_{i-1} and the current observation I_i, the new transformation $\Delta\mathrm{M}_i$ by the regression function is estimated as

$$\Delta\mathrm{M}_i = \varphi(\alpha_i(\mathrm{M}_{i-1}^{-1})). \tag{34}$$

The problem reduces to learning and updating the regression function φ.

During the learning step, a training set of K random affine transformation matrices $\{\Delta\mathrm{M}_j\}_{j=1\ldots K}$ are generated around the identity matrix (Figure 8). The approximation is good enough since the transformations are in a small neighborhood of the identity. The object coordinates are transformed by multiplying on the left with $\Delta\mathrm{M}_j^{-1}$ and the descriptor α_j is computed at $\Delta\mathrm{M}_j^{-1}.\mathrm{M}_i^{-1}$. The transformation M_i^{-1} moves the object back to the unit square. The training set consists of samples $\{\alpha_j, \Delta\mathrm{M}_j\}_{j=1\ldots K}$. The size of the training set is kept relatively small $K = 200$. Since number of samples is smaller than the dimension of the feature space, $K < d$, the system is underdetermined. To relieve this, the ridge regression is applied to estimate the regression coefficients.

Since objects can undergo appearance changes in time, it is necessary to adapt to these variations. The model update achieves reestimating the regression function. During tracking, a set of random observations are generated at each frame with the same method described above. The observations stored for most recent frames constitute the update training set. More details and an importance sampling based adaptation can be found in [12].

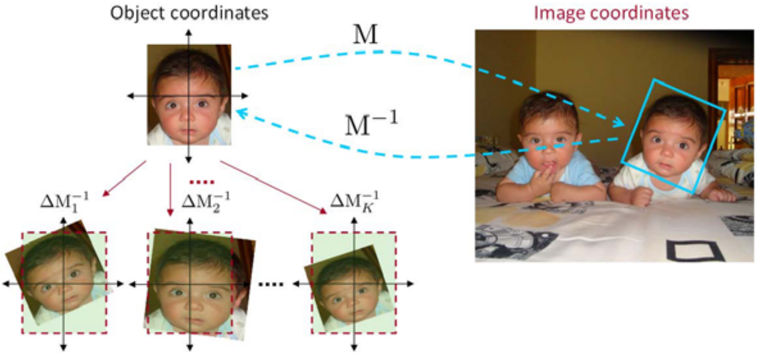


Fig. 8. Random transformations are applied in object coordinates to generate the training features for regression function estimation

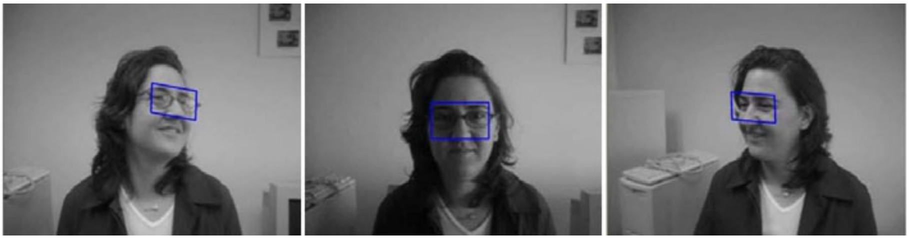


Fig. 9. Regression tracking on manifold for a given region. Note that the tracking is still valid even the region undergoes out-of-plane rotations.

5.2 Pose Invariant Detection

Above method can be used to build an affine invariant object detection algorithm by incorporating a class specific regression function to an existing pose dependent detector. Instead of learning a tracking function of the specific target object, a regression function of the object class is trained. The learning is performed on the training set generated by applying a total of K random affine transformations to multiple samples from the same class, e.g. face images. The training stage is an offline process and a more complicated model can be learned compared to tracking applications. However, the learned function should be evaluated fast at runtime, since the tracker is initiated at several locations for each test image.

On a sparse grid on the test image a sparse scan of the image is performed. At each grid point the class specific regression function is applied and the region it converges is determined. This scan finds all the locations in the motion space (e.g. affine) which resemble the object model. The object detector is then evaluated only at these locations. The benefits of the approach is two-fold. First, the size of the search space drastically reduces. Secondly, it performs continuous estimation of the target pose in contrast to the existing techniques perform search on a quantized space. Utilizing a pose dependent object detection

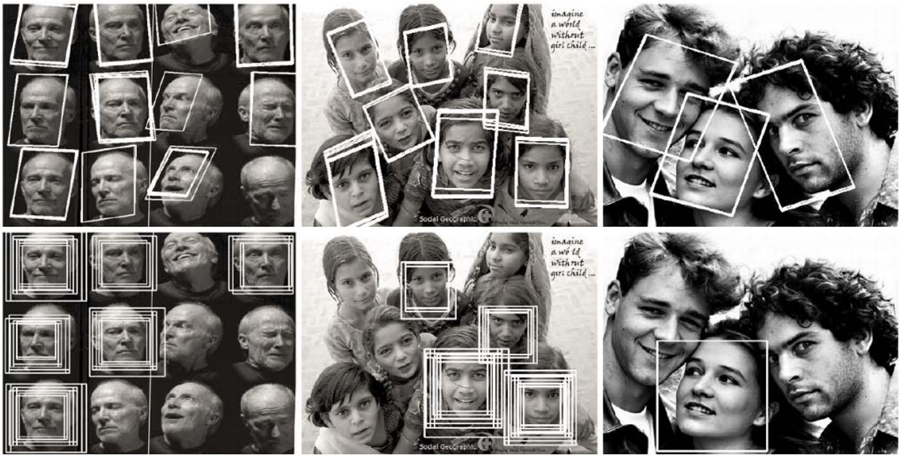


Fig. 10. (Top) Class specific affine invariant face detection. (Bottom) VJ multi-pose face detector results for sample images containing non-frontal faces.

algorithm (e.g., frontal in upright position), the method enables to detect objects in arbitrary poses.

In experiments on a face dataset which consists of 803 face images from CMU, MIT and MERL datasets, the Viola and Jones (VJ) face detector [13] evaluated at the affine warped face images could detect only 5% of the faces that are norm 0.5 distant. The Lie algebra based estimation is significantly superior by achieving 95.6% detection for the same images. Sample detection results for affine invariant detection of faces are given in Figure 10.

6 Classifiers on Riemannian Manifolds

Let $\{(X_i, y_i)\}_{i=1...N}$ be the training set with respect to class labels, where $X_i \in \mathcal{M}$, $y_i \in \{0, 1\}$. Our task is to find a classifier $Z(X) : \mathcal{M} \mapsto \{0, 1\}$, which divides the manifold into two sets based on the training samples of labeled points. A function to divide a manifold is an intricate notion compared to Euclidean spaces. A linear classifier that is represented by a point and a direction vector on $\mathbb{R}^2$ separates the space into two. However, such lines on the 2-torus cannot divide the manifold. A straightforward approach for classification would be to map the manifold to a higher dimensional Euclidean space, which can be considered as flattening or charting the manifold. However, there is no such mapping that globally preserves the distances between the points on the manifold in general.

6.1 Local Maps and Boosting

One can design an incremental approach by training several weak classifiers on the tangent space and combining them through boosting. Since the mappings

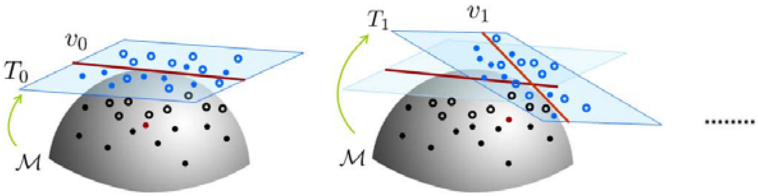


Fig. 11. Illustration of successive learning of weak classifiers on tangent spaces

from neighborhoods on the manifold to the Euclidean space are homeomorphisms around the neighborhood of the points, the structure of the manifold is preserved locally in tangent spaces, thus, the classifiers can be trained on the tangent space at any point on the manifold. The mean of the points (16) minimizes the sum of squared distances on the manifold, therefore it is a good approximation up to the first order.

At each iteration, the weighted mean of the points where the weights are adjusted through boosting are computed. The points to the tangent space are mapped at the mean and a weak classifier on this vector space is learned. Since the weights of the samples which are misclassified during earlier stages of boosting increase, the weighted mean moves towards these points producing more accurate classifiers for these points (Figure 11). The approach minimizes the approximation error through averaging over several weak classifiers.

6.2 LogitBoost on Riemannian Manifolds

The probability of X being in class 1 is represented by

$$p(X) = \frac{e^{Z(X)}}{e^{Z(X)} + e^{-Z(X)}} \qquad Z(X) = \frac{1}{2}\sum_{l=1}^{L} z_l(X). \tag{35}$$

The LogitBoost algorithm learns the set of regression functions $\{z_l(X)\}_{l=1\ldots L}$ (weak learners) by minimizing the negative binomial log-likelihood of the data

$$-\sum_{i=1}^{N} [y_i \log(p(X_i)) + (1 - y_i)\log(1 - p(X_i))] \tag{36}$$

through Newton iterations. At the core of the algorithm, LogitBoost fits a weighted least square regression, $z_l(X)$ of training points $X_i \in \mathbb{R}^d$ to response values $\beta_i \in \mathbb{R}$ with weights w_i.

The LogitBoost algorithm on Riemannian manifolds is similar to original LogitBoost, except differences at the level of weak learners. In our case, the domain of the weak learners are in $\mathcal{M}$ such that $z_l(X) : \mathcal{M} \mapsto \mathbb{R}$. Following the discussion of the previous section, the regression functions are learned in the tangent space at the weighted mean of the points on the manifold. The weak learners are defined as

$$z_l(X) = v_l(\mathrm{vec}_{\bar{X}_l}(\log_{\bar{X}_l}(X))) \tag{37}$$

```
**Input:** Training set $\{(X_i, y_i)\}_{i=1...N}$, $X_i \in \mathcal{M}$, $y_i \in \{0,1\}$

– Start with weights $w_i = 1/N$, $i = 1...N$,
  $Z(X) = 0$ and $p(X_i) = \frac{1}{2}$
– Repeat for $l = 1...L$
    • Compute the response values and weights
      $\beta_i = \frac{y_i - p(X_i)}{p(X_i)(1-p(X_i))}$
      $w_i = p(X_i)(1 - p(X_i))$
    • Compute weighted mean of the points
      $\bar{X}_l = \arg\min_{Y \in \mathcal{M}} \sum_{i=1}^{N} w_i d^2(X_i, Y)$ (17)
    • Map the data points to the tangent space at $X_l$
      $x_i = \mathrm{vec}_{\bar{X}_l}(\log_{\bar{X}_l}(X_i))$
    • Fit the function $v_l(x)$ by weighted least-square regression of $\beta_i$
      to $x_i$ using weights $w_i$
    • Update $Z(X) \leftarrow Z(X) + \frac{1}{2} z_l(X)$ where $z_l$ is defined in (37)
      and $p(X) \leftarrow \frac{e^{Z(X)}}{e^{Z(X)} + e^{-Z(X)}}$
– Output the classifier sign
  $[Z(X)] = \text{sign } [\sum_{l=1}^{L} z_l(X)]$
```

Fig. 12. LogitBoost on Riemannian Manifolds

and learn the functions $v_l(x) : \mathbb{R}^d \mapsto \mathbb{R}$ and the weighted mean of the points $\bar{X}_l \in \mathcal{M}$. Notice that, the mapping vec (13), gives the orthogonal coordinates of the tangent vectors. For functions $\{v_l\}_{l=1...L}$, it is possible to use any form of weighted least squares regression such as linear functions, regression stumps, etc., since the domain of the functions are in $\mathbb{R}^d$.

6.3 Object Detection

Due to the articulated structure and variable appearance of the human body, illumination and pose variations, human detection in still images presents a challenge. For this task, $K = 30$ LogitBoost classifiers on $\mathbb{S}_8^+$ are combined with rejection cascade, as shown in Figure 13. Weak classifiers $\{v_l\}_{l=1...L}$ are linear regression functions learned on the tangent space of $\mathbb{S}_8^+$

$$\begin{bmatrix} m & n & |I_m| & |I_n| & \sqrt{I_m^2 + I_n^2} & |I_{mm}| & |I_{nn}| & \arctan \frac{|I_m|}{|I_n|} \end{bmatrix}^T \tag{38}$$

The covariance descriptor of a region is an 8×8 matrix and due to symmetry only upper triangular part is stored, which has only 36 different values. The tangent space is $d = 36$ dimensional vector space as well.

Let N_p and N_n be the number of positive and negative images in the training set. Since any detection window sampled from a negative image is a negative sample, it is possible to generate much more negative examples than the number of negative images. Suppose that the kth cascade level is being trained. All the possible detection windows on the negative training images are classified with the cascade of the previous $(k-1)$ LogitBoost classifiers. The samples which are misclassified form the possible negative set. Since the cardinality of the possible negative set is very large, examples from this set are sampled as the negative examples at cascade level k. At every cascade level, all the positive training images are considered as the positive training set.

A very large number of covariance descriptors can be computed from a single detection window and it is computationally intractable to test all of them. At each boosting iteration of kth LogitBoost level, subwindows are sampled, and normalized region covariance descriptors are constructed. The weak classifiers representing each subwindow are learned, and the best classifier which minimizes negative binomial log-likelihood (36) is added to the cascade level k.

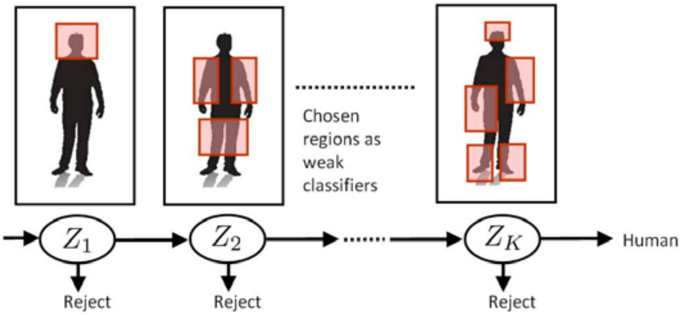


Fig. 13. Cascade of LogitBoost classifiers. The kth classifier selects normalized region covariance descriptors of the corresponding subregions.

Each level of cascade detector is optimized to correctly detect at least 99.8% of the positive examples, while rejecting at least 35% of the negative examples. In addition, a margin constraint between the positive samples and the decision boundary is enforced. Let $p_k(X)$ be the probability of a sample being positive at cascade level k, evaluated through (35). Let X_p be the positive example that has the $(0.998N_p)$th largest probability among all the positive examples. Let X_n be the negative example that has the $(0.35N_n)$th smallest probability among all the negative examples. Weak classifiers are added to cascade level k until $p_k(X_p) - p_k(X_n) > \tau$, where $\tau = 0.2$. When the constraint is satisfied, a new sample is classified as positive by cascade level k if $p_k(X) > p_k(X_p) - \tau > p_k(X_n)$ or equivalently $Z_k(X) > Z_k(X_n)$.

Since the sizes of the pedestrians in novel scenes are not known a priori, the images are searched at multiple scales. Utilizing the classifier trained on the INRIA dataset, sample detection examples for crowded scenes with pedestrians having variable illumination, appearance, pose and partial occlusion are shown in Figure 14.

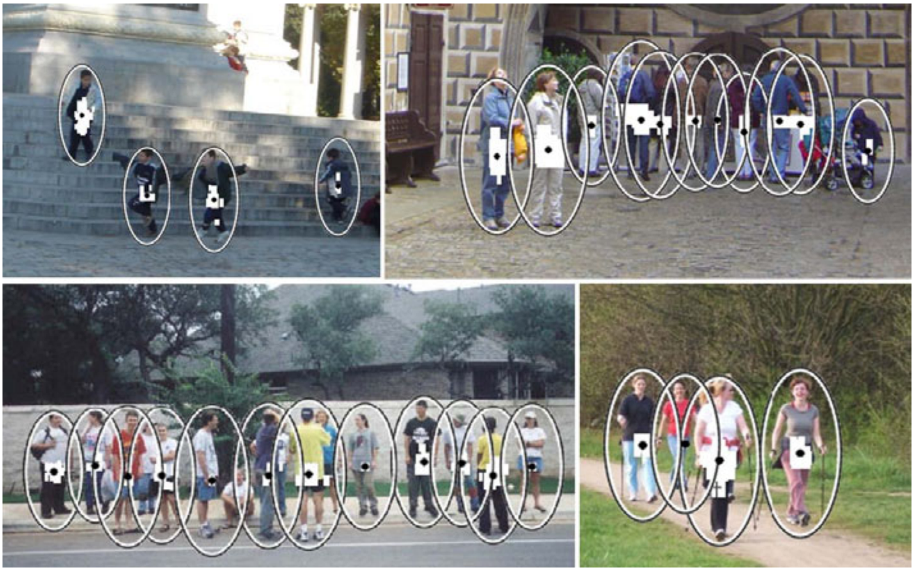


Fig. 14. Detection examples using cascade of LogitBoost classifiers on manifold. White dots show all the detection results. Black dots are the modes generated and the ellipses are average detection window sizes. There are extremely few false detections and misses.

7 Conclusions

Several parameter spaces that commonly occur in computer vision problems have Riemannian manifold structure including invertible affine transformations, nonzero quaternions with multiplication, general linear group (invertible square real matrices), real matrices with unit determinant, orientation-preserving isometries, real orthogonal matrices, and symplectic matrices.

Manifold based methods provide major improvements over the existing Euclidean techniques as demonstrated in this paper.

Acknowledgments

Thanks to Oncel Tuzel, Peter Meer and Pan Pan for their contributions and inspiring discussions.

References

1. Boothby, W.M.: An Introduction to Differentiable Manifolds and Riemannian Geometry, 2nd edn. Academic Press, London (1986)
2. Rossmann, W.: Lie Groups: An Introduction Through Linear Groups. Oxford Press (2002)

3. Forstner, W., Moonen, B.: A metric for covariance matrices. Technical report, Dept. of Geodesy and Geoinformatics, Stuttgart University (1999)
4. Pennec, X., Fillard, P., Ayache, N.: A Riemannian framework for tensor computing. International Journal of Computer Vision 66(1), 41–66 (2006)
5. Tuzel, O., Porikli, F., Meer, P.: Region covariance: A fast descriptor for detection and classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 589–600. Springer, Heidelberg (2006)
6. Porikli, F.: Integral histogram: A fast way to extract histograms in Cartesian spaces. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, vol. 1, pp. 829–836 (2005)
7. Georgescu, B., Shimshoni, I., Meer, P.: Mean shift based clustering in high dimensions: A texture classification example. In: Proc. 9th International Conference on Computer Vision, Nice, France, pp. 456–463 (2003)
8. Karcher, H.: Riemannian center of mass and mollifier smoothing. Commun. Pure Appl. Math. 30, 509–541 (1977)
9. Tuzel, O., Subbarao, R., Meer, P.: Simultaneous multiple 3D motion estimation via mode finding on Lie groups. In: Proc. 10th International Conference on Computer Vision, Beijing, China, vol. 1, pp. 18–25 (2005)
10. Hastie, T., Tibshirani, R., Freidman, J.: The Elements of Statistical Learning. Springer, Heidelberg (2001)
11. Miller, E., Chefd'hotel, C.: Practical non-parametric density estimation on a transformation group for vision. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (2003)
12. Porikli, F., Pan, P.: Regressed importance sampling on manifolds for efficient object tracking. In: Proc. 6th IEEE Advanced Video and Signal based Surveillance Conference (2009)
13. Viola, P., Jones, M.: Robust real-time object detection. International Journal of Computer Vision (2001)

Classification and Trees*

Luc Devroye

School of Computer Science, McGill University
Montreal, Canada H3A 2K6

This paper is dedicated to the memory of Pierre Devijver.

Abstract. Breiman, Friedman, Gordon and Stone recognized that tree classifiers would be very valuable to practicing statisticians. Their CART algorithm became very popular indeed. Designing tree-based classifiers, however, has its pitfalls. It is easy to make them too simple or too complicated so that Bayes risk consistency is compromised. In this talk, we explore the relationship between algorithmic complexity of tree-based methods and performance.

Extended Abstract

In scientific applications, the dual objective of a classification method is to classify and explain. It is this argument that makes partition methods interesting—these are methods in which the space is split up into disjoint sets. On each set, classification is performed by a simple majority vote. More formally, if $(X,Y) \in R^d \times \{0,1\}$ is the unknown underlying distribution of a datum (X) and its class (Y), and the data consist of independent identically distributed copies of (X,Y), namely $D_n = ((X_1,Y_1),\ldots,(X_n,Y_n))$, then a classifier is an estimator $g_n(X,D_n) \in \{0,1\}$ of Y, and the probability of error is

$$L_n = \mathbb{P}\{g_n(X,D_n) \neq Y|D_n\}.$$

The Bayes error $L^* = \inf_g \mathbb{P}\{g(X) \neq Y\}$ is the smallest error one can hope to obtain. If g were known, then we could consider consider the partition into $A = \{x \in R^d : g(x) = 1\}$ (the unknown Bayes discriminant set) and its complement. And indeed, most classifiers can be considered as partition classifiers for the partition (A_n, A_n^c), where $A_n = \{x \in R^d : g_n(x) = 1\}$, where A_n is an approximation of A.

What matters however is the explanatory aspect—how can the partition be described and constructed? For example, histograms based on regular grids could be considered, but they suffer from several drawbacks—first and foremost, they ignore the possible clustering in the data, and secondly, they can hardly be called instructional tools for explaining data. Thirdly, even modest dimensions quickly lead to histograms with underpopulated cells.

* The author's research was sponsored by NSERC Grant A3456.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 40–44, 2010.

This has led many researchers to consider smart and simple partitions. The linear discriminant, and the perceptron (Rosenblatt, 1962), are based upon partitions by hyperplanes. The question is whether we have universality, i.e., does $L_n \to L^*$ in probability as $n \to \infty$ for any distribution of (X, Y)? The answer is negative if one linear discriminant is used—how can it hope to get close to the unknown Bayes discriminant set A, which can be of arbitrary form? But the answer is affirmative, provided that linear discriminants are cleverly used as building blocks.

Buoyed by the intriguing comparison between brain function and learning machines, early methods of classification often involved combinations of linear discriminants. For example, in the committee machine (Nilsson, 1965), many linear discriminants are considered, each delivering a vote to each halfspace (to class "one" on one side, and to the "zero" class on the other side). For a particular X, its votes are totalled to make a final decision. Neural networks can be considered as smooth generalizations of this simple machine. Grid histograms can be considered as generalizations of committee machines in which all separating hyperplanes are aligned with the axes and regularly spaced—they have more degrees of freedom though. While not important in high dimensions, grid histograms do have one salient feature—they lead to universally consistent rules provided that the grid cell sizes shrink to zero with n and the average number of points per cell tends to infinity with n. The question then is whether committee machines are universally consistent. For example, we optimize k hyperplanes by minimizing the errors on the data, and if $k \to \infty$ and $k = o(n)$, then one would expect universal consistency. However, this is unknown (see Problem 30.6 in Devroye, Györfi and Lugosi, 1996).

Partitions based on hyperplanes are called arrangements. All arrangements in turn can be emulated by trees in which decisions are made by verifying signs of linear expressions. This leads naturally to tree classifiers. Each node in such a classifier makes a decision based on whether x is in certain set (ball, halfspace, simplex, hyperrectangle, and so forth) or not. Leaves in the tree correspond to sets in a partition.

Tree classifiers come in many flavors. One can cross-categorize by the style of partition. At the top of the list are the linear partitions perpendicular to the axes, which we shall call orthogonal cuts. They were preferred in the popular CART method (Breiman, Friedman, Olshen, Stone, 1984) because of the easy way in which classifiers can be explained, one variable (coordinate) at a time. Trees obtained by consecutive orthogonal cuts are called k-d trees in the computer science literature. Linear cuts lead to so-called hyperplane search trees—they too were proposed in the 1970s. Occasionally, one finds partitions by membership in simplices and rectangles.

More fundamental is the type of information used to create the tree-based partition. If only the X_i's are used, the partition can at best attempt to keep close points together, hoping that the joint distribution of (X, Y) shows some smoothness. Yet, ignoring the Y_i's has its advantages. For example, one obtains universal consistency under the following simple (and optimal) conditions. Let

C be the cell of the partition in which a random datum X falls, and let its diameter and cardinality be D and N, respectively. Then $D \to 0$ and $N \to \infty$ in probability suffice (Devroye, Györfi and Lugosi, 1996, p. 94). An example includes the median tree partition—split each coordinate in turn at the median of the X_i's, until each leaf cell has about k points. Then $k \to \infty$ and $k = o(n)$, and the existence of a density for X are the only conditions needed (Devroye, Györfi and Lugosi, 1996, p. 323).

However, ignoring the Y_i's is against human nature. For data on the real line ($d = 1$), there is an optimal binary split that minimizes the error on the data itself, which we shall call a Stoller (after Stoller, 1954). The Stoller split can be used in any direction, and indeed, one could consider the best linear or orthogonal split for $d > 1$. However, even today, we are missing simple theorems with easy-to-check conditions for universal consistency when trees, or partitions, are based on D_n in general. There are indeed many pitfalls that lead to inconsistency.

Thirdly, and perhaps most importantly, tree classifiers can be categorized by the algorithm used in their construction. Consider first top-down constructions, in which we keep splitting leaves until we are satisfied. One can optimize a criterion at each step, choosing a leaf that is most promising to split. However, selecting a leaf that yields the best orthogonal Stoller split at each step is not good enough—it is generally not consistent (Devroye, Györfi and Lugosi, 1996, p. 335). One can however, remedy this by finding the best Stoller split of a leaf using hyperrectangles as separators (Devroye, Györfi and Lugosi, 1996, chapter 20.13). This is the greedy approach to design. At the other end of the spectrum is the one-shot design of a tree of a given complexity (number of nodes) using optimization, such as minimization of the error on the data. This is phenomenally expensive, but its consistency is usually easy to guaranteee thanks to powerful inequalities for the empirical measures of sets initially derived by Vapnik and Chervonenkis (1971).

Bottom-up strategies first make a fine partition, e.g., a partition in which each final cell has one X_i. Then, in a second step, cells are recombined in a given fashion. CART follows this approach.

Ensemble methods, popular in machine learning, are learning algorithms that construct a set of many individual classifiers (called base learners) and combine them to classify new data points by taking a weighted or unweighted vote of their predictions. It is now well-known that ensembles are often much more accurate than the individual classifiers that make them up. The success of ensemble algorithms on many benchmark data sets has raised considerable interest in understanding why such methods succeed and identifying circumstances in which they can be expected to produce good results. These methods differ in the way the base learner is fit and combined. For example, bagging (Breiman, 1996) proceeds by generating bootstrap samples from the original data set, constructing a classifier from each bootstrap sample, and voting to combine. In boosting (Freund and Shapire, 1996) and arcing algorithms (Breiman, 1991) the successive classifiers are constructed by giving increased weight to those points that have been frequently misclassified, and the classifiers are combined using weighted voting. For a comprehensive review of ensemble methods, we refer the reader to Dietterich (2000).

Breiman (2001) provides a general framework for tree ensembles called "random forests". Each tree depends on the values of a random vector sampled independently and with the same distribution for all trees. Thus, a random forest is a classifier that consists of many decision trees and outputs the class that is the mode of the classes output by individual trees.

Random forests have been shown to give excellent performance on a number of practical problems. They work fast, generally exhibit a substantial performance improvement over single tree classifiers such as CART, and yield generalization error rates that compare favorably to the best statistical and machine learning methods.

Different random forests differ in how randomness is introduced in the tree building process, ranging from extreme random splitting strategies (Breiman (2000), Cutler and Zhao (2001)) to more involved data-dependent strategies (see Amit and Geman (1997), Breiman (2001), or Dietterich (2000)). Some attempts to investigate the consistency of random forests are by Breiman (2000, 2004), and Lin and Jeon (2006), who establish a connection between random forests and adaptive nearest neighbor methods.

Many examples of Breiman-style random forests are analyzed by Biau, Devroye and Lugosi (2008), and Biau and Devroye (2008). For example, sample k data from D_n uniformly at random, make a random k-d tree in a certain way and split until X has exactly one X_i in its cell. Record the vote Y_i. Now repeat many times and estimate Y by a majority rule. Without repetition, there is no consistency, but averaging the votes leads under some conditions to consistent rules.

References

[1] Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. Neural Computation 9, 1545–1588 (1997)

[2] Biau, G., Devroye, L., Lugosi, G.: Consistency of random forests and other averaging classifiers. Journal of Machine Learning Research 9, 2015–2033 (2008)

[3] Biau, G., Devroye, L.: On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. Technical Report (2008)

[4] Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)

[5] Breiman, L.: Arcing classifiers. The Annals of Statistics 24, 801–849 (1998)

[6] Breiman, L.: Some infinite theory for predictor ensembles. Technical Report 577, Statistics Department, UC Berkeley (2000), http://www.stat.berkeley.edu/~breiman

[7] Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001)

[8] Breiman, L.: Consistency for a simple model of random forests. Technical Report 670, Statistics Department, UC Berkeley (2004), http://www.stat.berkeley.edu/~breiman

[9] Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. CRC Press, Boca Raton (1984)

[10] Cutler, A., Zhao, G.: Pert – Perfect random tree ensembles. Computing Science and Statistics 33, 490–497 (2001)

[11] Devroye, L., Györfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer, New York (1996)

[12] Dietterich, T.G.: An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Machine Learning 40, 139–157 (2000)

[13] Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)

[14] Freund, Y., Shapire, R.: Experiments with a new boosting algorithm. In: Saitta, L. (ed.) Machine Learning: Proceedings of the 13th International Conference, pp. 148–156. Morgan Kaufmann, San Francisco (1996)

[15] Lin, Y., Jeon, Y.: Random forests and adaptive nearest neighbors. Journal of the American Statistical Association 101, 578–590 (2006)

[16] Nilsson, N.J.: Learning Machines: Foundations of Trainable Pattern Classifying Systems. McGraw-Hill, New York (1965)

[17] Rosenblatt, F.: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington (1962)

[18] Stoller, D.S.: Univariate two-population distribution-free discrimination. Journal of the American Statistical Association 49, 770–777 (1954)

[19] Vapnik, V.N., Chervonenkis, A.: On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its Applications 16, 264–280 (1971)

Structural Patterns in Complex Networks through Spectral Analysis

Ernesto Estrada

Department of Mathematics and Statistics, Department of Physics and
Institute of Complex Systems, University of Strathclyde, Glasgow,
G1 1XQ U.K.
ernesto.estrada@strath.ac.uk

Abstract. The study of some structural properties of networks is introduced from a graph spectral perspective. First, subgraph centrality of nodes is defined and used to classify essential proteins in a proteomic map. This index is then used to produce a method that allows the identification of superhomogeneous networks. At the same time this method classify non-homogeneous network into three universal classes of structure. We give examples of these classes from networks in different real-world scenarios. Finally, a communicability function is studied and showed as an alternative for defining communities in complex networks. Using this approach a community is unambiguously defined and an algorithm for its identification is proposed and exemplified in a real-world network.

Keywords: subgraph centrality, Estrada index, communicability, network communities.

1 Introduction

The study of complex networks has become a major field of interdisciplinary research in XXI century [1-3]. These networks are the skeleton of complex systems in a variety of scenarios ranging from social and ecological to biological and technological systems [4]. One of the main objectives of this research is the understanding of the structural organizational principles of such networks [5]. Network structure determines most -if not all- of network functions. Important dynamical processes taken place on networks are very much determined by their structural organization [6]. Then, some universal topological properties which explain some of the dynamical and functional properties of networks have been observed, such as ‘small-world’ [7] and ‘scale-free’ [8] phenomena. Despite the ubiquity of these phenomena in real-world systems, they have not been able to explain many of the structural and dynamical processes involving complex networks. Consequently, the search for other structural invariants that describe properties of complex networks in terms of structural parameters is needed. Among these other approaches spectral methods occupy an important place.

Spectral graph theory is a well established branch of the algebraic study of graphs [9]. Despite there are many results in this field they are mostly applicable to small graphs and not to gigantic complex networks having thousands or even millions of

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 45–59, 2010.

nodes. Without an excess of criticisms it can be said that many on the bounds found in spectral graph theory are very far from the real value when applied to large graphs, which make these approximation useless for practical purposes. On the other hand, the study of spectral properties of complex networks has been mainly focused to the study of the spectral density function in certain classes of random graphs [10-13]. This gives little information about the structure of real-world complex networks, which differ from random graphs in many structural characteristics.

Here we attack the problem from a different perspective. We attempt the definition of some spectral invariants for nodes and networks which give important structural information about the organization of these very large graphs. First, we study the characterization of local spectral invariants, in particular subgraph centrality [14] as a way for accounting for a 'mesoscale' characterization of nodes in a network. Using this concept we show analytically the existence of four universal topological classes of networks and give examples from the real-world about each of them [15, 16]. Finally, we study a communicability function [17] which allows to identify communities in complex networks [17, 18].

2 Background

We consider here networks represented by simple graphs $G := (V, E)$. That is, graphs having $|V| = n$ nodes and $|E| = m$ links, without self-loops or multiple links between nodes. Let $\mathbf{A}(G) = \mathbf{A}$ be the adjacency matrix of the graph whose elements A_{ij} are ones or zeroes if the corresponding nodes i and j are adjacent or not, respectively. A walk of length k is a sequence of (not necessarily different) vertices $v_0, v_1, \cdots, v_{k-1}, v_k$ such that for each $i = 1,2 \cdots, k$ there is a link from v_{i-1} to v_i. Consequently, these walks communicating two nodes in the network can revisit nodes and links several times along the way, which are sometimes called "backtracking walks." Walks starting and ending at the same node are named closed walks.

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ be the eigenvalues of the adjacency matrix in a non-increasing order and let $\varphi_j(p)$ be the pth entry of the jth eigenvector which is associated with the eigenvalue λ_j [9]. The number of walks $\mu_k(p, q)$ of length k from node p to q is given by

$$\mu_k(p,q) = \left(\mathbf{A}^k\right)_{pq} = \sum_{j=1}^{n} \varphi_j(p)\varphi_j(q)\lambda_j^k \ . \tag{1}$$

3 Local Patterns: Subgraph Centrality

A 'centrality' measure is a characterization of the 'importance' or 'relevance' of a node in a complex network. The best known example of node centrality is the "degree centrality", DC [4], which is interpreted as a measure of immediate influence of a node over its nearest neighbors. Several other centrality measures have been studied for real world networks, in particular for social networks. For instance, betweenness centrality (*BC*) measures the number of times that a shortest path between nodes i and j travels through a node k whose centrality is being measured. On the other hand, the farness of a node is the sum of the lengths of the geodesics to every other vertex. The reciprocal of farness is closeness centrality (*CC*). A centrality measure, which is not restricted to shortest paths [4], is defined as the principal or dominant eigenvector of the adjacency matrix A of a connected network. This centrality measure simulates a mechanism in which each node affects all of its neighbors simultaneously [4]. In fact, if we designate the number of walks of length L starting at node i by $N_L(i)$ and the total number of walks of this length existing in the network by $N_L(G)$. The probability that a walk selected at random in the network has started at node i is simply[19]:

$$P_L(i) = \frac{N_L(i)}{N_L(G)} \ . \tag{2}$$

Then, for non-bipartite connected network with nodes $1,2,\ldots,n$, it is known that for $L \to \infty$, the vector $[P_L(1) \quad P_L(2) \quad \cdots \quad P_L(n)]$ tends toward the eigenvector centrality of the network [19]. Consequently, the elements of *EC* represent the probabilities of selecting at random a walk of length L starting at node i when $L \to \infty$: $EC(i) = P_L(i)$.

If we compare degree and eigenvector centrality we can see that the first account for very local information about the interaction of a node and its nearest neighbors only. However, eigenvector centrality accounts for a more global environment around a node, which in fact includes all nodes of the network. Then, an intermediate characterization of the centrality of a node is needed in such a way that regions closest to the node in question make a larger contribution than those regions which are far apart from it. This sort of 'mesoscopic' type of centrality is obtained by considering the *subgraph centrality* of a node.

The subgraph centrality of a node is defined as the weighted sum of all closed walks starting and ending at the corresponding node [14]. If we designate by $\mu_k(i)$ the number of such closed walks of length k starting and ending at node i, the subgraph centrality is given by

$$EE(i) = \sum_{k=0}^{\infty} \frac{\mu_k(i)}{k!} \ , \tag{3}$$

where the factorial penalization guaranties that walks visiting nearest neighbors receive more weights than those visiting very distant nodes. It is straightforward to realize that the subgraph centrality of node i converges to the i th diagonal entry of the exponential of the adjacency matrix:

$$
\begin{aligned}
EE(i) &= (\mathbf{I})_{ii} + (\mathbf{A})_{ii} + \frac{\left(\mathbf{A}^2\right)_{ii}}{2!} + \frac{\left(\mathbf{A}^3\right)_{ii}}{3!} + \cdots + \frac{\left(\mathbf{A}^k\right)_{ii}}{k!} + \cdots \\
&= \left(\mathbf{I} + \mathbf{A} + \frac{\mathbf{A}^2}{2!} + \frac{\mathbf{A}^3}{3!} + \cdots + \frac{\mathbf{A}^k}{k!} + \cdots\right)_{ii} \\
&= \left(e^{\mathbf{A}}\right)_{ii} .
\end{aligned}
\tag{4}
$$

This index can be expressed in terms of the spectrum of the adjacency matrix of the corresponding network as [14]:

$$
EE(i) = \sum_{j=1}^{n} \left[\varphi_j(i)\right]^2 e^{\lambda_j} \ , \tag{5}
$$

where $\varphi_j(i)$ is the i th entry of the eigenvector associated with the j th eigenvalue λ_j of the adjacency matrix.

The subgraph centrality can be split into the contributions coming from odd and even closed walks as follows [20]:

$$
EE(i) = EE_{odd}(i) + EE_{even}(i) = \sum_{j=1} \left[\varphi_j(i)\right]^2 \sinh(\lambda_j) + \sum_{j=1} \left[\varphi_j(i)\right]^2 \cosh(\lambda_j), \tag{6}
$$

The sum of all subgraph centralities for the nodes of a network is known as the Estrada index of the graph and has been extensively studied in the mathematical literature (see [21, 22] and references therein):

$$
EE(G) = \sum_{i=1}^{n} EE(i) = tr\left(e^{A}\right) = \sum_{j=1}^{n} e^{\lambda_j} \ . \tag{7}
$$

In Fig. 1 we illustrate an example of the discriminant power of the subgraph centrality for the nodes of a graph. The graph illustrated in Fig. 1 displays the same degree, closeness and eigenvector centrality for all nodes. However, subgraph centrality identifies the three nodes at the top as the most central as they take part in triangles, while the others not. The second group of nodes according to their subgraph centrality is formed by two nodes taken place in no triangle but in three squares, while the least central nodes take part only in two squares but in no triangle.

A real-world example of the utility of centrality measures is provided by the identification of essential proteins in a protein-protein interaction (PPI) network. A PPI is a map of the physical interactions taken place between proteins in a cell. These interactions between proteins are responsible for many, if not all, biological functions of proteins in a cell. In every organism there are some proteins which are essential for the functioning of its cells. Knocking out these essential proteins produces the death of this organism. If such organism is a pathogenic one, then essential proteins are good targets for drugs attempting to kill such pathogen. Consequently, the *in silico* identification of essential proteins can play an important role in drug design by

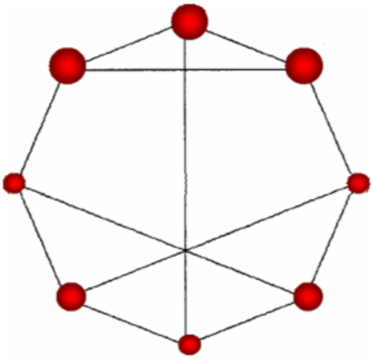


Fig. 1. Illustration of a simple graph in which all nodes have the same degree, closeness and eigenvector centralities. Subgraph centrality differentiates between three types of nodes, which are drawn with sizes proportional to $EE(i)$.

accelerating the process in which some protein targets are identified. Here an example is provided about the utility of centrality measures in identifying such essential proteins in the yeast PPI.

The PPI of *Saccharomyces cerevisiae* (yeast) was compiled by Bu *et al.* [23] from data obtained by von Mering *et al.* [24] by assessing a total of 80,000 interactions among 5400 proteins by assigning each interaction a confidence level. Here we study the main connected component of this network consisting of 2224 proteins sharing 6608 interactions. They were selected from 11,855 interactions between 2617 proteins with high and medium confidence in order to reduce the interference of false positives, from which Bu et al. [23] reported a network consisting of 2361 nodes and 6646 links (http://vlado.fmf.uni-lj.si/pub/networks/data/bio/Yeast/Yeast.htm). We illustrate the main connected component of this PPI in Fig. 2A.

In order to test the efficacy of different centrality measures in identifying essential proteins we ranked all proteins in the yeast PPI according to their subgraph, eigenvector, degree, closeness and betweenness centrality. Then, we select the top 5% of these proteins and analyze which of them has been reported experimentally as essential for yeast. As a null model we rank all proteins randomly, select the top 5% of these ranks and count the number of essential proteins. We take here the average of 1000 random realizations. In Fig. 2B we illustrate the results obtained by using this approach. As can be seen as average a random selection of proteins in yeast is able to identify only 25% of essential proteins. All centrality measures analyzed display significantly larger percentages of essential proteins identified than the random selection method. Both spectral methods used, the eigenvector centrality and the subgraph centrality, identify more than 50% of essential proteins in this proteome. In particular, subgraph centrality identifies 56.4% of essential proteins in the top 5% of the proteins [25]. In closing, centrality measures which are based only on topological information contained in the PPI network account for important biological information of yeast proteome.

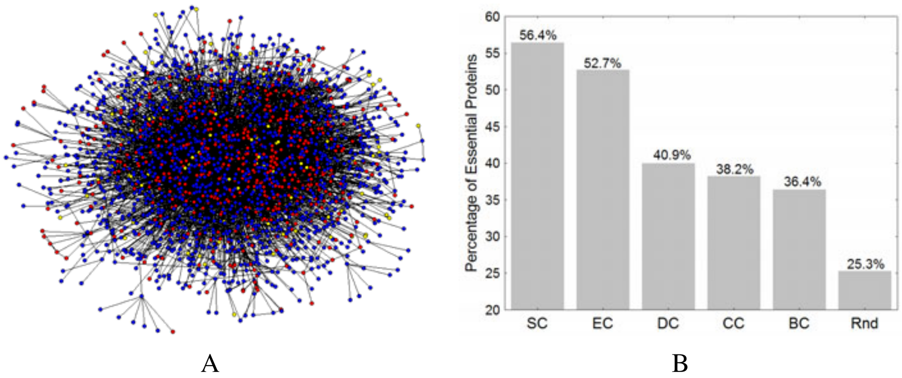


Fig. 2. (A) Illustration of the protein-protein interaction (PPI) network of yeast. Every node represents a protein and two nodes are linked if the corresponding proteins have been found to interact physically. Red nodes represent essential proteins, blue represent non-essential and yellow represent proteins with unknown essentiality. (B) Percentage of essential proteins identified by different centrality measures in the yeast PPI. SC, EC, DC, CC and BC stand for subgraph, eigenvector, degree, closeness and betweenness centrality, and Rnd stands for the average of 1000 random realizations.

4 Global Patterns

4.1 Structural Classes of Networks

The simplest class of networks we can consider is one consisting of very homogeneous structure. In these networks 'what you see locally is what you get globally'. Thus, describing the structure of a part of these networks gives an idea of their global topological structures. In order to have a quantitative criterion for classifying these networks we can consider a subset of nodes $S \subseteq V$ with cardinality $|S|$. Let $|\partial S|$ denotes the boundary of S, which is the number of links between a node in S and a node which is not in this set. Let us introduce the *expansion or isoperimetric constant* of the network as [26]:

$$\phi(G) = \inf\left\{\frac{|\partial S|}{|S|}, S \subseteq V, 0 < |S| \leq \frac{|V|}{2} < +\infty\right\}, \tag{8}$$

In a 'superhomogeneous' network as the ones described in the previous paragraph it is expected that $\phi(G) = O(1)$, which means that the number of links inside the subset S is approximately the same as the number of links going out from it for all the subsets $S \subseteq V$ in the network. This means that high expansion implies high homogeneity and better connectivity of the network, which means that the number of links that must be removed to separate the network into isolated chunks is relatively high in comparison with the number of nodes in the network.

A well-known result in spectral graph theory relates the expansion constant and the eigenvalues of the adjacency matrix. That is, if G is a regular graph with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$, then the expansion factor is bounded as [26],

$$\frac{\lambda_1 - \lambda_2}{2} \leq \phi(G) \leq \sqrt{2\lambda_1(\lambda_1 - \lambda_2)}, \tag{9}$$

which means that a network has good expansion if the gap between the first and second eigenvalues of the adjacency matrix ($\Delta\lambda = \lambda_2 - \lambda_1$) is sufficiently large. In closing, a superhomogeneous network, also known as expander, is characterized by a very large spectral gap $\Delta\lambda = \lambda_2 - \lambda_1$.

Let us consider what happen to the subgraph centralilty in these superhomogeneous networks. Without any loss of generality we study here the contribution of odd closed walks to the subgraph centrality $EE_{odd}(i)$. We can write the expression for the odd-subgraph centrality in the following way by noting that $EC(i) = \varphi_1(i)$

$$EE_{odd}(i) = [EC(i)]^2 \sinh(\lambda_1) + \sum_{j=2} [\varphi_j(i)]^2 \sinh(\lambda_j) , \tag{10}$$

Because the network we are considering here is a superhomogeneous one we can assume that $\lambda_1 >> \lambda_2$ in such a way that

$$[EC(i)]^2 \sinh(\lambda_1) >> \sum_{j=2} [\gamma_j(i)]^2 \sinh(\lambda_j) , \tag{11}$$

Consequently, in a superhomogeneous network we can approximate the odd-subgraph centrality as,

$$EE_{odd}(i) \approx [EC(i)]^2 \sinh(\lambda_1) , \tag{12}$$

which can be written as a straight line by applying logarithm as [15]:

$$\log[EC(i)] = \log A + \eta \log[EE_{odd}(i)], \tag{13}$$

where, $A \approx [\sinh(\lambda_1)]^{-0.5}$ and $\eta \approx 0.5$.

We have seen previously that eigenvector centrality is a characterization of a node environment that takes into account infinite walks visiting all nodes in the network. On the other hand, subgraph centrality is a mesoscopic characterization of the node environment giving a measure of the cliquishness of a close neighbourhood around it. Consequently, in a superhomogeneous network a log-log plot of $EC(i)$ *vs.* $EE_{odd}(i)$ displays a perfect straight line fit

$$\log EC^{Homo}(i) = 0.5 \log EE_{odd}(i) - 0.5 \log[\sinh(\lambda_1)] , \tag{14}$$

indicating a perfect scaling between local and global environment of a node. In other words, "what you see locally is what you get globally" in such networks. Deviations from *perfect superhomogeneity* can be accounted by measuring the departure of the points from the straight line respect to $\log EC^{Homo}(i)$ [16]:

$$\Delta \log EC(i) = \log \frac{EC(i)}{EC^{Homo}(i)} = \log \left\{ \frac{[EC(i)]^2 \sinh(\lambda_1)}{EE_{odd}(i)} \right\}^{0.5}, \tag{15}$$

Then, using (15) a network with $\Delta \log EC(i) \cong 0$ is classified as superhomogeneous. Other three classes can be identified, which correspond, respectively to the following cases [16]:

(i) $\Delta \log EC(i) \leq 0$ for all nodes: what you see locally is more densely connected that what you get globally, which indicates that the network contains 'holes' in its structure,

(ii) $\Delta \log EC(i) \geq 0$ for all nodes: what you see locally is less densely connected that what you get globally, which indicates the existence of a core-periphery structure of the network,

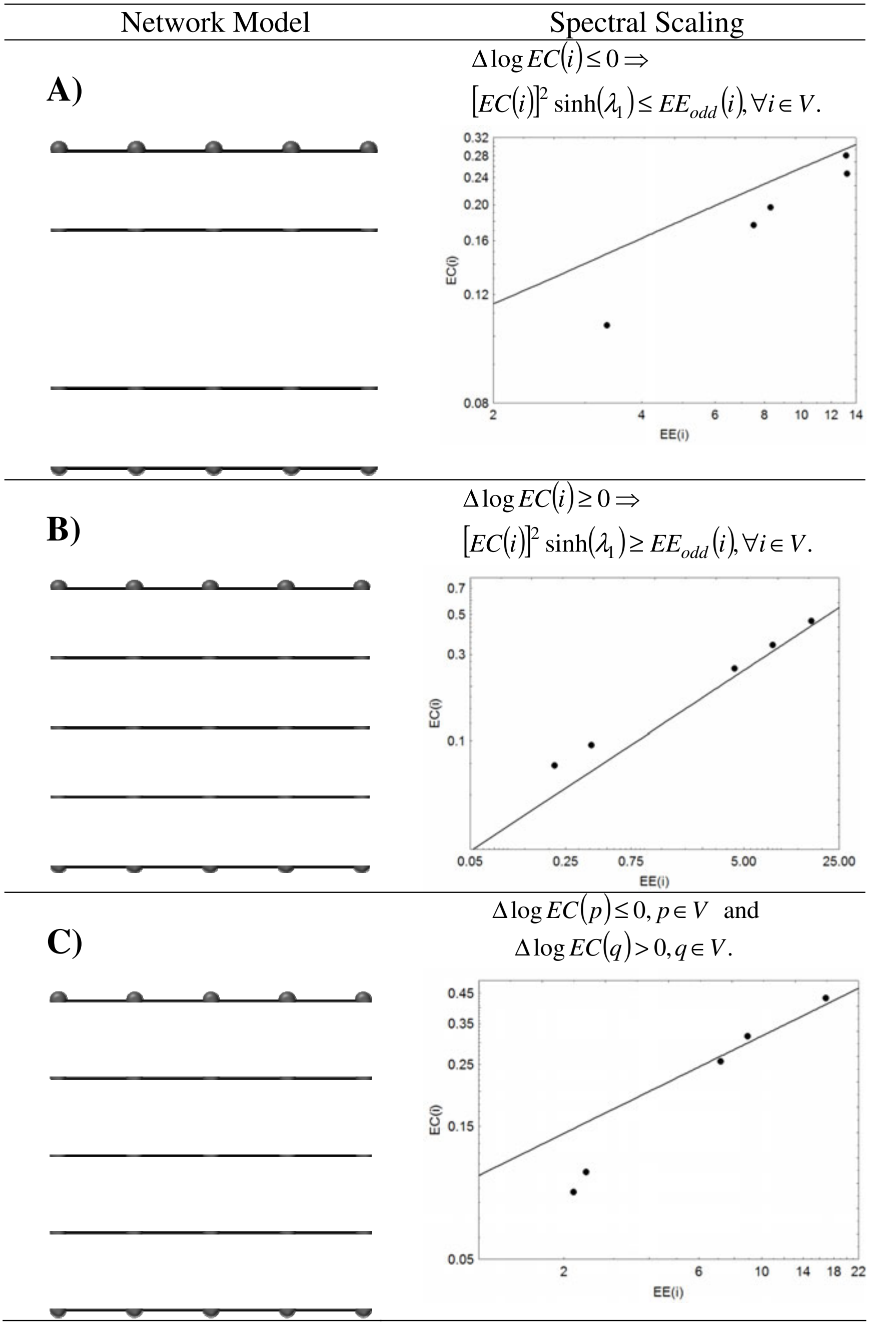
(iii) $\Delta \log EC(i) \leq 0$ for some nodes and $\Delta \log EC(i) > 0$ for the rest, which indicates the existence of a combination of the previous two patterns in a network.

The negative and positive deviations from the perfect scaling can be accounted by

$$\xi^+ = \sqrt{\frac{1}{N_+} \sum\nolimits_+ \left(\log \frac{\gamma_1(i)}{\gamma_1^{Ideal}(i)} \right)} \text{ and } \xi^- = \sqrt{\frac{1}{N_-} \sum\nolimits_- \left(\log \frac{\gamma_1(i)}{\gamma_1^{Ideal}(i)} \right)}$$

where $\sum_+$ and $\sum_-$ are the sums carried out for the N_+ points having $\Delta \log \gamma_1(i) > 0$ and for the N_- having $\Delta \log \gamma_1(i) < 0$, respectively. In Fig. 3 we illustrate these three patterns of networks together with their spectral scaling.

In Fig. 4 we illustrate one example of each of the four structural patterns found in complex networks. The first network is a 1997 version of Internet at autonomous system, which displays a large homogeneity as can be seen in the perfect spectral scaling given in the same figure. The negative and positive deviations from perfect scaling for this network are 6.21×10^{-4} and 1.20×10^{-3}, respectively. The second network corresponds to the residue-residue interaction network in the protein with Protein Data Bank code (1ash), which corresponds to the structure of *Ascaris suum* hemoglobin domain I at 2.2 angstroms resolution. This network corresponds to the class of positive deviations from perfect scaling, which indicates the presence of structural holes in its structure. These structural holes correspond to the cavities protein structures have, which in many cases display biological functionality and represent important binding sites for proteins [27]. The third network correspond to the food web of Canton Creek, which is primarily formed by trophic interactions between invertebrates and algae in a tributary, surrounded by pasture, of the Taieri River in the South Island of New Zealand. This network is characterized by a central core of species with a large number of interactions among them and a periphery of species which weakly interact to each other and with the central core. The final network represents social ties in a karate club in USA, which eventually polarizes into two factions due to an internal conflict. It is characterized by two main clusters or communities, followers of the administrator and followers of the trainer in which intersection the presence of holes is observed. At the same time each of the two clusters form some small core-periphery structure giving rise to the spectral scaling observed.



A)

$$\Delta \log EC(i) \leq 0 \Rightarrow$$
$$[EC(i)]^2 \sinh(\lambda_1) \leq EE_{odd}(i), \forall i \in V.$$

B)

$$\Delta \log EC(i) \geq 0 \Rightarrow$$
$$[EC(i)]^2 \sinh(\lambda_1) \geq EE_{odd}(i), \forall i \in V.$$

C)

$$\Delta \log EC(p) \leq 0, p \in V \quad \text{and}$$
$$\Delta \log EC(q) > 0, q \in V.$$

Fig. 3. Illustration of the three patterns of networks that deviate from perfect spectral scaling. The spectral scaling is a log-log plot of the eigenvector centrality, *EC(i)* versus subgraph centrality, *EE(i)* for all nodes in the graph.

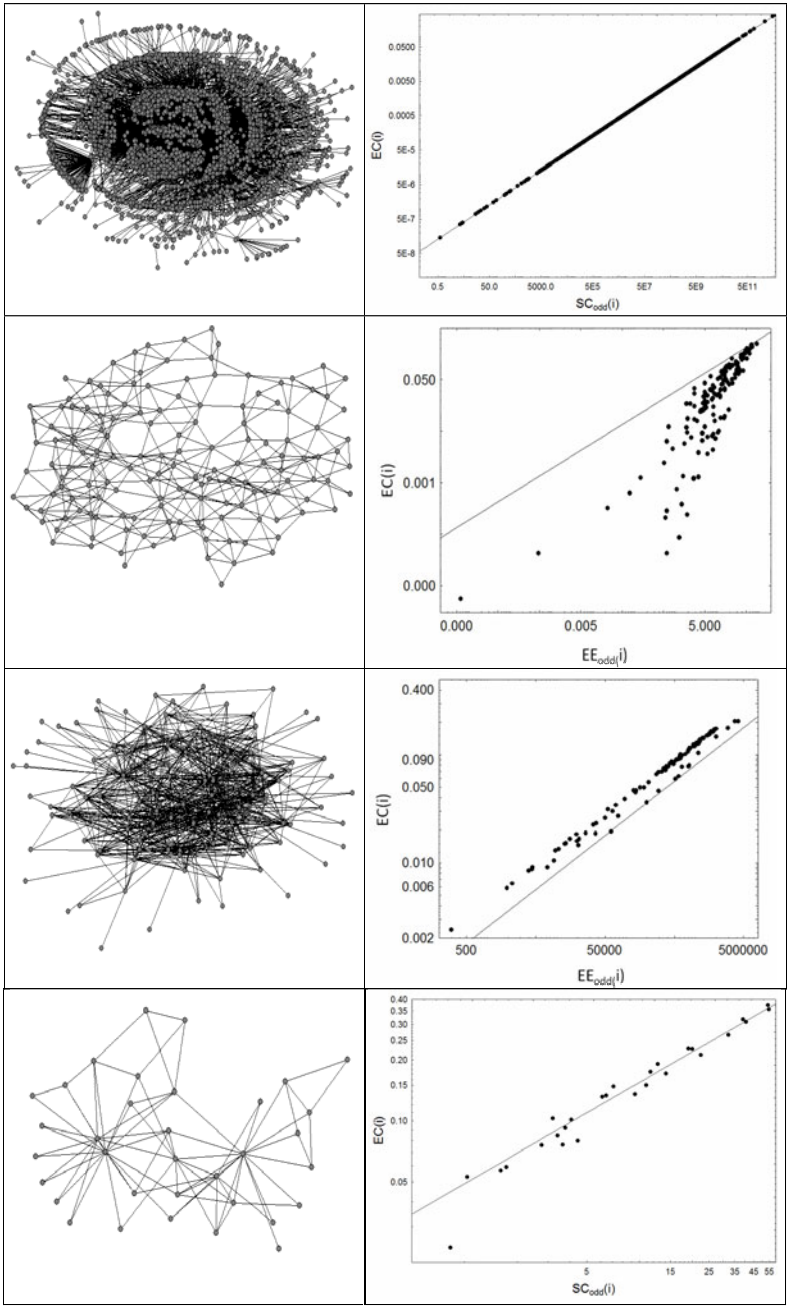


Fig. 4. Illustration of the four structural patterns in real-world complex networks. The first corresponds to Internet autonomous system in 1997, the second is the protein residue network of 1ASH, the third represents a food web of Canton Creek and the fourth corresponds to a social network of friendship ties in a karate club.

A characteristic feature of all networks which are not superhomogeneous is that nodes can be grouped together in certain clusters or communities. These communities can play an important role in understanding the structure and dynamics of complex networks in different scenarios. There are several approaches to detect communities in networks which are used today [28]. In the following section we explain one which is based on the concept of communicability between nodes in a network.

4.2 Communicability and Communities in Networks

In continuation with the line we have followed in the previous sections we define the communicability between a pair of nodes in a network as follows [17]:

> The communicability between a pair of nodes p,q in a network is a weighted sum of all walks starting at node p and ending at node q, giving more weight to the shortest walks.

This definition accounts for the known fact that in many situations the communication between a pair of nodes in a network does not take place only through the shortest path connecting them. A mathematical formulation of this concept is obtained by considering the sum of all walks of different lengths that connect nodes p and q [17]:

$$G_{pq} = \sum_{k=0}^{\infty} \frac{\left(\mathbf{A}^k\right)_{pq}}{k!} = \left(e^{\mathbf{A}}\right)_{pq} , \tag{16}$$

which can be expressed in terms of the eigenvalues and eigenvectors of the adjacency matrix as follows

$$G_{pq} = \sum_{j=1}^{n} \varphi_j(p)\varphi_j(q)e^{\lambda_j} . \tag{17}$$

The detection of communities by using the communicability function is based on the analysis of the sign of the term $\varphi_j(p)\varphi_j(q)e^{\lambda_j}$, which can be either positive or negative on the basis of the signs of the pth and qth components of the corresponding eigenvector. We can think that the eigenvectors of the adjacency matrix represent vibrational normal modes of the network. The sign of the pth component of the jth eigenvector indicates the direction of the vibration. If two nodes, p and q, have the sign for the jth eigenvector it indicates that these two nodes are vibrating in the same direction. As we have previously seen all entries of the principal eigenvector φ_1 have the same sign. Consequently, we consider it as a translational movement of the whole network. Then, we can divide the communicability function into three contributions [17]:

$$G_{pq} = \left[\varphi_1(p)\varphi_1(q)e^{\lambda_1}\right] + \sum_{2 \le j \le n}^{\text{intra-cluster}} \varphi_j(p)\varphi_j(q)e^{\lambda_j} + \sum_{2 \le j \le n}^{\text{inter-cluster}} \varphi_j(p)\varphi_j(q)e^{\lambda_j} \tag{18}$$

where the term ‘intra-cluster’ refers to the sum over all components $\varphi_j(p)$ and $\varphi_j(q)$ having the same sign. The ‘inter-cluster’ term refer to the case when $\varphi_j(p)$ and $\varphi_j(q)$ have different signs. Note that the last term, i.e., the ‘inter-cluster’ communicability is negative. Then, as we are interested in partitioning the network into communities or clusters we simply subtract the translational contribution to obtain the difference between intra- and inter-cluster communicability [17]:

$$\Delta G_{pq} = \sum_{j=2}^{\text{intra-cluster}} \varphi_j(p)\varphi_j(q)e^{\lambda_j} - \left| \sum_{j=2}^{\text{inter-cluster}} \varphi_j(p)\varphi_j(q)e^{\lambda_j} \right| . \tag{19}$$

Note that for computing (19) it is not necessary to make any sign analysis of the eigenvectors of the network. It is enough to compute the communicability between two nodes and then subtract the translational term, i.e., $\Delta G_{pq} = G_{pq} - \varphi_1(p)\varphi_1(q)e^{\lambda_1}$. Now, we can define a community in a network as follows [18]:

> A network community is a group of nodes $C \subseteq V$ for which the intra-cluster communicability is larger than the inter-cluster one: $\Delta G_{p,q}(\beta) > 0 \quad \forall (p,q) \in C$.

In practice, in order to find such communities we represent the values of $\Delta G_{p,q}$ between pairs of nodes as a matrix $\varDelta(G)$ and then we dichotomize such matrix, such that the p,q entry of the new matrix is 1 if, and only if $\Delta G_{p,q} > 0$ and zero otherwise. This new matrix can be considered as the adjacency matrix of a new graph, which we call the *communicability graph* $\Theta(G)$ [18]. The nodes of $\Theta(G)$ are the same as the nodes of G, and two nodes p and q in $\Theta(G)$ are connected if, and only if, $\Delta G_{p,q} > 0$ in G. Finally, a community is identified as a clique in the communicability graph [18].

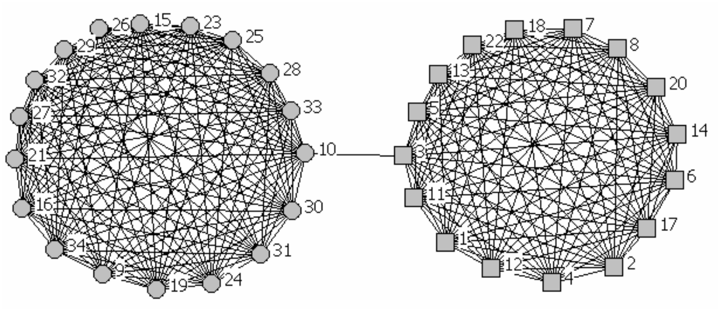


Fig. 5. Communicability graph for the network of friendship ties in a karate club. Circles and squares are used to represent the two known communities existing in this network as a consequence of its polarization as followers of the administrator and followers of the trainer.

As an example we illustrate in Fig. 5 the communicability graph for the social network of friendship ties in a karate club given in Fig. 4. The analysis of the cliques in this communicability graph indicates the existence of 5 overlapped communities, which are given below:

C_1 : {10,15,16,19,21,23,24,26,27,28,29,30,31,32,33,34};
C_2 : {9,10,15,16,19,21,23,24,27,28,29,30,31,32,33,34};
C_3 : {10,15,16,19,21,23,24,25,26,27,28,29,30,32,33,34};
C_4 : {1,2,3,4,5,6,7,8,11,12,13,14,17,18,20,22};
C_5 : {3,10}.

The overlap between two communities C_i and C_j can be computed by using an appropriate index $S_{C_iC_j}$ [18]. Then, communities can be merged together according to a given mergence parameter α in such a way that a new matrix is created according to

$$O_{C_iC_j} = \begin{cases} 1 \text{ if } S_{C_iC_j} \geq \alpha \\ 0 \text{ if } S_{C_iC_j} < \alpha, \text{ or } C_i = C_j \end{cases},$$

and the process is finished when no pair of communities have overlap larger than α. Applying this criterion with $\alpha = 0.5$ the following two communities are obtained for the previously studied network: $U_1 = C_1 \cup C_2 \cup C_3 \cup C_5$ and $U_2 = C_4 \cup C_5$, which are the two communities observed experimentally for this network.

5 Conclusions

The study of spectral invariants is an interesting alternative for characterizing the structure and properties of complex networks. We have studied here some invariants which are based on the concept of walks in networks and its relation with eigenvalues and eigenvectors of the adjacency matrices of such networks. Subgraph centrality, spectral scaling and communicability are three of these measures characterizing local or global properties of networks. Similar concepts have been extended to study betweenness [29], bipartitions [30], as well as to account for other matrix functions [31]. Recently, subgraph centrality has been used to study [32] the topological evolution in dense granular materials. It proved to be a good indicator of the topological dynamic in such materials with very good correlation with the constitutive properties of nonaffine deformation and dissipation, spatially and with respect to strain. On the other hand, communicability was used as a classifier in human brain networks [33]. In this work two groups of brain networks are studied, one group corresponds to healthy humans and the other to patients who suffer stroke in the last six months. The discriminating power of the communicability function of a normalized weighted matrix was higher than other spectral methods for differentiating between the two studied groups. All these examples show the versatility of these spectral measures for studying the structure and properties of complex networks.

Acknowledgments. This work is partially funded by the Principal of the University of Strathclyde through the New Professors' Fund.

References

1. Strogatz, S.H.: Exploring complex networks. Nature 410, 268–276 (2001)
2. Albert, R., Barabási, A.-L.: Statistical mechanics of complex networks. Rev. Mod. Phys. 74, 47–97 (2002)
3. Newman, M.E.J.: The structure and function of complex networks. SIAM Rev. 45, 167–256 (2003)
4. Newman, M.E.J.: Networks. An Introduction. Oxford University Press, Oxford (2010)
5. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.-U.: Complex Networks: Structure and Dynamics. Phys. Rep. 424, 175–308 (2006)
6. Barrat, A., Barthélemy, M., Vespignani, A.: Dynamical Processes on Complex Networks. Cambridge University Press, Cambridge (2008)
7. Watts, D.J., Strogatz, S.H.: Collective dynamics of "small-world" networks. Nature 393, 440–442 (1998)
8. Barabási, A.-L., Albert, R.: Emergence of scaling in complex networks. Science 286, 509–512 (1999)
9. Cvetković, D., Rowlinson, P., Simić, S.: An Introduction to the Theory of Graph Spectra. Cambridge University Press, Cambridge (2010)
10. Farkas, I.J., Derényi, I., Barabási, A.-L., Vicsek, T.: Spectra of "Real–World" Graphs: Beyond the Semi-Circle Law. Phys. Rev. E 64, 26704 (2001)
11. Goh, K.-I., Kahng, B., Kim, D.: Spectra and eigenvectors of scale-free networks. Phys. Rev. E 64, 051903 (2001)
12. Dorogovtsev, S.N., Goltsev, A.V., Mendes, J.F.F., Samukhin, A.N.: Spectra of complex networks. Phys. Rev. E 68, 46109 (2003)
13. De Aguiar, M.A.M., Bar-Yam, Y.: Spectral analysis and the dynamic response of complex networks. Phys. Rev. E 71, 16106 (2005)
14. Estrada, E., Rodríguez-Velázquez, J.A.: Subgraph centrality in complex networks. Phys. Rev. E 71, 56103 (2005)
15. Estrada, E.: Spectral scaling and good expansion properties in complex networks. Europhys. Lett. 73, 649–655 (2006)
16. Estrada, E.: Topological Structural Classes of Complex Networks. Phys. Rev. E 75, 016103 (2007)
17. Estrada, E., Hatano, N.: Communicability in complex networks. Phys. Rev. E 77, 36111 (2008)
18. Estrada, E., Hatano, N.: Communicability Graph and Community Structures in Complex Networks. Appl. Math. Comput. 214, 500–511 (2009)
19. Cvetković, D., Rowlinson, P., Simić, S.: Eigenspaces of Graphs. Cambridge University Press, Cambridge (1997)
20. Estrada, E., Rodríguez-Velázquez, J.A.: Spectral measures of bipartivity in complex networks. Physical Review E 72, 046105 (2005)
21. Deng, H., Radenković, S., Gutman, I.: The Estrada Index. In: Cvetković, D., Gutman, I. (eds.) Applications of Graph Spectra, pp. 123–140. Math. Inst., Belgrade (2009)
22. Benzi, M., Boito, P.: Quadrature rule-based bounds for functions of adjacency matrices. Lin. Alg. Appl. 433, 637–652 (2010)

23. Bu, D., Zhao, Y., Cai, L., Xue, H., Zhu, X., Lu, H., Zhang, J., Sun, S., Ling, L., Zhang, N., Li, G., Chen, R.: Topological structure analysis of the yeast protein-protein interaction network of budding yeast. Nucl. Ac. Res. 31, 2443–2450 (2003)
24. von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S.G., Field, S., Bork., P.: Comparative assessment of large-scale data sets of protein-protein interactions. Nature 417, 399–403 (2002)
25. Estrada, E.: Virtual identification of essential proteins within the protein interaction network of yeast. Proteomics 6, 35–40 (2006)
26. Hoory, S., Linial, N., Wigderson, A.: Expander graphs and their applications. Bull. Am. Math. Soc. 43, 439–561 (2006)
27. Estrada, E.: Universality in Protein Residue Networks. Biophys. J. 98, 890–900 (2010)
28. Fortunato, S.: Community detection in graphs. Phys. Rep. 486, 75–174 (2010)
29. Estrada, E., Higham, D.J., Hatano, N.: Communicability betweenness in complex networks. Physica A 388, 764–774 (2009)
30. Estrada, E., Higham, D.J., Hatano, N.: Communicability and multipartite structure in complex networks at negative absolute temperatures. Phys. Rev. E 78, 026102 (2008)
31. Estrada, E.: Generalized walks-based centrality measures for complex biological networks. J. Theor. Biol. 263, 556–565 (2010)
32. Walker, D.M., Tordesillas, A.: Topological evolution in dense granular materials: A complex networks perspective. Int. J. Sol. Struct. 47, 624–639 (2010)
33. Crofts, J.J., Higham, D.J.: A weighted communicability measure applied to complex brain networks. J. Roy. Soc. Interface 33, 411–414 (2009)

Graph Embedding Using an Edge-Based Wave Kernel

Hewayda ElGhawalby[1,2] and Edwin R. Hancock[1,⋆]

[1] Department of Computer Science, University of York,
YO10 5DD, UK
[2] Faculty of Engineering, Suez Canal university, Egypt
{howaida,erh}@cs.york.ac.uk

Abstract. This paper describes a new approach for embedding graphs on pseudo-Riemannian manifolds based on the wave kernel. The wave kernel is the solution of the wave equation on the edges of a graph. Under the embedding, each edge becomes a geodesic on the manifold. The eigensystem of the wave-kernel is determined by the eigenvalues and the eigenfunctions of the normalized adjacency matrix and can be used to solve the edge-based wave equation. By factorising the Gram-matrix for the wave-kernel, we determine the embedding co-ordinates for nodes under the wave-kernel. We investigate the utility of this new embedding as a means of gauging the similarity of graphs. We experiment on sets of graphs representing the proximity of image features in different views of different objects. By applying multidimensional scaling to the similarity matrix we demonstrate that the proposed graph representation is capable of clustering different views of the same object together.

Keywords: Wave Equation, Pseudo Riemannian manifolds, Edge-based Laplacian, Graph Embedding.

1 Introduction

Graph embeddings have found widespread use in machine learning and pattern recognition for the purposes of clustering, analyzing and visualizating relational data. However, they have also proved to be useful as a means of graph characterization. There are many examples in the literature including ISOmap [13], the Laplacian eigenmap [1], and the heat-kernel embedding [16], to name a few. Once embedded, a graph can be characterised using a feature-vector that characterises the point-set distribution resulting from the embedding [15]. This kind of representation is convenient since a Euclidean vector space makes available powerful geometric analysis tools for data analysis, not available for discrete or structural representations. However, such an embedding assumes that the original relational data is metric. Sometimes, however, this is not the case. This is the case when the matrix characterisation of the relational graph contains negative

[⋆] This work was supported by the EU FET project SIMBAD (213250). Edwin Hancock was supported by a Royal Society Wolfson Research Merit Award.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 60–69, 2010.

eigenvalues, i.e. it is not positive semi-definite. Under these circumstances the graph embeds not into a Euclidean space, but into pseudo-Euclidean or Krein space [12]. This problem has attracted relatively little attention in the literature. Our aim in this paper is to embed the nodes of a graph as points on the surface of a pseudo-Riemannian manifold in a pseudo-Euclidean space, and to use the resulting point-set as the basis from which to compute graph characteristics. To provide a framework for our study, we turn to the wave kernel. This is the solution of a wave equation, which is an important second-order linear partial differential equation that describes the propagation of a variety of waves. Crucially, the solutions are complex and therefore reside in a pseudo-Euclidean space. Although the wave equation has been extensively studied in the continuous domain, there has been relatively little effort devoted to understanding its behavior on a graph. In common with the heat kernel, the wave kernel can be defined in terms of a combinatorial Laplacian. However, in the case of the wave kernel this is the edge-based Laplacian, introduced by Friedman [6].

In this paper we explore how to solve the edge-based wave equation, in terms of the eigensystem of the edge-based Laplacian. Since the solution is a sinusoid, it contains both real and imaginary parts. Hence, we embed the nodes of the graph as points residing on a pseudo-Riemannian manifold, determined by the eigenvalues and eigenvectors of the edge-based Laplacian. In our experiments on graphs extracted from 2D image data, we use this matrix for the purpose of graph visualization. The remainder of this paper is organized as follows: In Section 2 we commence by embedding graphs onto pseudo Riemannian manifolds. First we show how to find the solution of the wave equation on a graph using the edge-based Laplacian in §2.1. Then we construct the coordinate matrix for the pseudo-Euclidean embedding in §2.2. Finally, §2.3 is devoted to establishing the edge-based Laplacian matrix. In Section 3 we illustrate how to manipulate vectors in a pseudo Euclidean space, commencing by computing the square distance between any arbitrary pair of vectors in §3.1. In §3.2 we show how to construct an orthonormal basis, and in §3.3 how to project vectors from a pseudo Euclidean space onto a 2D sub-space. Section 4 gives a brief review for the Hausdorff distance as a tool of comparing sets of unordered observations resulting from the embedding of the graphs. Section 5 presents our experimental evaluaton. Finally, conclusions are drawn and future directions of research suggested in Section 6.

2 Embedding Graphs into Pseudo Riemannian Manifolds

2.1 Edge-Based Wave Equation

Friedman [6] has developed a graph-based version of the wave equation that has many of the properties of the classical Laplacian wave equation. The development is based on a variant of the combinatorial Laplacian referred to as the edge-based Laplacian. This graph theoretic version of the wave equation provides an interesting link with the continuous wave eqaution, and has a simple physical interpretation. The edges of the graph can be viewed as taut strings, joined together at the vertices. In fact, the edge-based Laplacian has been shown in the physics literature to be the "limiting case" of a "quantum wire" [8].

Graph theory defines a combinatorial Laplacian, L, as an operator on functions whose domain is the set of vertices of a graph. On the vertex-set the wave equation is $U_{tt} = -LU$ (the negative sign is due to that the combinatorial Laplacians are positive semi-definite). However, this wave equation fails to give a finite speed for wave propagation. As a result there is no simple link between the graph theoretic wave equation and its continuous counterpart. To overcome this problem a so-called edge-based wave equation $W_{tt} = -L_E W$ was introdiced by Friedman [6], where L_E is the edge-based Laplacian, which is a better approxomation to the continuous Laplacian (i.e. the second derivative) than the combinatorial Laplacian L. The edge-based wave equation has unit wave propagation speed, while that based on the combinatorial Laplacian L has infinite speed.

For the edge-based Laplacian, the eigenfunction f satisfies $L_E f = \lambda f$ and $Lf = 0$ where $\Lambda_E = \{\lambda\}$ is the set of Laplacian eigenvalues. In fact, if L is normalized and the graph under study has each edge weight equal to unity, then L is similar to $\left(1 - \cos\sqrt{L_E}\right)$. That is to say if Δ is a continuous Laplacian then $\widetilde{\Delta} = 1 - \cos\sqrt{-\Delta}$ is the corresponding combinatorial Laplacian. Hence, the eigenvalue λ is in Λ_E if and only if $\left(1 - \cos\sqrt{\lambda}\right)$ is in Λ (the set of all eigenvalues of the combinatorial Laplacians). Note that Λ_E is an infinite set of non-negative values (whose square roots are periodic), and exclude those which are multiples of π from Λ_E. Recall that the general solution of the wave equation

$$\begin{aligned} W_{tt} &= -L_E W \\ W|_{t=0} &= \quad f \\ W_t|_{t=0} &= \quad g \end{aligned} \tag{1}$$

has the form

$$W = \frac{\sin\left(\sqrt{L_E}t\right)}{\sqrt{L_E}} g + \cos\left(\sqrt{L_E}t\right) f \tag{2}$$

For our work it suffices to compute the fundamental solution W that satisfies $W|_{t=0} = 0$ and $W_t|_{t=0} = 1$, that is

$$W = \frac{\sin\sqrt{L_E}t}{\sqrt{L_E}} \tag{3}$$

Since, L_E is positive semi-definite [6], W can be approximated using the MacLaurin series, giving

$$W = t[I - \frac{1}{6} L_E t^2] \tag{4}$$

Now we can consider the nodes of the graph as residing on a pseudo-Riemannian manifold and the edges as geodesics on the manifold.

2.2 The Embedded Coordinates Matrix

Positive definite Riemannian manifolds can be represented in one of two ways. Either a) their properties are defined intrinsically, or b) they can be regarded as subsets of a Euclidean space of higher dimension. Following the work of Nash [10,11] and Whitney [14], it has been known for some time that these approaches are equivalent, in the sense that any intrinsically defined Riemannian manifold can be embedded, with appropriate differentiability, into a Euclidean space. In [2], Clarke showed that the same situation holds in the case of pseudo-Riemannian manifolds, with metrics of indefinite signature.

The pseudo-Euclidean space generalizes the well-known Euclidean space to the case where inner products are indefinite. This effectively amounts to two Euclidean spaces, one of which has a positive semi-definite inner product and the second with a negative semi-definite inner product. For squared Euclidean distances, the embedding is determined by the pseudo Gram matrix $C = -\frac{1}{2}QWQ$ derived from the kernel matrix W, where $Q = ee^T - I$and $e = (1, ..., 1)^T$. If the matrix with the embedding co-ordinates as columns is X^T, then

$$C = -\frac{1}{n}XX^T \tag{5}$$

In the pseudo Euclidean space

$$C = -\frac{1}{n}X \begin{pmatrix} M & 0_{(p+q)\times k} \\ 0_{k\times(p+q)} & 0_{k\times k} \end{pmatrix} X^T \tag{6}$$

where

$$M = \begin{pmatrix} I_{p\times p} & 0_{p\times q} \\ 0_{q\times p} & -I_{q\times q} \end{pmatrix} \tag{7}$$

and $0_{k\times k}$ is the $k \times k$ matrix filled with zeros, and $p + q + k = n$. We can then write $XMX^T = \Phi\Lambda\Phi^T = \Phi|\Lambda|^{\frac{1}{2}}M|\Lambda|^{\frac{1}{2}}\Phi^T$, where Φ is the column-matrix of the eigenvectors and Λ the diagonal matrix of the corresponding eigenvalues. The vectors are recovered via the transformation $X_L = \Phi_L|\Lambda_L|^{\frac{1}{2}}$, where Φ_L is the column-matrix of the selected eigenvectors and Λ_L the diagonal matrix of the corresponding eigenvalues. Hence, the columns of X_L are the vectors in the pseudo-Euclidean space.

2.3 Edge-Based Eigenvalues and Eigenfunctions

Before we experiment with our embedding, we need first to construct the edge-based Laplacian matrix. We follow the procedure given in [6] where the edge-based eigenvalues and eigenfunctions are determined using those of a normalized adjacency matrix. To commence, consider a finite graph denoted by $G = (V, E)$ with node-set V and edge-set of edges $E \subseteq V \times V$, with all edges of unit weight. The elements of the adjacency matrix A of the graph G are

$$A(u, v) = \begin{cases} 1 & if(u, v) \in E \\ 0 & otherwise \end{cases} \tag{8}$$

Let T be a diagonal matrix whose elements are the degrees of the nodes of G, that is $T(u,u) = \sum_{v \in V} A(u,v) = deg_u$. By dividing each row of the adjacency matrix A by its corresponding deg_u, we obtain the normalized adjacency matrix $\widetilde{A}$. For each eigenvalue, λ of $\widetilde{A}$ there is a unique value of $\cos^{-1}(\lambda) \in [0,\pi]$. The edge-based eigenvalues are $2n\pi + \cos^{-1}(\lambda)$ and $2(n+1)\pi - \cos^{-1}(\lambda)$, where $\{n = 0,1,2,...\}$. Hence, if $\omega \in \{\Re \setminus n\pi\}$ then ω^2 is an edge-based eigenvalue if and only if $\cos\omega$ is an eigenvalue of $\widetilde{A}$. For each corresponding eigenfunction, f, of $\widetilde{A}$, f can be extended to obtain an edge-based eigenfunction [6]. To summarize, for the edge-based eigenpair (f,λ), we have that:

1- $\cos\lambda$ is an eigenvalue of $\widetilde{A}$,
2- f is an eigenfunction of $\widetilde{A}$; that is $\widetilde{A}f = \cos\lambda\, f$,
3- $L_E f = \lambda f$ and $Lf = 0$.

The existence of a complete set of eigenvalues and eigenfunctions for the continuous Laplacian has been demonstrated in [7]. Friedman has extended the analysis to the edge-based Laplacian for finite graphs [6]. To outline the theory, let G be a finite graph. For G there exists eigenpairs f_i, λ_i for the edge-based Lalacian, such that

1- $0 \leqq \lambda_1 \leqq \lambda_2 \leqq ...,$
2- f_i satisfies the Dirichlet (Neumann) conditions,
3- f_i forms a complete orthonormal basis for $L^2_{Dir}(G)$ $(L^2(G))$,
4- $\lambda_i \to \infty$.

Physically, the equations $L_E f = \lambda f$ and $Lf = 0$ describe the vibrational modes associated with a taut strings on each edge that are joined together at the vertices. If we excite or "pluck" the system, it would produce tones with frequencies $\sqrt{\lambda}$, with λ ranging over the edge-based eigenvalues.

3 Pseudo Euclidean Space

A pseudo Euclidean space is an n-dimensional space $r_1, r_2, ..., r_n$ where $r_i = r$ or ir and r is a set of real numbers and $i = \sqrt{-1}$. In this section we describe how to manipulate vectors in a pseudo Euclidean space. Firstly, we explain how to compute the square distance between any arbitrary pair of vectors. Secondly, we show how to construct an orthonormal basis. Thirdly, we show how to project vectors from a pseudo Euclidean space onto a 2D sub-space.

3.1 Distance Function

With a pseudo Euclidean space R^n we assign a symmetric bilinear form $\rho : R^n \times R^n \to R$, $\rho(x,y) = x^T S y$ where S is the matrix whose elements $s_{ij} = \frac{1}{2}(d_{ii}^2 + d_{jj}^2 - d_{ij}^2)$; d is a distance function with pairwise distances d_{ij} for all $1 \leqslant i,j \leqslant n$. For any two vectors $x, y \in R^n$, $\rho(x,y)$ is the inner product of x and y and $\|x-y\|^2 = \rho(x-y, x-y)$ is the squared distance between x and y. Since S is real symmetric, there is an orthogonal matrix Ψ and a diagonal matrix Γ such that $S = \Psi \Gamma \Psi^T$, the elements δ_i of Γ are the eigenvalues of S arranged in order and the columns of Ψ are the correspondingly ordered eigenvectors. It is worth mentionng that if the matrix S has negative eigenvalues, then the squared distance between two vectors in the pseudo Euclidean space may be negative. It for this reason that we do not speak about "distance" between vectors in pseudo Euclidean space. Moreover, the fact that the squared distance between two vectors vanishes does not imply that these two vectors are the same. This is not the case in a Euclidean space.

3.2 An Orthonormal Basis

The columns $\{b_i\}, i = 1, ..., n$ of the matrix $B = I\Psi$ represent an orthogonal basis of R^n, since S is the matrix of ρ with respect to the natural basis $\{e_i\}, i = 1, ..., n$ where $e_i = (0, ..., 1_i, ..., 0)$. We can therefore write the bilinear form ρ with respect to the basis b_i as $S_b = \Psi^T S\Psi$, so that $S_b = \Gamma$. For any two vectors x and y in R^n, $\rho(x, y) = x^T Sy = [x_b^T \Psi^T][\Psi S_b \Psi^T][\Psi y_b \Psi^T]$. Hence, $\rho(x, y) = x_b^T S_b y_b = x_b^T \Gamma y_b$. Accordingly, the inner product of x and y can be written as $\rho(x, y) = \Sigma_{i=1}^n \delta_i (x_b)_i (y_b)_i$ and the squared distance as $\|x - y\|^2 = \Sigma_{i=1}^n \delta_i \left([x_b]_i - [y_b]_i\right)^2$. The matrix $X_b = X\Psi$ has as columns the coordinates with respect to the basis $\{b_i\}$. Conversely, the coordinate matrix $X_e = X_o \Psi^T$ has as columns the co-ordinate vectors with respect to an orthogonal natural basis.

Let us define a diagonal matrix $J = diag(j_{ij})$ with elements

$$j_i = \begin{cases} 1 & \delta_i > 0 \\ 0 & \delta_i = 0 \\ -1 & \delta_i < 0 \end{cases}$$

and $i = 1, \ldots, n$, Furthermore, let $\widetilde{\Gamma} = diag(\gamma_i)$ where

$$\gamma_i = \begin{cases} |\delta_i| & if\ \delta_i \neq 0 \\ 1 & otherwise \end{cases}$$

Now consider the matrix $\widetilde{\Psi} = \Psi \widetilde{\Gamma}^{-\frac{1}{2}}$. The first l columns of this matrix are orthonormal vectors with respect to $\{b_i\}$. To show this consider the matrix

$$\widetilde{\Psi}^T S \widetilde{\Psi} = \widetilde{\Gamma}^{-\frac{1}{2}} \Psi^T [\Psi S_b \Psi^T] \Psi \widetilde{\Gamma}^{-\frac{1}{2}} = \widetilde{\Gamma}^{-\frac{1}{2}} \Gamma \widetilde{\Gamma}^{-\frac{1}{2}} = J$$

The diagonal elements $J_i, i = 1, \ldots, l$ are either 1 or -1, while the remainder are zeros. Hence, the first l columns of the matrix $\widetilde{B} = B\widetilde{\Gamma}^{-\frac{1}{2}}$ form an orthonormal basis of R^l. Finally, for the matrix X_e whose columns are the co-ordinate vectors in the pseudo Euclidean space with respect to the natural basis $\{e_i\}_{i=1,\ldots,n}$, the corresponding matrix of coordinates with respect to the orthonormal basis $\{b_i\}_{i=1,\ldots,n}$ is $X_{\widetilde{b}} = X_e \widetilde{\Psi}$.

3.3 Projection into a 2D Subspace

Suppose we order eigenvalues of the matrix S so that first l^+ eigenvalues are positive, the following l^- are negative and the remainder are zero, where $l = l^+ + l^-$. As a result $\{b_i\}_{1 \leq i \leq l}$, and the first l columns of the matrix B given in

Section 3.2 form an orthogonal basis of the space R^l. Using the first l columns of the matrix $\widetilde{\Psi}$, we can locate the projections of the column vectors of the matrix X onto the space R^l with respect to $\{b_i\}_{1\leq i\leq l}$ as $X_l = B\widetilde{\Psi}^T$. To obtain the coordinates of X_l with respect to the orthonormal basis $\widetilde{b_i}$, we construct the matrix $X_{l_{\tilde{b}}} = \widetilde{B}\widetilde{D}_l^{-\frac{1}{2}}\Psi_l^T = (\overline{p}_1|\overline{p}_2|...)$, where $\widetilde{D}_l = diag(\gamma_i), 1 \leq i \leq l$ is the l^{th} leading principle submatrix of $\widetilde{D}$ and $\overline{p}_i$ is the projected coordinate vector of the i^{th} node of G. Again we can define the inner product of two arbitrary vectors, $x \; and \; y$, as $\rho(x,y) = \Sigma_{i=1}^n \delta_i (x_b)_i (y_b)_i$ and the squared distance as $\|x-y\|^2 = \Sigma_{i=1}^n \delta_i \left([x_b]_i - [y_b]_i\right)^2$.

To avoid problems associated with dealing with a space of high dimensionality, we will ignore the dimensions for which the eigenvalues are small in magnitude. Therefore, if we arrange the eigenvalues in descending order by their absolute values, the first k eigenvalues (typically $k = 2 \; or \; 3$) where $k < l$ will span a space R^k in which we can project the exact vector representation of the pseudo Euclidean space R^n.

4 Hausdorff Distance

We experiment with the wave kernel embedding as a graph characterization for the purposes of graph-matching. We represent the graphs under study using sets of coordinate vectors corresponding to the embedded node position, and compute the similarity of the sets resulting from different graphs using the robust modified Hausdorff distance. The Hausdorff distance provides a means of computing the distance between sets of unordered observations when the correspondences between the individual items are unknown. In its most general setting, the Hausdorff distance is defined between compact sets in a metric space. Given two such sets, we consider for each point in one set is the closest point in the second set. The modified Hausdorff distance is the average over all these values. More formally, the modified Hausdorff distance (MHD) [4] between two finite point sets $A = \left\{\overline{p}_1^A|\overline{p}_2^A|...|\overline{p}_{|V_A|}^A\right\}$ and $B = \left\{\overline{p}_1^B|\overline{p}_2^B|...|\overline{p}_{|V_B|}^B\right\}$ representing the projected embeddings of the graphs $G_A(V_A, E_A)$ and $G_B(V_B, E_B)$

$$H(A,B) = \max(h(A,B), h(B,A)) \tag{9}$$

where the directed modified Hausdorff distance from A to B is defined to be

$$h(A,B) = \frac{1}{N_A}\sum_{a\in A}\min_{b\in B}\|a-b\| \tag{10}$$

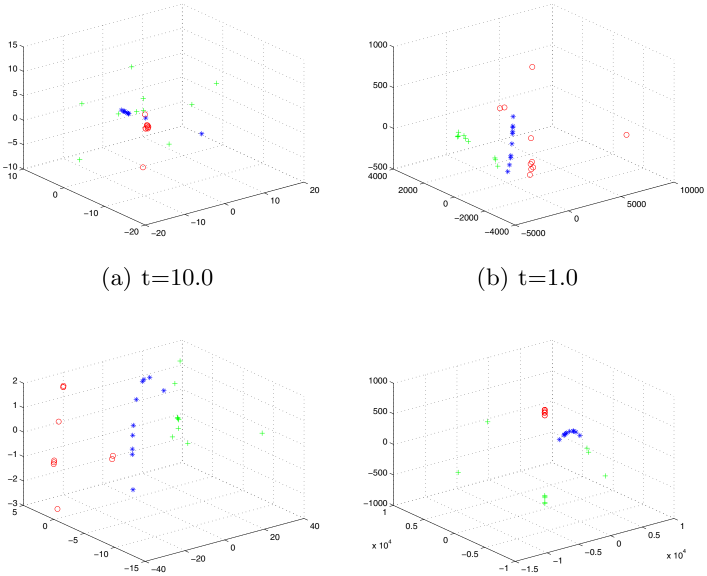
and $\|.\|$ is some underlying norm on the points of A and B (e.g., the L2 or Euclidean norm). We can now write the distances between two graphs as follows:

$$h_{MHD}(G_A, G_B) = \frac{1}{|V_A|}\sum_{i\in V_A}\min_{j\in V_B}\|\overline{p}_i^A - \overline{p}_j^B\|) \tag{11}$$

5 Experiments and Results

In our experiments we aim to investigate weather the edge-based wave kernel embedding can be used as a graph characterization, for gauging the similarity of graphs, and hence clustering them. We use the standard CMU,MOVI and chalet house sequences as our data set [9]. These data sets contain different views of model houses from equally spaced viewing directions. We have used ten views for each of the three houses. We have also applied our method to objects selected from the COIL database. This contains 72 different views of each object from equally spaced viewing directions. For each image corner features are extracted, and Delaunay graphs representing the arrangement of the feature points are constructed.

To commence, we compute the eigensystem of the edge-based Laplacian from the eigensystem of the normalized adjacency matrix, and hence compute the edge-based Laplacian matrix introduced in Section 2.3. The edge-based wave kernel then is computed as described in Section 2.1 with the values of $t = 10.0, 1.0, 0.1$ and 0.01. From the wave-kernel we compute the embedding co-ordinate matrix, whose columns are the coordinates of the embedded nodes in a pseudo-Euclidean space. Finally, we project the co-ordinate vectors onto a pseudo-Euclidean space with low dimension using the orthonormal basis as shown in Section 3. With the vector representations residing in a low dimension space we construct the distances matrices between the thirty different graphs using the modified Hausdorff distance [5]. Finally, we subject the distance matrices



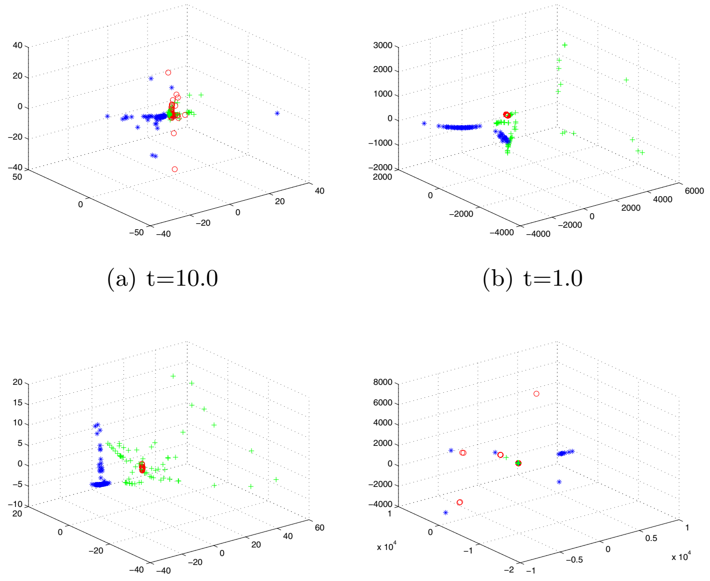
(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Fig. 1. MDS embedding obtained when using the Wave Kernel for the houses data



(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Fig. 2. MDS embedding obtained when using the Wave Kernel for the COIL data

Table 1. A rand index vs. t

	t=10	t=1.0	t=0.1	t=0.01
Houses data	0.2333	0.0000	0.0333	0.1000
COIL data	0.3333	0.3333	0.3333	0.7000

to multidimensional scaling MDS [3] to embed them into a 2D space. Here each graph is represented by a single point. Figure 1 shows the results obtained using the modified Hausdorff distance. The subfigures are ordered from left to right (up to down), using $t = 10.0, 1.0, 0.1$ and 0.01 in the wave kernel. We have also investigated the COIL data, and the results are shown in Figure 2.

To investigate the data in more detail Table 1 shows the Rand index for the data as a function of t. This index is computed as follows: 1) compute the mean for each cluster; 2) compute the distance from each point to each mean; 3) if the distance from correct mean is smaller than those to remaining means, then classification is correct, if not then classification is incorrect; 4) compute the Rand-index (incorrect/(incorrect+correct)).

There are number of conclusions to be drawn from the plots. First, the wave kernel gives good separation of the objects into distinct clusters particularly for values of t close to 1. Second, the individual objects form clear trajectories in the embedding space which correlate will with the view ordering.

6 Conclusion and Future Plan

In this paper we have established a procedure to embed the nodes of a graph into a pseudo-Riemannian manifold using the wave kernel, which is the solution of an edge-based wave equation. The edge-based Laplacian matrix was constructed using the eigensystem of the normalized adjacency matrix. Based on experiments on objects from two datasets (the York Model House and COIL datasets), we are confident that an edge-based wave kernel embedding can be used for the purpose of graph characterization.

To take this work further if the nodes of a graph residing on a pseudo-Riemannian manifold, then we can associate curvatures with the edges of the graph since these can be viewed as geodesics on the manifold [15]. In future research aimed at developing the work reported in this paper, we aim to investigate if we can use the geometry of the pseudo-Riemannian manifold to characterize graphs.

References

1. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Leen, T.K., Dietterich, T.G., Tresp, V. (eds.) Advances in neural information processing systems, vol. 14. MIT Press, Cambridge (2002)
2. Clarke, C.J.S.: On the global isometric embedding of pseudo-riemannian manifolds. In: Proceedings of Royal Society of London. A, vol. 314, pp. 417–428 (1970)
3. Cox, T., Cox, M.: Multidimensional Scaling. Chapman-Hall, Boca Raton (1994)
4. Dubuisson, M., Jain, A.: A modified hausdorff distance for object matching, pp. 566–568 (1994)
5. ElGhawalby, H., Hancock, E.R.: Measuring graph similarity using spectral geometry. In: Campilho, A., Kamel, M.S. (eds.) ICIAR 2008. LNCS, vol. 5112, pp. 517–526. Springer, Heidelberg (2008)
6. Friedman, J., Tillich, J.P.: Wave equations for graphs and the edge based laplacian. Pacific Journal of Mathematics 216(2), 229–266 (2004)
7. Gilbarg, D., Trudinger, N.S.: Elliptic Partial Differential Equations of Second Order. Springer, Heidelberg (1983)
8. Hurt, N.E.: Mathematical physics of quantum wires and devices. Kluwer Academic Publishers, Dordrecht (2000)
9. Luo, B., Wilson, R.C., Hancock, E.R.: Spectral embedding of graphs. Pattern Recogintion [36], 2213–2230 (2003)
10. Nash, J.F.: C1-isometric imbeddings. Ann. Math. 60, 383–396 (1954)
11. Nash, J.F.: The imbedding problem for riemannian manifolds. Ann. Math. 63, 20–63 (1956)
12. Pekalska, E., Haasdonk, B.: Kernel discriminant analysis for positive definite and indefinite kernels. IEEE transactions on pattern analysis and machine intelligence 31(6), 1017–1032 (2009)
13. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
14. Whitney, H.: Differentiable manifolds. Ann. of Math. 37(2), 645–680 (1936)
15. Xiao, B., Hancock, E.R., Wilson, R.C.: Graph characteristics from the heat kernel trace. Pattern Recognition 42(11), 2589–2606 (2009)
16. Xiao, B., Hancock, E.R., Yu, H.: Manifold embeddingforshapeanalysis. Neurocomputing 73, 1606–1613 (2010)

A Structured Learning Approach to Attributed Graph Embedding

Haifeng Zhao[1], Jun Zhou[2,3], and Antonio Robles-Kelly[2,3]

[1] School of Comp. Sci. & Tech., Nanjing Univ. of Sci. & Tech., Nanjing 210094, China
[2] RSISE, Bldg. 115, Australian National University, Canberra ACT 0200, Australia
[3] National ICT Australia (NICTA)*, Locked Bag 8001, Canberra ACT 2601, Australia

Abstract. In this paper, we describe the use of concepts from structural and statistical pattern recognition for recovering a mapping which can be viewed as an operator on the graph attribute-set. This mapping can be used to embed graphs into spaces where tasks such as categorisation and relational matching can be effected. We depart from concepts in graph theory to introduce mappings as operators over graph spaces. This treatment leads to the recovery of a mapping based upon the graph attributes which is related to the edge-space of the graphs under study. As a result, this mapping is a linear operator over the attribute set which is associated with the graph topology. Here, we employ an optimisation approach whose cost function is related to the target function used in discrete Markov Random Field approaches. Thus, the proposed method provides a link between concepts in graph theory, statistical inference and linear operators. We illustrate the utility of the recovered embedding for shape matching and categorisation on MPEG7 CE-Shape-1 dataset. We also compare our results to those yielded by alternatives.

1 Introduction

In the pattern analysis community, there has recently been renewed interests in the embedding methods motivated by graph theory. One of the best known of these is ISOMAP [1]. Related algorithms include locally linear embedding which is a variant of PCA that restricts the complexity of the input data using a nearest neighbor graph [2], and the Laplacian eigenmap that constructs an adjacency weight matrix for the data-points and projects the data onto the principal eigenvectors of the associated Laplacian matrix [3]. Collectively, these methods are sometimes referred to as manifold learning theory.

Embedding methods can also be used to transform the relational-matching problem into a point-pattern matching problem in a high-dimensional space. The idea is to find matches between pairs of point sets when there are noises, geometric distortion and structural corruption. This problem arises in shape analysis, motion analysis and stereo reconstruction. The main challenge in graph matching is how to deal with differences in node and edge structure. One of the most elegant approaches to the graph matching problem has been to use graph spectral methods [4], and exploit information conveyed

* National ICT Australia is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 70–79, 2010.

by the eigenvalues and eigenvectors of the adjacency matrix. More recently, Sebastian and Kimia [5] have used a distance metric analogous to the string edit distance to perform object recognition from a dataset of shock graphs.

The main argument levelled against the methods mentioned above is that they adopt a heuristic approach to the relational matching problem by using a goal-directed graph similarity measure. To overcome this problem, several authors have proposed more general approaches using ideas from information and probability theory. For instance, Wong and You [6] defined an entropic graph-distance for structural graph matching. Christmas, Kittler and Petrou [7] have shown how a relaxation labeling approach can be employed to perform matching using pairwise attributes whose distribution is modeled by a Gaussian. Wilson and Hancock [8] have used a MAP (maximum *a posteriori*) estimation framework to accomplish purely structural graph matching. Recently, Caetano *et al.* have proposed a method to estimate the compatibility functions for purposes of learning graph matching [9].

In this paper, we aim at estimating a linear mapping so as to embed a graph into a high-dimensional space where distances between nodes correspond to the structural differences between graphs. This can be viewed as a statistical learning process in which the goal of computation is the recovery of a linear operator which maps the attribute-set of a graph onto an embedding space in order to minimise a cost function arising from a Markovian formulation. In this manner, the recovered mapping is related to the space defined by the graph edge-set while being an operator on the attribute-set. Such an embedding permits the use of metrics in the target space for relational matching and shape categorisation tasks.

Thus, the motivation here is to recover a statistically optimal solution for the graph embedding problem. The bulk of the work elsewhere in the literature hinges in the use of dimensionality reduction techniques or relational similarity and matching algorithms. Here we take a more general view of the problem through learning. This learning approach leads to the statistical methods, where, for Graphical Models, MRFs are the ideal choice due to their use of pairwise potentials. Moreover, the method provides a link between structural and statistical pattern recognition techniques through the algebraic graph theory [10], graph spectral methods [4] and Markov Random Fields [11].

2 Graph Theory and Statistical Learning

Here we work with a data set Γ of attributed graphs. As mentioned earlier, we aim at learning a linear mapping $\mathcal{T}$ that can be used to embed the attributes of the graph-vertices into a space of dimensionality Ω whose basis is the optimal transformation of a linear map from the vertex to the edge space. In this manner, the embedding will reflect the structure of the edge-space of the graph while being based upon its attribute-set. This has two main advantages. Firstly, the target space for the learned mapping will reflect the structure of the graphs under study. Since similar graphs should have akin edge-spaces, this provides an embedding that is inherently related to a graph topology common to the set Γ. Secondly, note that the mapping $\mathcal{T}$ embeds the vertex-attributes into the graph edge-space according to a linear operator drawn from spectral geometry. This is not only practically useful but theoretically important since it provides a link between the spectra of graphs and linear operators.

2.1 Structured Learning

To commence, we require some formalism. Let $G_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{A}_i)$ denote the graph indexed i in Γ with node-set $\mathcal{V}_i = \{V_{i,1}, \ldots, V_{i,|\mathcal{V}_i|}\}$, edge-set $\mathcal{E}_i = \{e|V_{i,a}, V_{i,c} \in \mathcal{V}_i\}$ and attribute-set $\mathcal{A}_i = \{A_{i,1}, \ldots, A_{i,|\mathcal{V}_i|}\}$. Here, we aim at learning a global mapping $\mathcal{T}$ which is a matrix whose dimensionality is $\Omega\times \mid \mathcal{A}_i \mid$. In other words, we aim at recovering an operator which can embed any of the attributes for a given $G_i \in \Gamma$ into a space $\Re^{\Omega}$.

In this manner, the aim of computation is the recovery of the optimal transformation matrix over the field of attributes for the graphs in Γ. To recover this transformation matrix, we provide a link to Markov Random Field (MRF) models so as to abstract the problem into a graphical setting which takes profit of the inherent strength of Markovian approaches as related to pairwise potentials. To commence, we associate each $V_{i,a} \in \mathcal{V}_i$ with a hidden variable $\mathcal{X}_a$ in the state space Λ. The probability distribution represented by the MRF is given by

$$P(\mathcal{X}) = \frac{1}{Z} \prod_{G_i \in \Gamma} \prod_{V_{i,a} \in \mathcal{V}_i} \zeta_i(\mathcal{X}_a) \prod_{\substack{G_i \in \Gamma \\ G_k \in \Gamma}} \prod_{\substack{V_{i,a} \in \mathcal{V}_i \\ V_{k,b} \in \mathcal{V}_k}} \varphi_{i,k}(\mathcal{X}_a, \mathcal{X}_b) \tag{1}$$

where $\mathcal{X} = \{\mathcal{X}_a\}_{a=1,\ldots,|\mathcal{V}_i|}$ is the set of hidden variables, $\zeta_i(\mathcal{X}_a)$ and $\varphi_{i,k}(\mathcal{X}_a, \mathcal{X}_b)$ are unitary and binary potential functions which determine the likelihood of the graphs in the data set corresponding to the state $\varrho \in \Lambda$ and $Z = \int_{\Lambda} P(\mathcal{X})$ is the normalisation factor.

Since this normalisation factor is invariant with respect to $\mathcal{X}_a$, the inference of the above MRF model can be recast as an Maximum A Posteriori (MAP) estimation problem to maximise the probability $P(\mathcal{X})$ over the state space Λ. Moreover, we can consider $\mathcal{X}_a$ as a vector of continuous variables whose elements can be viewed as the linear product such that $\mathcal{X}_a = \mathcal{T}\mathbf{A}_i(a)$, where $\mathbf{A}_i(a)$ is the row indexed a of the matrix $\mathbf{A}_i$, whose rows correspond to the node attribute set $\mathcal{A}_i$ for the graph G_i. In other words, the hidden variables correspond to the embeddings of the graph-vertex attributes onto the space defined by the linear mapping $\mathcal{T} : \mathcal{A}_i \mapsto \Re^{\Omega}$. From an alternative viewpoint, we can consider $\mathcal{X}_a$ to be the weighted analogue of the attribute-vector for the a^{th} vertex in the graph indexed i in Γ.

Taking the logarithm of Equation 1, we have

$$\log P(\mathcal{X}) = \sum_{G_i \in \Gamma} \sum_{V_{i,a} \in \mathcal{V}_i} \mathcal{X}_a^T c_i(a) + \sum_{\substack{G_i \in \Gamma \\ G_k \in \Gamma}} \sum_{\substack{V_{i,a} \in \mathcal{V}_i \\ V_{k,b} \in \mathcal{V}_k}} \mathcal{X}_a^T w_{i,k}(a,b) \mathcal{X}_b \tag{2}$$

where $\log \zeta_i(\mathcal{X}_a) = \mathcal{X}_a^T c_i(a)$ and $\log \varphi_{i,k}(\mathcal{X}_a, \mathcal{X}_b) = \mathcal{X}_a^T w_{i,k}(a,b)\mathcal{X}_b$ are determined by the potential functions. Note that, in the expression above, $c_i(a)$ is a vector and $w_{i,k}(a,b)$ is a matrix, respectively. Also, the normalisation factor has become an additive constant and, as a result, we have removed it from further consideration.

Maximising the above cost function is equivalent to solving the original MRF inference problem, as defined in Equation 1. The cost function is in quadratic form and, hence, it is a natural choice to apply quadratic programming techniques to solve the relaxation problem. However, the Hessian of Equation 2 is determined by the coefficients of the second order term $w_{i,k}(a,b)$ which are not necessarily convex. A number of techniques have been proposed to relax the discrete problem above and convert the MRF cost functional into more tractable forms. Along these lines, some examples are SDP [12], SOCP [13], and spectral relaxation [14].

Instead of finding a continuous relaxation for the original cost function of the MRF model, we propose an alternative cost function which is closely related to it. Notice that the first and the second terms on the right-hand-side of the cost function in Equation 2 can be treated as correlation terms. The first of them measures the correlation between the graph and the single node potential. The second term measures the compatibility between graphs and the pairwise node-potential $w_{i,k}(a,b)$. By thinking of correlation as a measure of similarity and viewing it as an inverse distance, we can transform the maximisation problem at hand into a minimisation one. To do this, the $L2$ norm is a natural choice. The corresponding cost function is hence defined as follows

$$\min f(\mathcal{X}) = \sum_{G_i \in \Gamma} \sum_{V_{i,a} \in \mathcal{V}_i} ||c_i(a) - \mathcal{X}_a||^2 + \eta \sum_{\substack{G_i \in \Gamma \\ G_k \in \Gamma}} \sum_{\substack{V_{i,a} \in \mathcal{V}_i \\ V_{k,b} \in \mathcal{V}_k}} ||w_{i,k}(a,b)||^2 ||\mathcal{X}_a - \mathcal{X}_b||^2 \tag{3}$$

where η is a regularisation constant. For the sake of consistency, we have used vector norms where appropriate.

The reformulation of the cost function as above has several appealing properties. First, it is closely related to the MRF model in terms of its physical meaning. Like the MRF, our cost function also accommodates two complementary terms, i.e. a term which measures the compatibility between the data and its transformed field variable and a smoothness term which can be used to enforce the consistency between the variables for those nodes corresponding to the same graph, i.e. $i = k$. The main difference in the cost functions above is the replacement of the inner product with squared distance. Secondly, the cost function defined above is convex. Thus, we can always attain globally optimal solutions for the relaxed problem on the continuous variables. Moreover, the problem can be reduced to that of solving a sparse linear system of equations with positive semidefinite Hessian.

2.2 The L2-Norm

In this section, we explore the use of the L2-norm for purposes of recovering the mapping $\mathcal{T}$. We show how the extremisation of the cost function defined in Equation 3 can be reduced to that of solving a sparse linear system of equations. Recall that we have let $\mathcal{X}_a = \mathcal{T}\mathbf{A}_i(a)$, then the cost function can be rewritten as follows

$$\begin{aligned} \underset{\mathcal{T}}{\operatorname{argmin}} f(\mathcal{X}) = & \sum_{G_i \in \Gamma} \sum_{V_{i,a} \in \mathcal{V}_i} ||c_i(a) - \mathcal{T}\mathbf{A}_i(a)||^2 \\ & + \eta \sum_{\substack{G_i \in \Gamma \\ G_k \in \Gamma}} \sum_{\substack{V_{i,a} \in \mathcal{V}_i \\ V_{k,b} \in \mathcal{V}_k}} ||w_{i,k}(a,b)||^2 ||\mathcal{T}\mathbf{A}_i(a) - \mathcal{T}\mathbf{A}_k(b)||^2 \end{aligned} \tag{4}$$

Further, by using the factorisation properties of matrix norms and enforcing consistency between those nodes corresponding to the same graph, i.e. $||w_{i,k}(a,b)|| = 1$ iff $i = k$ and zero otherwise, we can greatly simplify the equation above as

$$\underset{\mathcal{T}}{\text{argmin}} f(\mathcal{X}) = \sum_{G_i \in \Gamma} \sum_{V_{i,a} \in \mathcal{V}_i} ||c_i(a) - \mathcal{T}\mathbf{A}_i(a)||^2 + \eta\gamma||\mathcal{T}||^2 \tag{5}$$

where

$$\gamma = \sum_{\substack{G_i \in \Gamma \\ G_k \in \Gamma}} \sum_{\substack{V_{i,a} \in \mathcal{V}_i \\ V_{k,b} \in \mathcal{V}_k}} ||w_{i,k}(a,b)||^2 ||\mathbf{A}_i(a) - \mathbf{A}_k(b)||^2 \tag{6}$$

Since γ does not depend on $\mathcal{T}$, and, hence, becomes a constant, from now on, and for the sake of convenience, we use the shorthand $\lambda = \eta\gamma$.

To minimise the cost function above, in practice, we can treat the problem as a continuous relaxation one which leads to a convex quadratic optimisation problem. To this end, we constraint the rows of the transformation matrix to add up to unity and introduce the vector of lagrange multipliers $\mathbf{N}$. The cost function becomes

$$g = f(\mathcal{X}) - \mathbf{N}^T(\mathcal{T}^T\mathbf{e} - \mathbf{e}) \tag{7}$$

where $\mathbf{e}$ is a vector of ones whose dimensionality is given by the context.

With these ingredients, we compute the partial derivative with respect to $\mathcal{T}$. We get

$$\frac{\partial g}{\partial \mathcal{T}} = 2(\mathcal{T}\mathbf{A}_i^T - \mathbf{C}_i^T)\mathbf{A}_i^T + 2\lambda\mathcal{T} \tag{8}$$

where $\mathbf{C}_i$ is a matrix whose a^{th} row corresponds to the vector $c_i(a)$ for the node indexed a in the graph G_i and $\mathbf{A}_i$ is the matrix defined earlier.

We now introduce the shorthands $\mathbf{F}_i = 2\mathbf{A}_i^T\mathbf{A}_i$ and $\mathbf{G}_i = 2\mathbf{C}_i^T\mathbf{A}_i^T$. As a result, we can now write the partial derivative above in the following manner

$$\frac{\partial g}{\partial \mathcal{T}} = \mathcal{T}\mathbf{F}_i - \mathbf{G}_i + 2\lambda\mathcal{T} \tag{9}$$

Following a similar approach, we can compute the partial of the function g with respect to the Lagrange multipliers in $\mathbf{N}$. By equating both partial derivatives to zero, we can write the solution as a the linear equation. This linear equation can be written using matrix notation in a straightforward manner by adding over the graphs in Γ.

3 Implementation Issues

Based on the above components, we now turn our attention to the implementation and application of our embedding method. Here, training can be viewed as the stage where the linear mapping $\mathcal{T}$ is learned from the graph vertex-attributes and the PCA of the incidence mapping. The testing stage then becomes the use of the mapped graph attributes into the target space for different purposes, for example, categorisation and relational matching.

The training stage starts from constructing the attributes of the graph-vertices. The attribute-set is application dependent. Here, we view, in general, the vertex-attributes

$\mathbf{A}_i(a)$ as vectors, where each of these has a one-to-one correspondence to a graph vertex. This also permits the computation of the weight matrix $\mathcal{W}$ with elements $\mathcal{W}(a,c)$ for the graph G_i. The weight matrix $\mathcal{W}$ can be related to the un-normalised Laplacian through the relationship $\mathcal{L} = \boldsymbol{D} - \mathcal{W}$, where $\boldsymbol{D}$ is a diagonal matrix such that $\boldsymbol{D} = diag(deg(1), deg(2), \ldots, deg(|\mathcal{V}_i|))$ and $deg(c) = \sum_{a=1}^{|\mathcal{V}_i|} \mathcal{W}(a,c)$ is the degree of the node indexed c in the graph [4].

The use of the graph Laplacian is important, since it permits the computation of the unary potentials $c_i(a)$. Consider the mapping $\mathcal{I}$ of the functions $\mathscr{g}(e)$ over the set of edges $\mathcal{E}_i$ to all the functions $\mathscr{h}(\cdot)$ over the set of vertices $\mathcal{V}_i$. The incidence mapping $\mathcal{I}$ is then an operator such that $\mathcal{I}\mathscr{g}(e) = \mathscr{h}(e_+) - \mathscr{h}(e_-)$, where the nodes $V_{i,a} = e_+$ and $V_{i,c} = e_-$ are the head and tail, respectively, of the edge $e \in \mathcal{E}_i$. As a result, $\mathcal{I}$ is a $\mid \mathcal{V}_i \mid \times \mid \mathcal{E}_i \mid$ matrix which satisfies

$$\mathcal{L} = \mathcal{I}\mathcal{I}^T \tag{10}$$

Note that the incidence mapping $\mathcal{I}$ is independent of the orientation of the edges in $\mathcal{E}_i$. Moreover, it is an operator, so it is independent of the vertex-basis, i.e. it is permutation invariant [10]. Further, the incidence mapping is recovered via a Young-Householder [15] decomposition on $\mathcal{L}$. With these ingredients, we proceed to define the unary potential $\zeta_i(\mathcal{X}_a)$ as an exponential family over the optimal linear transformation of the incidence mapping $\mathcal{I}$ into a space in $\Re^{\Omega}$. Thus, in practice, we can recover the potential $\zeta_i(\mathcal{X}_a)$ using the Principal Component Analysis (PCA) of the matrix $\mathcal{I}$. This is, we perform an eigendecomposition on $\mathcal{I}$ so as to select the leading Ω eigenvectors of the incidence mapping. This yields $c_i(a) = [\phi_1(a), \phi_2(a), \ldots, \phi_\Omega(a)]^T$, where $\phi_k(a)$ is the a^{th} coefficient of the k^{th} eigenvector of the incidence mapping $\mathcal{I}$ for the graph indexed i in Γ, such that $\phi_k = [\phi_k(1), \phi_k(2), \ldots, \phi_k(\mid \mathcal{V}_i \mid)]^T$.

Our choice of unary potential hinges in the developments in [16]. It can be shown that the space spanned by the PCA analysis is equivalent to the vertex-to-node scatter for the graph. Thus, we can view the terms $c_i(a)\mathcal{X}_a^T\mathbf{e}$ as the projections of the vectors $\mathcal{X}_a$ onto the subspace defined by the principal directions of the covariance for the mapping between the sets $\mathcal{V}_i$ and $\mathcal{E}_i$ in G_i. With $c_i(a)$ at hand, the linear mapping matrix $\mathcal{T}$ can be solved by extremising $f(\mathcal{X})$ as described in the previous sections.

As related to computational complexity, note that the embedding recovery is effected via Quadratic Programming and, therefore can be solved in polynomial time. The embedding operation, in practice, is a matrix multiplication, which can also be rendered computationally efficient. In summary, the step sequence of the method is as follows:

1. For every graph in Γ, compute the corresponding incidence mapping $\mathcal{I}$ via the Young-Householder decomposition of the Laplacian $\mathcal{L}$.
2. Compute the vectors $c_i(a)$ via PCA on the incidence mappings for the graphs in the data set.
3. Compute the linear mapping $\mathcal{T}$ by extremising the cost function in Equation 5.

Using the the linear mapping matrix $\mathcal{T}$, we can transform any $\mathbf{A}_i(a)$ into a target space, where each graph is represented as a matrix whose a^{th} row corresponds to the coordinates associated to the attribute indexed a in the i^{th} graph in Γ. As a result, relational matching between graphs can be performed by comparing the distances between the transformed attributes. This is due to the fact that there is a known one-to-one relationship between vertices and attributes in the graph. Further, these row vectors can be used to represent each graph as a probability distribution of pairwise vertex distance in the target space. In practice, these can be done via a histogram of distance frequencies whose bin-centres in the embedding space can be recovered using a clustering method such as k-means or maximum-likelihood estimation (MLE) approaches. This can be viewed as a codebook in the target space. In this way, we transfer the structural representation of a graph into a statistical representation that can be used for categorisation or relational matching tasks.

4 Experimental Results

Now, we turn our attention to the applicability of the embedding $\mathcal{T}$ to shape categorisation and relational matching settings. We use the MPEG7 CE-Shape-1 shape database, which contains 1400 binary shapes of 70 different classes with 20 images in each category. Fig. 1 shows some examples in the dataset. We have represented each shape as a graph whose vertices correspond to contour pixels sampled in regular intervals. Here, we have sampled 1 in every 10 pixels on the shape contours. With the sample contour pixels, we build a fully connected graph whose edge-weights are given by the Euclidean distances on the image plane between each pair of pixel-sites. Thus, the entries of the weight matrix for the graph correspond to the pairwise distances between the image-coordinates for every pair of vertices in the graph. The weigh matrix is then normalised to unity so as to have every weight in the graph in the interval $[0, 1]$. The attribute set is given by the frequency histogram of these distances for every clique. That is, for the a^{th} vertex in G_i, $\mathbf{A}_i(a)$ is given by the histogram for the edge-weights for the clique centered at the node indexed a. In our experiments, we have used 12 bins for the frequency histogram computation.

4.1 Relational Matching

Firstly, we illustrate the applicability of the embedding for relational matching making use of sample shapes in the dataset. We have learned the embedding $\mathcal{T}$ for the MPEG7 CE-Shape-1 database so as to embed the set of graphs corresponding to the shapes

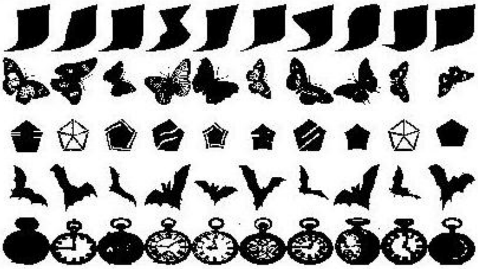


Fig. 1. Samples images from the MPEG7 CE-Shape-1 dataset

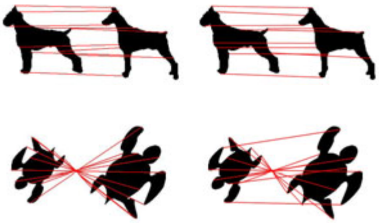


Fig. 2. Example matching results for our embedding (left-hand column) and graduated assignment (right-hand column)

into a space whose $\Omega = 70$. Once the embedding is at hand, relational matching was performed by viewing the node matching task as a point-matching one in the embedding space. We have used the coordinates $\mathcal{X}_a = \mathcal{T}\mathbf{A}_i(a)$ in the target space in order to compute the distances between nodes in the graph-vertices to be matched. The correspondences are deemed to be the nearest neighbours for the vertex embeddings for each of the graphs under study. That is, the vertex a in the data graph is a match to the node b in the model graph iff the Euclidean distance between the corresponding embedded coordinate vectors $\mathcal{X}_a$ and $\mathcal{X}_b$ is minimum for all the nodes in the graph pair.

In the the right-hand panels of Fig. 2, we show the results obtained using the recovered embedding $\mathcal{T}$. In the left-hand panel are the results obtained via graduated assignment [17]. Note that, from the panels, its clear that the distances in the target space provide a means to relational matching. Moreover, qualitatively, the results obtained making use of the embedding $\mathcal{T}$ show less mis-assignments than those recovered using the alternative.

4.2 Shape Categorisation

As mentioned earlier, one of the advantages of the embedding strategy adopted here is that it provides a means to connect structural pattern recognition with statistical pattern recognition. Here, we employ the histogram of pairwise distances in the embedding space for the coordinates $\mathcal{X}_a = \mathcal{T}\mathbf{A}_i(a)$ and construct a frequency histogram as a graph feature vector that can be used to obtain a "codeword" for each graph. To this end, we have used the frequency histograms for the distances between pairs of embeddings $\mathcal{X}_a = \mathcal{T}\mathbf{A}_i(a)$ for those attributes in the same graph. These distance histograms have been used to recover a codebook which is computed using k-means clustering, where $k = 200$. Using the pairwise distances for the histogram representation of graphs, we can construct a codebook for all shapes, which we have performed categorisation using a linear SVM. This enables us to perform supervised learning and, thus, the proposed method can take advantage of the recent progresses in machine learning.

For our shape categorisation experiments, we divided the graphs in the MPEG-7 dataset dataset into a training and a testing set. Each of these contains half of the graphs in each dataset. This is, we have used 700 graphs for training and 700 for testing. In contrast to our relational matching examples, here we have recovered the embedding

Table 1. Shape categorisation result comparison on the MPEG7-CE-Shape-1 dataset

Method	Proposed Method	Skeletal Contexts [18]	Shape Contexts [19]	Structured Learning [20]
Accuracy	91.8%	88.4%	76.51%	87.3%

matrix $\mathcal{T}$ making use of those graphs in the training set only. We have tuned the SVM parameters using ten-fold cross validation. The experiments were done on a server with Xeon 2.33GHz CPU and 16G memory. In our experiments, the main computational burden was at training time, where graph generation took approximately 10 minutes, whereas the k-means application and SVM training took 50s.

The categorisation results are shown in Table 1. For purposes of comparing our results with alternatives elsewhere in the literature, we show recognition rates for the skeletal matching method by Demirci *et al.* [18], the shape context method by Belongie *et al.* [19] and the structured learning method by Chen *et al.* [20]. The former two methods are unsupervised categorisation ones, while the last one is a supervised learning method. As shown in Table 1, our method shows a margin of improvement over the alternatives. Note that the alternative methods above have been specifically designed to provide optimum performance on binary shapes. Our method, on the other hand, makes a very simple abstraction of the shape in hand and can be naturally adapted to any shape whose structure can be captured by a relational structure. Moreover, our method is quite general in nature, permitting different tasks, such as the shape matching and categorisation, to be effected in a single computational framework.

5 Conclusions

In this paper, we have proposed a method to recover a mapping which is based upon the graph attribute-set and, at the same time, is inherently related to the graph topology. We have done this by drawing a link between the incidence mapping and a linear operator over the graph-vertex attributes. This linear operator is, in fact, a mapping that can be used for purposes of embedding graphs in a space where matching and categorisation tasks can be effected. We recover this embedding using a Markovian formulation which can be viewed as a learning process over a common topology for the set of graphs under study. This learning process is based upon a cost function which is convex in nature. We exemplify the utility of our method for shape categorisation and matching on MPEG7 CE-Shape-1 dataset.

Acknowledgement

This work was done while Haifeng Zhao was a visiting scholar at the Canberra Research Laboratory of NICTA. He is supported by the National Natural Science Foundation of China (NSFC) under No.60775015.

References

1. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319–2323 (2000)
2. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: NIPS. Number, vol. 14, pp. 634–640 (2002)
4. Chung, F.R.K.: Spectral Graph Theory. American Mathematical Society, Providence (1997)
5. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Shock-based indexing into large shape databases. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 731–746. Springer, Heidelberg (2002)
6. Wong, A.K.C., You, M.: Entropy and distance of random graphs with application to structural pattern recognition. IEEE TPAMI 7, 599–609 (1985)
7. Christmas, W.J., Kittler, J., Petrou, M.: Structural matching in computer vision using probabilistic relaxation. IEEE TPAMI 17(8), 749–764 (1995)
8. Wilson, R., Hancock, E.R.: Structural matching by discrete relaxation. IEEE TPAMI 19(6), 634–648 (1997)
9. Caetano, T., Cheng, L., Le, Q., Smola, A.: Learning graph matching. In: ICCV, pp. 14–21 (2007)
10. Biggs, N.L.: Algebraic Graph Theory. Cambridge University Press, Cambridge (1993)
11. Bremaud, P.: Markov Chains, Gibbs Fields, Monte Carlo Simulation and Queues. Springer, Heidelberg (2001)
12. Keuchel, J.: Multiclass image labeling with semidefinite programming. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 454–467. Springer, Heidelberg (2006)
13. Kumar, M., Torr, P., Zisserman, A.: Solving markov random fields using second order cone programming relaxations. In: CVPR, pp. 1045–1052 (2006)
14. Cour, T., Shi, J.: Solving markov random fields with spectral relaxation. In: Intl. Conf. on Artificial Intelligence and Statistics (2007)
15. Young, G., Householder, A.S.: Discussion of a set of points in terms of their mutual distances. Psychometrika 3, 19–22 (1938)
16. Ding, C., He, X.: K-means clustering via principal component analysis. In: ICML, pp. 225–232 (2004)
17. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. IEEE TPAMI 18(4), 377–388 (1996)
18. Demirci, M.F., Shokoufandeh, A., Dickinson, S.J.: Skeletal shape abstraction from examples. IEEE TPAMI 31(5), 944–952 (2009)
19. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE TPAMI 24(24), 509–522 (2002)
20. Chen, L., McAuley, J.J., Feris, R.S., Caetano, T.S., Turk, M.: Shape classification through structured learning of matching measures. In: CVPR (2009)

Combining Elimination Rules in Tree-Based Nearest Neighbor Search Algorithms

Eva Gómez-Ballester[1], Luisa Micó[1], Franck Thollard[2],
Jose Oncina[1], and Francisco Moreno-Seco[1]

[1] Dept. Lenguajes y Sistemas Informáticos
Universidad de Alicante, E-03071 Alicante, Spain
{eva,mico,oncina,paco}@dlsi.ua.es
[2] Grenoble University, LIG
BP 53, 38041 Grenoble Cedex 9
thollard@univ-st-etienne.fr

Abstract. A common activity in many pattern recognition tasks, image processing or clustering techniques involves searching a labeled data set looking for the nearest point to a given unlabelled sample. To reduce the computational overhead when the naive exhaustive search is applied, some fast nearest neighbor search (NNS) algorithms have appeared in the last years. Depending on the structure used to store the training set (usually a tree), different strategies to speed up the search have been defined. In this paper, a new algorithm based on the combination of different pruning rules is proposed. An experimental evaluation and comparison of its behavior with respect to other techniques has been performed, using both real and artificial data.

1 Introduction

Nearest Neighbor Search (NNS) is an important technique in a variety of applications including pattern recognition [6], vision [13], or data mining [1,5]. These techniques aim at finding the object of a set nearest to a given test object, using a distance function [6]. The use of a simple brute-force method is sometimes a bottleneck due to the large number of distances that should be computed and/or their computational effort. In this work we have considered the computational problem of finding nearest neighbors in general metric spaces. Spaces that may not be conveniently embedded or approximated in an Euclidean space are of particular interest. Many techniques have been proposed for using different types of structures (vp-tree [16], GNAT [3], sa-tree [10], AESA [14], M-tree [4]): the tree-based techniques are nevertheless more popular. The Fukunaga and Narendra algorithm (FNA [7]) is one of the first known tree-based example of this type of techniques. It prunes the traversal of the tree by taking advantage, as the aforementioned methods, of the triangular inequality of the distance between the prototypes. This sets up a general framework for designing and evaluating new pruning rules, as stated in [9].

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 80–89, 2010.

In this paper we study the combination of different pruning rules: recent table rule [12], a rule that is based on information stored in the sibling node (the sibling rule [9]), the original rule from the FNA (Fukunaga and Narendra rule, FNR), and a generalization of both the sibling rule and the FNR one [9]. We end up with a new algorithm for combining the rules that significantly reduces the number of distance computations.

The algorithm is evaluated on both artificial and real world data and compared with state-of-the-art methods.

The paper is organized as follows: we will first recall the FNA algorithm and define the general framework of the new algorithm (in particular how the tree is built). We then review the different rules we aim at combining (section 3). We then propose our new algorithm (section 4). Sections 5 presents the experimental comparison.

2 The Basic Algorithm

The FNA is a fast tree-based search method that can work in general metric spaces. In the original FNA the c-means algorithm was used to define the partition of the data. In the work by Gómez-Ballester et al [8] many strategies were explored: the best one, namely the *Most Distant from the Father tree* (MDF), in which the representative of the left node is the same as the representative of its father, is the strategy used in the experiments presented in this work. Thus, each time when an expansion of the node is necessary, only one new distance

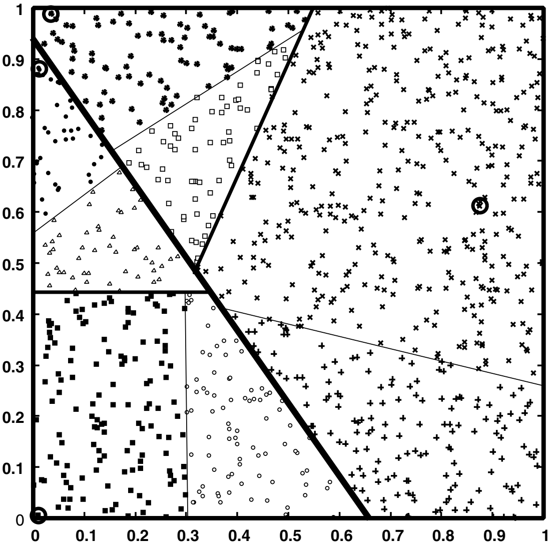


Fig. 1. Partition of the data using the MDF strategy. Representatives of each node in different levels are drawn as rings.

needs to be computed (instead of two), hence reducing the number of distances computed. This strategy was also successfully used by Noltomeier et al [11] in the context of bisector trees.

In the MDF tree each leaf stores a point of the search space. The information stored in each node t is S_t, the set of points stored in the leaves of t sub-tree, M_t (the representative of S_t) and the radius of S_t, $R_t = \text{argmax}_{x \in S_t} d(M_t, x)$. Figure 1 shows a partition of the data in a 2-dimensional unit hypercube. The root node will be associated with all the points of the set. The left node will represent all the points that belong to the hyperplane under the segment [(0, 0.95) ; (0.65,0)]; the right node will be associated with the other points. According to the MDF strategy, the representative of the right node (M_r) is the same as the father, and the representative of the left node (M_ℓ) is the most distant point to M_r. The space is then recursively partitioned.

3 A Review of Pruning Rules

Fukunaga and Narendra Rule (FNR)

The pruning rule defined by Fukunaga and Narendra for internal nodes makes use of the information in the node t to be pruned (with representative M_t and radius R_t) and the hyperspherical surface centered in the sample point x with radius $d(x, nn)$, where nn is current nearest prototype. To apply this rule it is necesary to compute the distance from the test sample to the representative of candidate node that aim to be eliminated. Figure 2a presents a graphical view of the Fukunaga and Narendra rule.

Rule: No $y \in S_t$ can be the nearest neighbor to x if $d(x, nn) + R_t < d(x, M_t)$

The Sibling Based Rule (SBR)

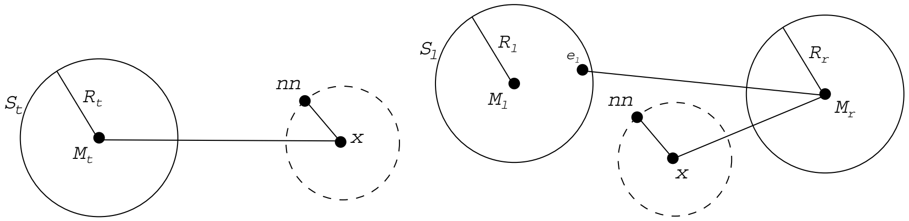
Given two sibling nodes r and ℓ, this rule requires that each node r stores the distance $d(M_r, e_\ell)$, that is the distance between the representative of the node, M_r, and the nearest point, e_ℓ, in the sibling node ℓ (S_ℓ). Figure 2b presents a graphical view of the Sibling based rule.

Rule: No $y \in S_\ell$ can be the nearest neighbor to x if $d(M_r, x) + d(x, nn) < d(M_r, e_\ell)$.

Unlike the FNR, SBR can be applied to eliminate node ℓ without computing $d(M_\ell, x)$, avoiding some extra distance computations at search time.

Generalized Rule (GR)

This rule is an iterated combination of the FNR and the SBR (due to space constraints we refer the reader to [9] for details on the generalized rule). In GR, the distance to the representative of a given node is needed to know if the node can be pruned or not.



(a) Geometrical view of FNR rule.

(b) Geometrical view of SBR rule.

The Table Rule (TR)

This recent rule [12] prunes the tree by taking the current nearest neighbor as a reference. In order to do so, a new distance should be defined:

Definition. Given a prototype or sample point p, the distance between p to a set of prototypes S is defined as

$$d(p,S) = \min_{y \in S} d(p,y)$$

At pre-process time, the distances from each prototype to each prototype set of each node t, S_t, in the tree are computed and stored in a table, allowing a constant time pruning. Note that the size of this table is quadratic in the number of prototypes since, as the tree is binary, the number of nodes is two times the number of prototypes.

Rule: No $y \in S_t$ can be the nearest neighbor to x if $2d(nn,x) < d(nn,S_t)$.

Figure 2 presents a graphical view of the table rule. Note that this rule can be used before computing the distance to the node that will be explored.

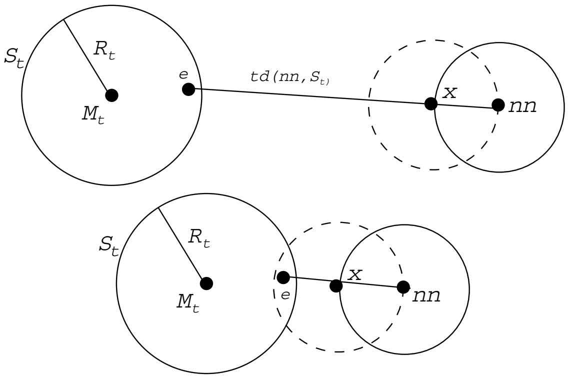


Fig. 2. Table rule and node S_t: situation where it can be pruned (up) and where it cannot (down)

4 CPR: Combining Pruning Rules Algorithm

In Algorithm 1 an efficient combination of pruning rules is proposed. Note that, as the GR generalizes both the FNR and the SBR, these two rules are not applied while the generalized one is activated (lines 11-19). When the MDF method is used to build the tree, it is important to note that each time a node is expanded, only one of the representatives is new (the left node), while the other (right) is the same as the father node (in this case, only the radius of the node can change). For this reason, in this case the distance $d_r = d(x, M_r)$ in line 9 is never computed (as it is already known). Then, when a node is examined during the search, every pruning that can be applied without computing a new distance is applied (lines 3 to 8). If none of these rules is able to prune, the distance to the current node is computed (line 9). The pruning rules that use the new distance are then applied (lines 11 to 28).

Algorithm 1. CPR(t,x)

```
   Data: $t$: a node tree; $x$: a sample point;
   Result: $nn$: the nearest neighbor prototype; $d_{\min}$: the distance to $nn$;
 1 if $t$ is not a leaf then
 2     $r = right\_child(t)$; $\ell = left\_child(t)$;
 3     if ( SBR($\ell$) || TR($\ell$) ) then
 4         if (no FNR(r)) && (no TR(r)) then
 5             CPR($r, x$) /* left (sibling) node has been pruned */;
 6         end
 7         Return /* ie prune both */ ;
 8     end
 9     $d_r = d(x, M_r)$ ;     $d_\ell = d(x, M_\ell)$;
10     update $d_{\min}$ and $nn$;
11     if Activated(GR) then
12         if $d_\ell < d_r$ then
13             if ( no GR($\ell$) ) then CPR($\ell, x$);
14             if ( no GR(r) ) then CPR($r, x$);
15         else
16             if (no GR(r)) then CPR($r, x$);
17             if (no GR($\ell$)) then CPR($\ell, x$);
18         end
19     else
20         if $d_\ell < d_r$ then
21             if (no FNR($\ell$)) && (no SBR($\ell$)) then CPR($\ell, x$);
22             if (no FNR(r)) && (no SBR(r)) then CPR($r, x$);
23         else
24             if (no FNR(r)) && (no SBR(r)) then CPR($r, x$);
25             if (no FNR($\ell$)) && (no SBR($\ell$)) then CPR($\ell, x$);
26         end
27     end
28 end
```

5 Experiments

We have performed some experiments in order to compare our algorithm with some state of the art methods. The first method, the multi-vantage-point tree (*mvp*), is a balanced tree requiring linear space where the arity can be extended and multiple pivots per node can be applied [2]. The second method is the Spatial Approximation Tree (*sat*), whose structure uses a graph based on Delaunay triangulation and it does not depend on any parameter [10]. The code of these algorithms comes from the SISAP library (`www.sisap.org`). We applied the *mvp* with only one pivot by node, a bucket size of 1 and an arity of 2 as this setting leads to better performances according to preliminary experiments on these data sets. All the experiments were performed on a Linux box with 16GB of memory.

From now and only for the graphs, the FNR rule (and respectively the SBR, GR and TR rules) will be abbreviated by "f" (respectively "s", "g" and "t"); consequently, combining the FBR and SBR will be referred as "fs". The combinations of rule "g" with "s" or "f" are not present as "g" generalizes these rules: every branch pruned by one of them is also pruned by "g".

In order to evaluate the performance of different combined rules, we present in this section the experiments on both artificial and real world data using different settings of our algorithm.

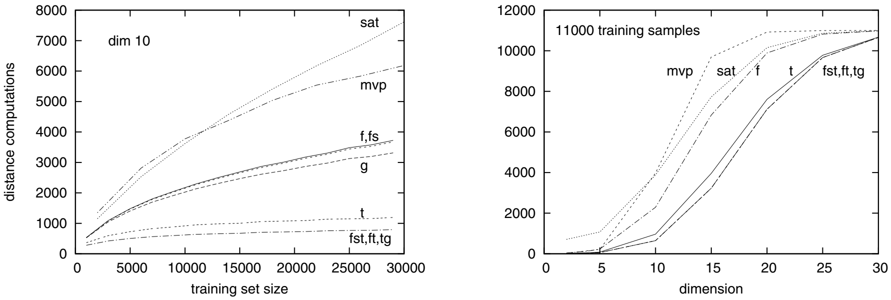
5.1 Artificial Data with Uniform Distributions

We consider here points drawn in a space of dimension n ranging from 5 to 30. The algorithms are compared with a growing number of prototypes. The size of the prototype sets ranged from $2,000$ prototypes to $30,000$ in steps of $4,000$. Each experiment measures the average distance computations of $10,000$ searches ($1,000$ searches over 10 different prototype sets). The samples are drawn from the same distribution.

Figure 3a shows the average number of distance computations in a 10-dimensional space following a uniform distribution. Standard deviation of measures is not included as it is almost negligible. As it can be seen, both *sat* and *mvp* are outperformed by the other pruning rules. Although the table rule also outperforms the FNR and GR ones, it is worth mentioning that these methods have a space consumption smaller than the table rule. In the case of small space capabilities, these methods should be preferred. Considering the classic FNA algorithm as a reference, we observe that GR and TR rules outperform the original rule, namely FNR. Moreover, it appears that combining the table rule, with either the sibling or generalized rule, does not perform better than combining the FNR and the table rule. This is important as the FNR rule has an effective computational cost smaller than the generalized rule. Furthermore, since the "g" rule also generalizes the sibling rule, the combination of "fst" does not perform better than "fg", as expected.

Another classic problem to address is *the curse of dimensionality*[1]. It expresses the fact that the volume of the unit hypercube increases exponentially with the

[1] The *curse of dimensionality* is usually considered in Euclidean spaces.



(a) Distance computations w.r.t. training set size in a 10-dimensional space.

(b) Distance computations w.r.t dimensionality.

Fig. 3. Comparison of different pruning rules combinations with *sat* and *mvp* algorithms

dimension of the space. In other words, the points tend to be at the same distance one to each other in great dimensions. In our setting, this will obviously prevent a large number of prunings: the algorithm will tend to behave like the brute force algorithm as the dimension increases. This algorithmic limitation is not a real problem since looking for a nearest neighbor does not make sense in a space where the distances between each pair of points are similar.

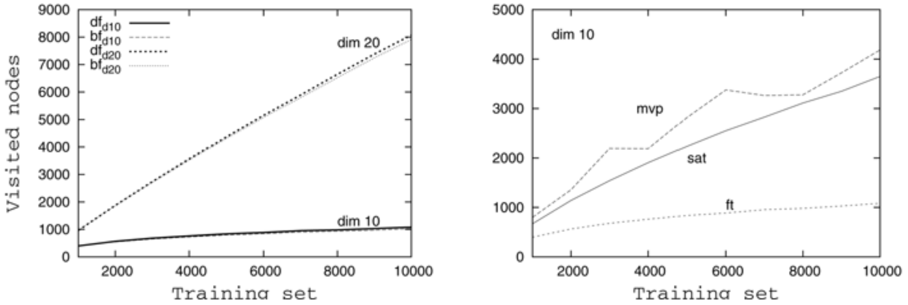
Figure 3b addresses a comparative analysis of the behavior of the methods as the dimension increases. The number of prototype is set to 11,000 points and the dimensionality ranges from 2 to 30. It can be observed here that the TR rule is less sensible to the dimensionality than the other methods. Moreover, as before, combining the TR rule with the FNR one still performs better than the other combinations: at dimension 25, the "ft" combination is able to save 20% of distance computations while the other methods compute all the distances, as the exhaustive search.

Two more experiments were performed: first, in order to show the differences when a best-first strategy is used instead of a depth-first strategy. In Figure 4a one can see that similar results are obtained, for this reason, only depth-first strategy is used in this work. Second, as well as the distance computations, the percentage of the database examined is analyzed for all the methods. Results can be seen in Figure 4b. As in the case of distance computations, the CPR method also reduces the overhead of the search visiting on average less nodes (or points in the data set).

5.2 Real World Data

To show the performance of the algorithms with real data, some tests were conducted on a spelling task. For these experiments, a database of 69,069 words of an English dictionary was used[2]. The input test of the speller was simulated

[2] Here again the databases are taken from the SISAP repository.



(a) Best-first (bf) versus depth-first (df) strategies in 10 and 20 dimensional spaces.

(b) Visited nodes w.r.t. training set size.

Fig. 4. Average number of visited nodes during the search for the best pruning rule combination, different search strategies and *sat* and *mvp* algorithms

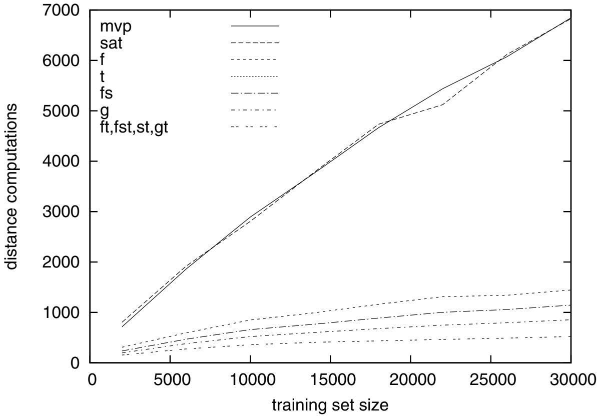


Fig. 5. Pruning rules combined in a spelling task in relation to others methods

distorting the words by means of random insertion, deletion and substitution operations over the words in the original dictionary. The Levenshtein distance [15] was used to compare the words. Dictionaries of increasing size (from 2,000 to 30,000) were obtained by extracting randomly words of the whole dictionary. Test points were obtained distorting the words in the training set. For each experiment, 1000 distorted words were generated and used as test set. To obtain reliable results, the experiments were repeated 10 times. The averages are showed on the plots.

The experiment performed in Figure 3a for artificial data (average number of distance computations using increasing size prototype sets) was repeated in the spelling task. Results are shown in Figure 5. The experiments show a reduction in the number of distance computations around 20% when the SBR rule is combined with the FNR, and around 40% for generalized rule with respect to the reference FNR rule. Moreover, when combining both the "f" and "t" rules (with or without the "g" rule), the resulting combination clearly outperforms the other combinations, as it happens with other kinds of data, saving 60% of the average number of distance computations.

6 Conclusions and Further Works

A new algorithm has been defined to optimize the combination of several pruning rules using the FNA tree-based search algorithm. When the rules are applied alone, reductions between 20% and 60% are obtained for low dimensions and this reduction decreases with the dimensionality (a normal behavior since the problem is getting harder with increasing dimensionalities) when comparing with the baseline FNR rule. When the rules are combined, more reductions in the average number of distance computations and in the overhead of the methods (measured as the average number of visited nodes or points), in particular can be observed (*e.g.* roughly 80% reduction in a 10-dimensional space). Similar results are also obtained on a real world task (namely a spelling task).

We are currently studying new pruning rules and combinations, and also how to use them in dynamic tree structures. We think also that this algorithm can be adapted with minor changes to other tree-based search methods not explored in this work.

Acknowledgments

The authors thank the Spanish CICyT for partial support of this work through projects TIN2009-14205-C04-C1 and TIN2009-14247-C02-02, the IST Programme of the European Community, under the PASCAL Network of Excellence, (IST–2006-216886), and the program CONSOLIDER INGENIO 2010 (CSD2007-00018).

References

1. Böhm, C., Krebs, F.: High performance data mining using the nearest neighbor join. In: ICDM 2002: Proceedings of the 2002 IEEE International Conference on Data Mining. IEEE Computer Society, Los Alamitos (2002)
2. Bozkaya, T., Ozsoyoglu, M.: Distance-based indexing for high-dimensional metric spaces. In: SIGMOD 1997: Proceedings of the 1997 ACM SIGMOD international conference on Management of data, pp. 357–368. ACM, New York (1997)
3. Brin, S.: Near neighbor search in large metric spaces. In: VLDB Conference, pp. 574–584 (1995)
4. Ciaccia, P., Patella, M., Zezula, P.: M-tree: An efficient access method for similarity search in metric spaces. In: VLDB Conference, pp. 426–435. Morgan Kaufmann Publishers, Inc., San Francisco (1997)
5. Dasarathy, B.V.: Data mining tasks and methods: Classification: nearest-neighbor approaches, pp. 288–298 (2002)
6. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, New York (2000)

7. Fukunaga, K., Narendra, P.M.: A branch and bound algorithm for computing k-nearest neighbors. IEEE Transactions on Computers, IEC 24, 750–753 (1975)
8. Gómez-Ballester, E., Micó, L., Oncina, J.: Some improvements in tree based nearest neighbour search algorithms. In: Sanfeliu, A., Ruiz-Shulcloper, J. (eds.) CIARP 2003. LNCS (LNAI), vol. 2905, pp. 456–463. Springer, Heidelberg (2003)
9. Gómez-Ballester, E., Micó, L., Oncina, J.: Some approaches to improve tree-based nearest neighbour search algorithms. Pattern Recognition 39(2), 171–179 (2006)
10. Navarro, G.: Searching in metric spaces by spatial approximation. In: SPIRE 1999: Proceedings of the String Processing and Information Retrieval Symposium, p. 141. IEEE Computer Society, Los Alamitos (1999)
11. Noltemeier, H., Verbarg, K., Zirkelbach, C.: Monotonous bisector* trees - a tool for efficient partitioning of complex scenes of geometric objects. In: Data Structures and Efficient Algorithms, Final Report on the DFG Special Joint Initiative, London, UK, pp. 186–203. Springer, Heidelberg (1992)
12. Oncina, J., Thollard, F., Gómez-Ballester, E., Micó, L., Moreno-Seco, F.: A tabular pruning rule in tree-based pruning rule fast nearest neighbour search algorithms. In: Martí, J., Benedí, J.M., Mendonça, A.M., Serrat, J. (eds.) IbPRIA 2007. LNCS, vol. 4478, pp. 306–313. Springer, Heidelberg (2007)
13. Shakhnarovich, G., Darrell, T., Indyk, P.: Nearest-Neighbor Methods in Learning and Vision. MIT Press, Cambridge (2006)
14. Vidal, E.: New formulation and improvements of the nearest-neighbour approximating and eliminating search algorithm (AESA). Pattern Recognition Letters 15, 1–7 (1994)
15. Wagner, R.A., Fischer, M.J.: The string-to-string correction problem. Journal of the Association for Computing Machinery 21(1), 168–173 (1974)
16. Yianilos, P.N.: Data structures and algorithms for nearest neighbor search in general metric spaces. In: Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, pp. 311–321 (1993)

Localized Projection Learning

Kazuki Tsuji, Mineichi Kudo, and Akira Tanaka

Division of Computer Science,
Graduate School of Information Science and Technology,
Hokkaido University, Sapporo, 060-0814, Japan
{kazuki,mine,takira}@main.ist.hokudai.ac.jp

Abstract. It is interesting to compare different criteria of kernel machines. In this paper, the following is made: 1) to cope with the scaling problem of projection learning, we propose a dynamic localized projection learning using k nearest neighbors, 2) the localized method is compared with SVM from some viewpoints, and 3) approximate nearest neighbors are demonstrated their usefulness in such a localization. As a result, it is shown that SVM is superior to projection learning in many classification problems in its optimal setting but the setting is not easy.

1 Introduction

In pattern recognition, the design of a classifier can be made through regression. To achieve this, a dummy output y is introduced instead of the class-label output, e.g., $y = +1$ for one class and $y = -1$ for another class in two-class problems as seen in the paradigm of the support vector machines (SVMs) [1]. The goal of this type of classifiers is to estimate the target function f such as $y = f(\boldsymbol{x})$, where $\boldsymbol{x} \in \mathbf{R}^p$ is a sample with p input features and $y \in \mathbf{R}$ is the corresponding output. We try to find a good approximator/regressor $\hat{f}$ from a limited number of training sample pairs $(\boldsymbol{x}_i, y_i)$ $(i = 1, \ldots, \ell)$. According to whether there is noise or not, a regressor or an approximator becomes more appropriate than the other.

In this paper we compare two typical but criterion-different kernel machines: SVM [1] and Projection Learning [2]. The goal of this paper is described in three-fold: 1) to propose a localized projection learning to bring scalability into the projection learning, 2) to examine how well the proposed localized projection learning is competitive to the original projection learning, and 3) using the localized projection learning, to compare the projection learning and the SVM in performance and in speed.

2 Reproducing Kernel Hilbert Spaces

Reproducing kernel Hilbert spaces [3,4] is the key concept to interpret above approaches in a unified framework.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 90–99, 2010.

Definition 1. [3] *Let $\mathbf{R}^p$ be a p-dimensional real vector space and let $\mathcal{H}$ be a class of functions defined on $\mathcal{D} \subset \mathbf{R}^p$, forming a Hilbert space of real-valued functions. A function $K(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ $(\boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathcal{D})$ is called the reproducing kernel if*

1. *For every $\tilde{\boldsymbol{x}} \in \mathcal{D}$, $K(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is a function of $\boldsymbol{x}$ belonging to $\mathcal{H}$.*
2. *For every $\tilde{\boldsymbol{x}} \in \mathcal{D}$ and every $f \in \mathcal{H}$, $f(\tilde{\boldsymbol{x}}) = \langle f(\boldsymbol{x}), K(\boldsymbol{x}, \tilde{\boldsymbol{x}})\rangle_{\mathcal{H}}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the inner product of the Hilbert space $\mathcal{H}$.*

The Hilbert space $\mathcal{H}$ that has a reproducing kernel is called a reproducing kernel Hilbert space (RKHS). The *reproducing property* (the second condition) enables us to treat a function value by the inner product of two elements of $\mathcal{H}$.

Next, we introduce the Schatten product [5] that is a convenient tool to reveal the reproducing property of kernels.

Definition 2. [5] *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be Hilbert spaces. The Schatten product of $g \in \mathcal{H}_2$ and $h \in \mathcal{H}_1$ is defined by*

$$(g \otimes h)f = \langle f, h \rangle_{\mathcal{H}_1} g, \quad f \in \mathcal{H}_1.$$

Note that $(g \otimes h)$ is a linear operator from $\mathcal{H}_1$ onto $\mathcal{H}_2$. It is easy to show that the following relations hold for $h, v \in \mathcal{H}_1$, $g, u \in \mathcal{H}_2$.

$$(h \otimes g)^* = (g \otimes h), \quad (h \otimes g)(u \otimes v) = \langle u, g \rangle_{\mathcal{H}_2} (h \otimes v),$$

where the superscript * denotes the adjoint operator.

3 Projection Learning (PL)

Let $\{(y_i, \boldsymbol{x}_i) | i = 1, \ldots, \ell\}$ be a given training data set $\boldsymbol{x}_i \in \mathbf{R}^p, y_i \in \mathbf{R}$, satisfying

$$y_i = f(\boldsymbol{x}_i) + n_i,$$

where f denotes the true function and n_i denotes a zero-mean additive noise.

In this paper, we assume that the unknown function f belongs to the RKHS $\mathcal{H}_K$ with the kernel function K. If $f \in \mathcal{H}_K$, then by the reproducing property above equation is rewritten as

$$y_i = \langle f(\boldsymbol{x}), K(\boldsymbol{x}, \boldsymbol{x}_i) \rangle_{\mathcal{H}_K} + n_i.$$

Let $\boldsymbol{y} = [y_1, \ldots, y_\ell]'$ and $\boldsymbol{n} = [n_1, \ldots, n_\ell]'$ with the superscript $'$ denoting the transposed matrix (or vector), then applying the Schatten product yields

$$\boldsymbol{y} = \left(\sum_{k=1}^{\ell} [\boldsymbol{e}_k^{(\ell)} \otimes K(\boldsymbol{x}, \boldsymbol{x}_k)] \right) f(\boldsymbol{x}) + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{e}_k^{(\ell)}$ denotes the kth vector of the canonical basis of $\mathbf{R}^\ell$. For a convenience of description, with the sample set $\boldsymbol{X} = \{x_1, \ldots, x_\ell\}$, we write

$$A_{K,X} = \left(\sum_{k=1}^{\ell} [\boldsymbol{e}_k^{(\ell)} \otimes K(\boldsymbol{x}, \boldsymbol{x}_k)] \right).$$

Here $A_{K,X}$ is a linear operator that maps an element of $\mathcal{H}_K$ onto $\mathbf{R}^\ell$. Then Eq.(1) can be simply written by

$$\boldsymbol{y} = A_{K,X} f + \boldsymbol{n}. \tag{2}$$

That is, the result of sampling of f on $\boldsymbol{X}$ of fixed ℓ samples was contaminated by noise $\boldsymbol{n}$ and observed as $\boldsymbol{y}$.

Projection learning [2] is derived to attain the minimum squared error on $\boldsymbol{X}$ between the target function f and an estimator $\hat{f}$ measured in $\mathcal{H}$ in the case of zero-mean noise. By solving (2) without $\boldsymbol{n}$, such an optimal estimator is given as

$$\begin{aligned} \hat{f}(\cdot) &= A^{+}_{K,X}\boldsymbol{y} = A^{*}_{K,X}(A_{K,X}A^{*}_{K,X})^{+}\boldsymbol{y} && (3)\\ &= \sum_{k=1}^{\ell} \boldsymbol{y}' G^{+}_{K,X}\boldsymbol{e}^{(\ell)}_k K(\cdot, \boldsymbol{x}_k), && (4) \end{aligned}$$

where $*$ is the adjoint matrix, $+$ is the Moore-Penrose generalized inverse, and $G_{K,X} = (K(\boldsymbol{x}_i, \boldsymbol{x}_j))$ is the $\ell \times \ell$ "Gram matrix."

When no error exists, Eq.(3) can be rewritten as

$$\hat{f} = A^{*}_{K,X}(A_{K,X}A^{*}_{K,X})^{+}A_{K,X}f = P_{R(A^{*}_{K,X})}f = \sum_{k=1}^{\ell}\alpha_k K(\boldsymbol{x}, \boldsymbol{x}_k).$$

As a result, we have $\hat{f}$ as the minimizer of

$$J_{PL}(\hat{f}) = ||f - \hat{f}||^2_{\mathcal{H}_K}, \hat{f} \in \text{span}\{K(\cdot, x_1), K(\cdot, x_2), \ldots, K(\cdot, x_\ell)\},$$

where span$\{\cdot\}$ is the closure of linear combinations of the elements.

It is also easy to show that $\hat{f}$ is equivalent to f on a sample set $\boldsymbol{X}$, that is, with $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_\ell)]'$ and $\hat{\boldsymbol{f}} = [\hat{f}(\boldsymbol{x}_1), \ldots, \hat{f}(\boldsymbol{x}_\ell)]'$, $\hat{\boldsymbol{f}} \equiv \boldsymbol{f}$ holds as long as $G^{+} = G^{-1}$.

The solution by PL is optimal in the sense of minimum squared error if no error is considered. However, things change when noise is taken into account. A clear relationship for the case is given by Tanaka *et al.* [6,7]. Roughly speaking, as the number of training samples increases, the projection error is reduced but the error caused by noise is increased, so that the total approximation error $||f - \hat{f}||^2_{\mathcal{H}_k}$ is unknown on whether reduced or increased. When noise exists, from (2) and (3), our estimate becomes

$$\hat{f} = A^{+}_{K,X}\boldsymbol{y} = A^{+}_{K,X}(A_{K,X}f + \boldsymbol{n}) = A^{+}_{K,X}A_{K,X}f + A^{+}_{K,X}\boldsymbol{n} = P_{R(A^{*}_{K,X})}f + A^{+}_{K,X}\boldsymbol{n}.$$

In [6], $||P_{R(A^{*}_{K,X})}f||^2_{\mathcal{H}_K} = \boldsymbol{f}'G^{+}_{K,X}\boldsymbol{f}$ and, in [7], $||A^{+}_{K,X}\boldsymbol{n}||^2_{\mathcal{H}_K} = \boldsymbol{n}'G^{+}_{K,X}\boldsymbol{n}$ were shown. Thus, we have the following relationship:

$$\begin{aligned} ||f - \hat{f}||^2_{\mathcal{H}_K} &= ||f - P_{R(A^{*}_{K,X})}f||^2_{\mathcal{H}_K} + ||A^{+}_{K,X}\boldsymbol{n}||^2_{\mathcal{H}_K} \\ &= ||f||^2_{\mathcal{H}_K} - \boldsymbol{f}'G^{+}_{K,X}\boldsymbol{f} + \boldsymbol{n}'G^{+}_{K,X}\boldsymbol{n}. && (5) \end{aligned}$$

Since $G^{+}_{K,X}$ is symmetric and non-negative definite, the second and third terms increase in their absolute values as the sizes of $\boldsymbol{f}, \boldsymbol{n}, G_{K,X}$ increase with increasing samples. Therefore a trade-off arises.

4 Support Vector Machines (SVM)

Although it was originally formulated for classification as the norm minimization under a separation condition, support vector machines are also seen as a regression algorithm. Indeed the criterion in a regression form is given by

$$J_{SVM}(\hat{f}) = \frac{1}{\ell}\sum_{i=1}^{\ell} |1 - y_i \hat{f}(x_i)|_{+} + \lambda \|\hat{h}\|^2_{\mathcal{H}_K}, \tag{6}$$

where $|\cdot|_{+}$ is a function to remain the value for a positive value and zero otherwise. Here, f is decomposed as $\hat{f} = \hat{h} + \hat{c}$ and $\hat{h} \in \mathcal{H}$ and $\hat{c}$ is a constant and $P\hat{f} = \hat{h}$. Therefore, if $\{1\} \subset \mathcal{H}_K$ then this equation is rewritten as

$$J_{SVM}(\hat{f}) = \frac{1}{\ell}\sum_{i=1}^{\ell} |1 - y_i \hat{f}(x_i)|_{+} + \lambda \|P\hat{f}\|^2_{\mathcal{H}_K}, \tag{7}$$

where $P\hat{f}$ is the projection of $\hat{f}$ into the orthogonal complement of $\{1\}$ in $\mathcal{H}_K$.

It is also known that the minimizer of this criterion has the form $\hat{f}(\cdot) = \sum_{i=1}^{c} c_i K(\cdot, x_i) + c_0$. That is, $P\hat{f}$ is the projection of $\hat{f}$ into span$\{K(\cdot, x_1), K(\cdot, x_2), \ldots, K(\cdot, x_\ell)\}$.

5 Localized Projection Learning (LPL)

Projection learning (4) can be expressed explicitly as

$$\begin{aligned}\hat{f}(\boldsymbol{x}) &= (K(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, K(\boldsymbol{x}, \boldsymbol{x}_\ell)) \begin{pmatrix} K(\boldsymbol{x}_1, \boldsymbol{x}_1) & K(\boldsymbol{x}_1, \boldsymbol{x}_2) & \cdots & K(\boldsymbol{x}_1, \boldsymbol{x}_\ell) \\ K(\boldsymbol{x}_2, \boldsymbol{x}_1) & K(\boldsymbol{x}_2, \boldsymbol{x}_2) & \cdots & K(\boldsymbol{x}_2, \boldsymbol{x}_\ell) \\ \vdots & \vdots & \ddots & \vdots \\ K(\boldsymbol{x}_\ell, \boldsymbol{x}_1) & K(\boldsymbol{x}_\ell, \boldsymbol{x}_2) & \cdots & K(\boldsymbol{x}_\ell, \boldsymbol{x}_\ell) \end{pmatrix}^{-1} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_\ell \end{pmatrix} \\ &= \boldsymbol{K}(\boldsymbol{x}) G^{-1}_{K,X} \boldsymbol{y}.\end{aligned}$$

In the localized version, we use data-dependent $k(\leq \ell)$ training samples for $\hat{f}$. Let N_i $(i = 1, 2, \ldots, k)$ be the ith nearest neighbor of $\boldsymbol{x}$ in $\boldsymbol{X}$. By limiting the sample set from $\boldsymbol{X}$ to $\boldsymbol{X}_{kNN}(\boldsymbol{x}) = \{\boldsymbol{x}_{N_1}, \boldsymbol{x}_{N_2}, \ldots, \boldsymbol{x}_{N_k}\}$, we have

$$\begin{aligned}\hat{f}(\boldsymbol{x}) &= (K(\boldsymbol{x}, \boldsymbol{x}_{N_1}), \ldots, K(\boldsymbol{x}, \boldsymbol{x}_{N_k})) \begin{pmatrix} K(\boldsymbol{x}_{N_1}, \boldsymbol{x}_{N_1}) & \cdots & K(\boldsymbol{x}_{N_1}, \boldsymbol{x}_{N_k}) \\ \vdots & \ddots & \vdots \\ K(\boldsymbol{x}_{N_k}, \boldsymbol{x}_{N_1}) & \cdots & K(\boldsymbol{x}_{N_k}, \boldsymbol{x}_{N_k}) \end{pmatrix}^{-1} \begin{pmatrix} y_{N_1} \\ y_{N_2} \\ \vdots \\ y_{N_k} \end{pmatrix} \\ &= \boldsymbol{K}_{kNN}(\boldsymbol{x}) G^{-1}_{K,\boldsymbol{X}_{kNN}} \boldsymbol{y}_{kNN}\end{aligned}$$

This localization is clearly effective only for kernels in which the value of $K(\boldsymbol{x}, \boldsymbol{y})$ takes near zero when $\boldsymbol{x}$ and $\boldsymbol{y}$ are far from each other in the original feature space, such as a radial basis function.

It should be noted that $\hat{f}$ changes depending on a given data $\boldsymbol{x}$. A singularity of $G_{K,X}$ often becomes a problem when ℓ is large, but the reduced $G_{K,X_{kNN}}$ becomes non-singular in most cases. Of course, the most advantage of LPL is the calculation cost compared with the original PL. Note that we have to calculate the inverse of $G_{K,X_{kNN}}$ of size $k \times k$ for every data $\boldsymbol{x}$, although the inverse of $G_{K,X}$ is possible to be pre-calculated in the original PL. However, the cost is negligible because k is ignorably small to ℓ.

Even in the performance, we might expect a little raise compared with PL. Let us consider a trade-off of Eq. (5). By choosing the k nearest neighbors of a given data, the first (approximation) term is almost kept compared with the case that all samples are used, while the second time is necessary reduced because of its smaller size. As a result, a better trade off can be expected to be realized.

6 Comparison between SVM and LPL

Now let us compare SVM, PL and LPL in their criteria to minimize. For simplicity, we assume that every training samples are correctly classified with a positive margin. We also assume $\{1\} \subset \mathcal{H}_K$.

Then, SVM minimizes

$$J_{SVM}(\hat{f}) = \|\hat{f}\|^2_{\mathcal{H}_K}, \hat{f} \in \mathrm{span}\{K(\cdot, x_1), K(\cdot, x_2), \ldots, K(\cdot, x_\ell)\}$$

under the condition of $\sum_{i=1}^{N} |1 - y_i \hat{f}(x_i)|_+ = 0$. Eventually, only support vectors $S_1, \ldots, S_t$ are necessary for minimizing

$$J_{SVM}(\hat{f}) = \|\hat{f}\|^2_{\mathcal{H}_K}, \hat{f} \in \mathrm{span}\{K(\cdot, x_{S_1}), K(\cdot, x_{S_2}), \ldots, K(\cdot, x_{S_t})\}.$$

Similarly, PL minimizes

$$J_{PL}(\hat{f}) = ||f - \hat{f}||^2_{\mathcal{H}_K}, \hat{f} \in \mathrm{span}\{K(\cdot, x_1), K(\cdot, x_2), \ldots, K(\cdot, x_\ell)\}.$$

under the condition of $\sum_{i=1}^{N} |f(x_i) - \hat{f}(x_i)| = 0$, and LPL minimizes, with k nearest neighbors,

$$J_{LPL}(\hat{f}) = ||f - \hat{f}||^2_{\mathcal{H}_K}, \hat{f} \in \mathrm{span}\{K(\cdot, x_{N_1}), K(\cdot, x_{N_2}), \ldots, K(\cdot, x_{N_k})\}.$$

under the condition of $\sum_{i=1}^{k} |f(x_{N_i}) - \hat{f}(x_{N_i})| = 0$. These criteria are shown in Fig.1 and Fig.2. The following is worth noticing:

1. SVM and PL find the estimator $\hat{f}$ in the same subspace of $\mathcal{H}_K$ when a constant is ignored.
2. Nevertheless, SVM finds the minimum norm solution in $||\hat{f}||$ and PL finds the minimum difference solution in $||f - \hat{f}||$. They take reverse directions.

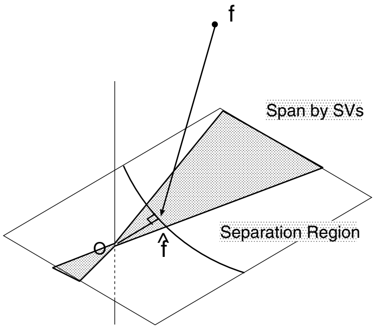


Fig. 1. Support Vector Machine

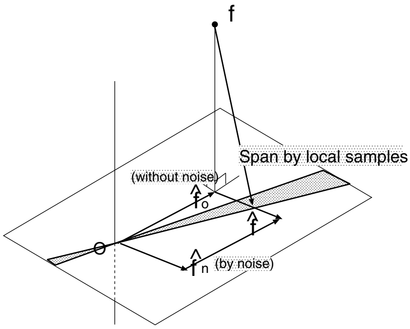


Fig. 2. Localized Projection Learning

3. LPL is expected to simulate PL well as long as k is large enough.
4. LPL has a high scalability because of the number of samples is limited to k.
5. Limiting the space by nearest neighbors in LPL works in a direction to worsen the approximation to the target but to increase the robustness against noise.
6. The necessary conditions are totally different. In SVM, the absolute value of $\hat{f}(\boldsymbol{x}_i)$ is not important as long as $y_i\hat{f}(\boldsymbol{x}_i) \geq 1$, while $\hat{f}(\boldsymbol{x}_i) = f(\boldsymbol{x}_i)$ has to be satisfied in PL. The latter is stronger than the former.

7 Approximate Nearest Neighbors

One advantage of LPL is to use k nearest neighbors for obtaining a scalability. However, when we want the exact k nearest neighbors, we need linear time both in dimensionality and in data size. Most sophisticated algorithms cannot beat this complexity in high dimensions. Therefore, for recent years, approximate nearest neighbors or probably correct nearest neighbors are gathering much attention [8,9]. In such a relaxation, the computational cost can be greatly reduced, usually sub-linear in sample size. Fortunately, in LPL, we do not necessary need the exact k nearest neighbors and suboptimal k nearest neighbors are acceptable. So, we can use such efficient techniques. We will use ANN [8] in this study. Its time complexity in search phase is $O(c_{p,\eta} \log \ell)$ with $c_{p,\eta} \leq p\lceil 1 + 6p/\eta \rceil^p$, where $\eta \geq 0$ is an approximation parameter.

8 Complexity

Time complexities of these algorithms are compared in Table 1. Usually the number t of support vectors is nearly proportional to the number ℓ of training samples. In Table 1, the number k of nearest neighbors in LPL is ignored because k is small enough compared with ℓ. The complexity of LPL comes from that of ANN to find k nearest neighbors. The other cost is quite low in LPL.

Table 1. Time complexity. Here, ℓ is the number of training samples, p is the dimensionality and t is the number of support vectors.

Phase	PL	LPL	SVM
Training	$O(\ell^2 p)$	$O(\ell p \log \ell)$	$O(\ell t p)$
Testing	$O(\ell p)$	$O(p \log \ell)$	$O(tp)$

Table 2. Statistics of datasets

Dataset	# samples (c_1, c_2)	Dim.
heart-statlog	270 (150, 120)	13
ionosphere	351 (225, 126)	34
sonar	208 (111, 97)	60
diabetes	768 (500, 268)	8
liver-disorders	345 (200, 145)	6
spambase	4601 (2788, 1813)	57

9 Experiments

We carried out experiments using one synthetic dataset and six real-life two-class datasets taken from UCI machine learning repository [10] (Table 2). The synthetic dataset is of two classes in 2-dimensional space (Fig. 3).

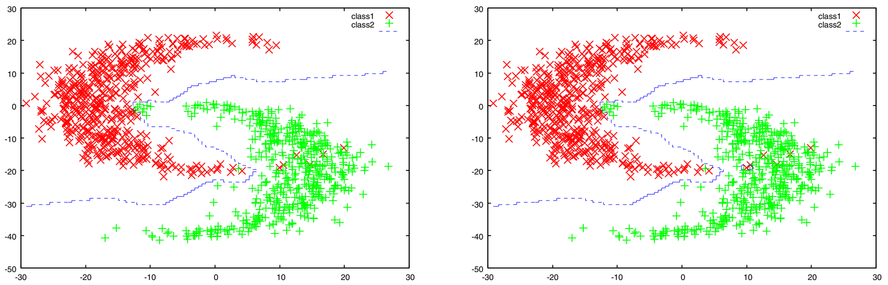
We compared PL, LPL and SVM. The SVM is the one implemented by **libsvm** [11] with a soft margin parameter $C = 1$. The kernel function is a Gaussian with a standard deviation σ. The number of nearest neighbors k in LPL was chosen to $k = \lfloor \log_{10} \ell + 1 \rfloor$ in order to simulate a consistent nature of k nearest neighbors, that is, k-NN approaches to Bayes classifier with $k = o(n)$. In ANN, we have used $\eta = 0.0$, that is, we found the exact nearest neighbors. Even if $\eta = 0.0$, the computational cost is known to be sublinear in ℓ for a low-dimensional space. The recognition rate was evaluated by 10-fold cross validation technique.

9.1 Comparison of PL and LPL

First we compared PL and LPL in performance and in time. The result for the synthetic dataset showed that the performance is almost the same in range $\sigma \in [0.01, 10]$. Indeed, there is almost no difference between the decision boundary of PL and that of LPL (Fig. 3). As for the time, LPL was faster than PL in both of training and testing phases (Fig. 4). This is consistent with the theoretical analysis in Table 1. Even in real-life datasets, this tendency was observed. As seen in Fig. 5, the recognition rates of LPL were comparative with those of PL, while the time for testing was greatly reduced as described later. Therefore, we will mainly compare LPL and SVM in the following.

9.2 Comparison of LPL and SVM

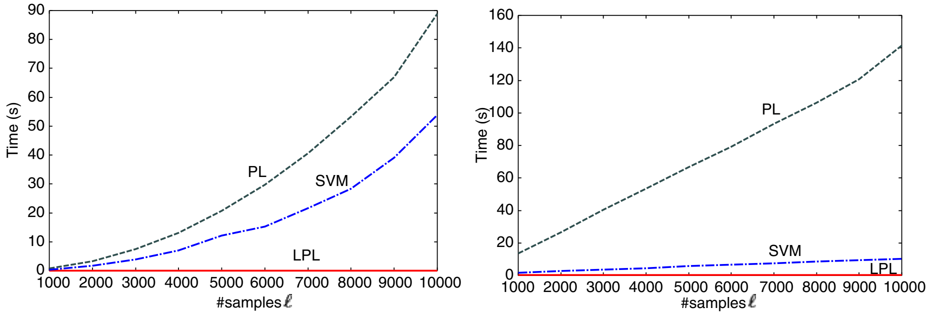
In comparison between LPL and SVM, the difference of time complexities appeared clearly in the synthetic data (Fig. 4). LPL is the fastest among three algorithms in both of training and testing phases. Even for **spambase**, which is the largest in sample size, LPL was remarkable faster than SVM. Indeed, the training time and testing time were 0.019 and 0.027 seconds, respectively, in LPL, while 6.461 and 0.632 seconds, respectively, in SVM. For the other datasets, an obvious difference was not observed because of the short time.



(a)PL

(b)LPL

Fig. 3. Decision boundaries of PL and LPL ($\sigma = 1.0$)



(a) Training time

(b) Testing time

Fig. 4. Time consumed by PL and LPL on a synthetic dataset. The curve of LPL is almost identical to the horizontal axis.

In performance, we have to be careful about the values of parameters, especially σ for Gaussian kernel. We have compared LPL and SVM in the same $\sigma = 1.0$ for time comparison. However, the optimal value of σ should be different in LPL and SVM. Therefore, we chose the best value σ^* in the range of $\sigma \in [0.01, 100]$ at 39 values in a log scale.

As shown in Fig. 5, with their optimal values, SVM was superior to LPL in most cases. This maybe implies that the separation criterion (largest margin criterion) employed in SVM is better than the approximation criterion (closest criterion) for classification problems.

9.3 Robustness

Although SVM was better than LPL with the optimal σ^*, such an optimal setting is time consuming and is sometimes even difficult to be made. So we examined how sensitive LPL and SVM are against the change of σ. In Fig. 6, a result is shown for **sonar** dataset. We can see that SVM shows a high performance only in a narrow range of σ and that LPL has a larger range for it and PL follows. The difference of robustness is also confirmed from a comparison of two cases of

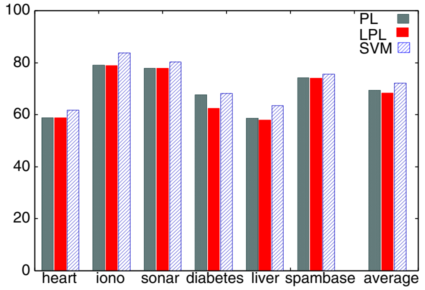


Fig. 5. Recognition rate of PL, LPL and SVM with optimal σ^*

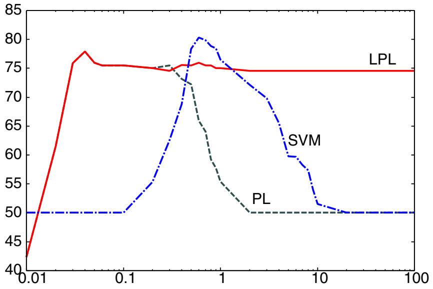
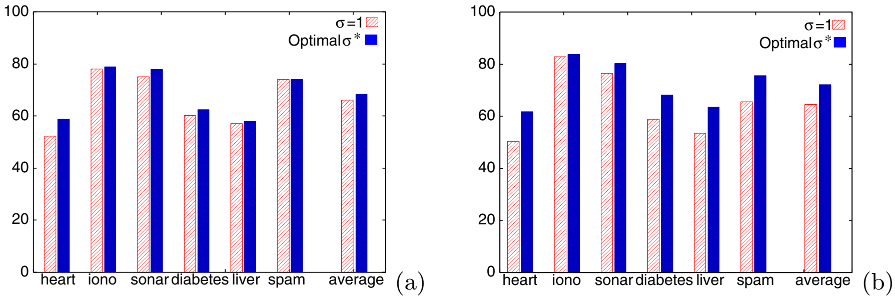


Fig. 6. Comparison of recognition rates for $\sigma \in [0.01, 100]$ on **sonar**



LPL

SVM

Fig. 7. Recognition rates for optimal σ^* and $\sigma = 1$

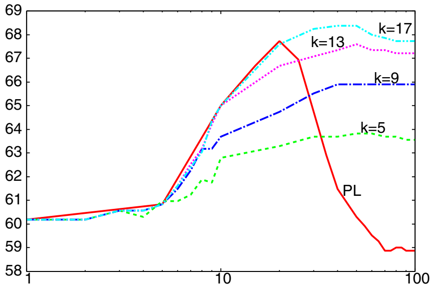


Fig. 8. Recognition rate of LPL for $k = 5, 9, 13, 17$ and $\sigma \in [1, 100]$ on **diabetes**

$\sigma = 1$ and $\sigma = \sigma^*$. In Fig. 7, we can see that the difference of LPL is smaller than that of SVM. It means that SVM is more sensitive than LPL.

The value of parameter k is also important in LPL, because the value determines the samples that affect the decision of a given sample. In **deabetes**, in which LPL was worse than SVM in performance largely, we have changed the value of k in $k = 5, 9, 13, 17$ (our default setting was $k = 3$) (Fig.8). From Fig.8, we can see that a better performance of LPL can be obtained with larger values of k. With $k = 17$, the performance of LPL is comparable with that of SVM. As a result, the optimal choice of k would improve the performance of LPL.

10 Conclusion

In this paper, scalability was brought to projection learning (PL) by localizing it with nearest neighbors of a given sample to be classified. First, it was confirmed that the classification performance is almost maintained by this localization and that the testing speed is rather improved, as it is sublinear in the number of training samples. In comparison of performance between SVM and LPL (localized PL), SVM was a little superior to LPL with the optimally chosen parameters. This might imply that the criterion of SVM is better than that of LPL for classification problems. However, we need more investigation. On the contrary, it was revealed that LPL is faster than SVM and is more robust than SVM against the parameter change. One attractive point of LPL is that it allows us to use approximate nearest neighbors so that we can use efficient algorithms even in high-dimensional problems.

One of the future studies is to compare the performance in multi-class problems. In that case, several ways of giving dummy quantitative variables have to be considered.

References

1. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1999)
2. Ogawa, H.: Neural Networks and Generalization Ability. IEICE Technical Report NC95-8, 57–64 (1995)
3. Aronszajn, N.: Theory of Reproducing Kernels. Transactions of the American Mathematical Society 68, 337–404 (1950)
4. Mercer, J.: Functions of Positive and Negative Type and Their Connection with The Theory of Integral Equations. Transactions of the London Philosophical Society A, 415–446 (1909)
5. Schatten, R.: Norm Ideals of Completely Continuous Operators. Springer, Berlin (1960)
6. Tanaka, A., Imai, H., Kudo, M., Miyakoshi, M.: Optimal Kernel in a Class of Kernels with an Invariant Metric. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 530–539. Springer, Heidelberg (2008)
7. Tanaka, A., Imai, H., Kudo, M., Miyakoshi, M.: Relationship Between Generalization Error and Training Samples in Kernel Regressors. To appear in ICPR 2010 (2010)
8. Arya, S., et al.: An optimal algorithm for approximate nearest neighbor searching fixed dimensions. Journal of the ACM 45-6, 891–923 (1998), http://www.cs.umd.edu/~mount/ANN/
9. Indyk, P., Motwani, R.: Approximate nearest neighbors: towards removing the curse of dimensionality. In: Proceedings of the 30th Annual ACM Symposium on Theory of Computing, pp. 604–613 (1998)
10. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
11. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines, http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Entropy-Based Variational Scheme for Fast Bayes Learning of Gaussian Mixtures*

Antonio Peñalver[1], Francisco Escolano[2], and Boyan Bonev[2]

[1] Miguel Hernández University, Elche, Spain
[2] University of Alicante, Spain

Abstract. In this paper, we propose a fast entropy-based variational scheme for learning Gaussian mixtures. The key element of the proposal is to exploit the incremental learning approach to perform model selection through efficient iteration over the Variational Bayes (VB) optimization step in a way that the number of splits is minimized. In order to minimize the number of splits we only select for spliting the worse kernel in terms of evaluating its entropy. Recent Gaussian mixture learning proposals suggest the use of that mechanism if a bypass entropy estimator is available. Here we will exploit the recently proposed Leonenko estimator. Our experimental results, both in 2D and in higher dimension show the effectiveness of the approach which reduces an order of magnitude the computational cost of the state-of-the-art incremental component learners.

1 Introduction

Mixture models, in particular those that use Gaussian kernels, have been widely used in areas involving statistical modeling of data like pattern recognition, computer vision, image analysis or complex probability density functions (pdfs) approximation. In statistical pattern recognition, mixture models provide a formal approach for clustering [1][2]. Mixtures model the data as being generated by one of a set of kernels. The estimation of the parameters of each kernel leads to a clustering of the data set. Whereas traditional clustering methods are based on heuristics (e.g. k-means algorithm) or hierarchical agglomerative techniques [3], mixture models allow us to address the problem of validating the parameters of a given model in a formal way. Mixture models are also suitable for representing complex class-conditional pdfs in Bayesian supervised learning scenarios [4][5] or Bayesian parameter estimation [6]. The task of estimating the parameters of a given mixture can be achieved with different approaches: maximum likelihood, maximum a posteriori (MAP) or Bayesian inference [7].

The same is true for the Bayesian Maximum a Posteriori (MAP) estimation approach that tries to find the parameters that correspond to the location of the MAP density function, and it is used when this density cannot be obtained directly [8]. Bayesian inference models the *a posteriori* parameter probability distribution, so it is assumed that the parameters are not uniquely described and they are modeled by probability density functions (pdfs) [7]. Thus, an additional set of hyperparameters is required in order to model the distribution of parameters. Then, the *a posteriori* probability of the data

* Funded by the Project TIN2008- 04416/TIN of the Spanish Government.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 100–108, 2010.

set is obtained by integration over the probability distribution of the parameters. The task of defining proper distribution functions for parameters can be computationally heavy and may result in intractable integrals. There are some approaches that try to solve those drawbacks: Laplacian method [9], Markov Chain Monte Carlo (MCMC) [10], and Variational methods [11][12]. Laplacian methods employ an approximation based on the Taylor expansion for the expression of the integrals [9]. However, in high dimensional contexts this approach is computationally expensive and may provide poor approximation results. MCMC methods require both an appropriate distribution selection and sampling techniques in order to draw suitable data samples. Besides, due to their stochastic nature, MCMC algorithms may require a long time to converge [10]. Variational algorithms are guaranteed to provide a lower bound of the approximation error [11]. In most approaches, parameter initialization is selected randomly, defined over a given range of values, but it could lead to overfitting and poor generalization [13]. Although the results show good performance in clustering, blind signal detection or color image segmentation, the computational complexity of the Variational EM algorithm is higher than the classic EM with the maximum likelihood criterion. However, variational methods are more suitable than EM-MDL based methods [14][15][8][16] for model-order selection. Thus in this paper, we propose an fast extension of the Variational Bayes (BV) method proposed in [17] for inferring Gaussian mixtures and solving the model-order selection problem.

2 Variational Bayes for Mixtures

Given N i.i.d. samples $\mathcal{X} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\}$ of a d-dimensional random variable X, their associated hidden variables $Z = \{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^N\}$ and the parameters Θ of the model, the Bayesian posterior is given by [18]:

$$p(Z, \Theta|X) = \frac{p(\Theta)\prod_{n=1}^{N} p(\boldsymbol{x}^n, \boldsymbol{z}^n|\Theta)}{\int p(\Theta)\prod_{n=1}^{N} p(\boldsymbol{x}^n, \boldsymbol{z}^n|\Theta)d\Theta} \ . \tag{1}$$

Since the integration w.r.t. Θ is analytically intractable, the posterior is approximated by a factorized distribution $q(Z, \Theta) = q(Z)q(\Theta)$ and the optimal approximation is the one that minimizes the variational free energy:

$$\mathcal{L}(q) = \int q(Z, \Theta) \log \frac{q(Z, \Theta)}{p(Z, \Theta|X)} d\Theta - \log \int p(\Theta) \prod_{n=1}^{N} p(\boldsymbol{x}^n|\theta) d\Theta \ , \tag{2}$$

where the first term is the Kullback-Leibler divergence between the approximation and the true posterior. As the second term is independent of the approximation, the Variational Bayes (VB) approach is reduced to minimize the latter divergence. Such minimization is addressed in a EM-like process alternating the updating of $q(\Theta)$ and the updating of $q(Z)$ [19]:

$$q(\Theta) \propto p(\Theta) \exp\left\{ \sum_{n=1}^{N} \langle \log p(\boldsymbol{x}^n, \boldsymbol{z}^n|\Theta) \rangle_{q(Z)} \right\} \tag{3}$$

$$q(Z) \propto \exp\left\{ \sum_{n=1}^{N} \langle \log p(\boldsymbol{x}^n, \boldsymbol{z}^n|\Theta) \rangle_{q(\Theta)} \right\} \tag{4}$$

When the posterior is modeled by a mixture we have that

$$p(X|\Omega) = \sum_{k=1}^{K} \pi_k p(X|\Omega_k), \tag{5}$$

where $0 \leq \pi_k \leq 1$, $k = 1, \dots, K$, $\sum_{k=1}^{K} \pi_k = 1$, K is the number of kernels, $\pi_1, \dots, \pi_K$ are the a priori probabilities of each kernel, and Ω_k are the parameters that describe the kernel. In Gaussian mixtures, $\Omega_k = \{\mu_k, \Sigma_k\}$, that is, the mean vector and the covariance matrix. Consequently we have

$$p(X, Z|\Theta) = \prod_{n=1}^{N} \prod_{k=1}^{K} z_k^n \pi_k p(\boldsymbol{x}^n|\Omega_k). \tag{6}$$

where $z^i = [z_1^n, \dots, z_K^n]$ is a binary vector and $z_m^n = 1$ and $z_p^n = 0$, if $p \neq m$, denote that $\boldsymbol{x}^n$ has been generated by the kernel m. Then, considering the complete mixture let $\mu = \{\mu_k\}$, $\Sigma = \{\Sigma_k\}$, $\pi = \{\pi_k\}$ and K the parameters of the model, that is, $\Theta = \{\mu, \Sigma, \pi, K\}$. Including in the parameter set the number of mixtures implies dealing with the problem of model order selection (obtain the optimal K). In [17], model order selection is implicitly solved within the Bayesian approach. In the latter framework, it is assumed that a number of $K - s$ components fit the data well in their region of influence (*fixed components*) and then model order selection is posed in terms of optimizing the parameters of the remaing s (*free components*). Let $\alpha = \{\pi_k\}_{k=1}^{s}$ the coefficients of the free components and $\beta = \{\pi_k\}_{k=s+1}^{K}$ the coefficients of the fixed components. Obviously, the sum of coefficients in α and β must be the unit. In addition, under the i.i.d. sampling assumption, the prior distribution of Z given α and β can be modeled by a product of multinomials:

$$p(Z|\alpha, \beta) = \prod_{n=1}^{N} \prod_{k=1}^{s} {\pi_k}^{z_k^n} \prod_{k=s+1}^{K} {\pi_k}^{z_k^n} \ . \tag{7}$$

Moreover, assuming conjugate Dirichlet priors over the set of mixing coefficients, we have that

$$p(\beta|\alpha) = \left(1 - \sum_{k=1}^{s} \pi_k\right)^{-K+s} \frac{\Gamma\left(\sum_{k=s+1}^{K} \gamma_k\right)}{\prod_{k=s+1}^{K} \Gamma(\gamma_k)} \cdot \prod_{k=s+1}^{K} \left(\frac{\pi_k}{1 - \sum_{k=1}^{s} \pi_k}\right)^{\gamma_k - 1} \ . \tag{8}$$

Then, considering fixed coefficients Θ is redefined as $\Theta = \{\mu, \Sigma, \beta\}$ and we have the following factorization:

$$q(Z, \Theta) = q(Z) q(\mu) q(\Sigma) q(\beta) \ . \tag{9}$$

Then, in [17], the optimization of the variational free energy yields:

$$q(Z) = \prod_{n=1}^{N} \prod_{k=1}^{s} {r_{k^n}}^{z_k^n} \prod_{k=s+1}^{K} {\rho_{k^n}}^{z_k^n} \tag{10}$$

$$q(\mu) = \prod_{k=1}^{K} \mathcal{N}(\mu_k | m_k, \Sigma_k) \tag{11}$$

$$q(\Sigma) = \prod_{k=1}^{K} \mathcal{W}(\Sigma_k | \nu_k, V_k) \tag{12}$$

$$q(\beta) = \left(1 - \sum_{k=1}^{s} \pi_k\right)^{-K+s} \frac{\Gamma\left(\sum_{k=s+1}^{K} \tilde{\gamma}_k\right)}{\prod_{k=s+1}^{K} \Gamma(\tilde{\gamma}_k)} \cdot \prod_{k=s+1}^{K} \left(\frac{\pi_k}{1 - \sum_{k=1}^{s} \pi_k}\right)^{\tilde{\gamma}_k - 1} \tag{13}$$

where $\mathcal{N}(.)$ and $\mathcal{W}(.)$ are respectively the Gaussian and Wishart densities, and the rest of parameters are obtained as specified in [17]. Furthermore, after the maximization of the free energy w.r.t. $q(.)$, it proceeds to update the coefficients in α. This interwinted process is repeated until convergence. However, how is model selection solved within this approach?

3 Model Order Selection in VB: The EBVS Approach

An incremental model order selection algorithm starts from a small number of components (one or two) and proceeds to split them until convergence. For instance, in [16] a unique initial kernel is used. However in [17] the VBgmm method [20] is used for training an initial $K = 2$ model. Then, in the so called VBgmmSplit, they proceed by sorting the obtained kernels and then trying to split them recursively. Each splitting consists of: (i) removing the original component, and (ii) replacing it by two kernels with the same covariance matrix as the original but with means placed in opposite directions along the maximum variabiability direction. Such direction is given by the principal axis (eigenvector ϕ) of the inverse of the original covariance matrix and the amount of displacement, and the amount of displacement is $\pm\sqrt{\lambda}\phi$, being λ the corresponding eigenvalues. If the original mixing coefficient is π the new coefficients are $\pi/2$. A more complex and robust split method is proposed in [21] and used efficiently in [16]. Independently of the split strategy, the critical point of VBgmmSplit is the *amount of splits needed until convergence*. At each iteration of the latter algorithm the K current exisiting kernels are splited. Consider the case of any split is detected as proper (non-zero π after running the VB update described in the previous section, where each new kernel is considered as *free*). Then, the number of components increases and then a new set of splitting tests starts in the next iteration. This means that if the algorithm stops (all splits failed) with K kernels, the number of splits has been $1 + 2 + \ldots + K = K(K + 1)/2$. Although the computational cost of a split is not critical, what is critical is the increasing amount of kernels considered for the VB optimization. Thus, it is important to control the number of splits because it has an important impact in the complexity of the VB optimization step. This is our main contribution in this paper, and we dubbed it the *Entropy-based Variational Scheme* (EBVS).

3.1 The EBVS Split Scheme

Instead of considering all the current kernels K at each iteration, we split only one kernel per iteration. In order to do so, we implement a selection criterion based on measuring the entropy of the kernels. According to [16], as the Gaussian distribution maximizes entropy among all the distributions with the same covariance, the lower the entropy of a kernel, the more suitable it is for being split. The main problem of this approach is the fact that entropy must be estimated and this may be a very difficult task if data dimensionality d is high (curse of dimensionality) if a bypass entropy estimator (no need to estimate the probability density function) is not used. For instance, in [16], the Entropic Graphs based estimator [22] is extrapolated from Rényi entropy to the Shannon one. However, if ones uses the recently proposed Leonenko's estimator [23] (see above) then there is no need of extrapolation, and asymptotic consistence is ensured. This is the entropy estimator used in this paper. Then, at each iteration of the algorithm we select the *worse*, in terms of low entropy, to be split. If the split is successful we will have $K+1$ kernels to feed the VB optimization in the next iteration. Otherwise, there is no need to add a new kernel and the process converges to K kernels. The key question here is that the overall process is linear (one split per iteration) with the number of kernels instead of being quadratic.

3.2 Entropy Estimation

A simple way to understand the k-NN entropy estimation proposed by Leonenko [23] is to look at the Shannon entropy formula $H(X) = -\int f(x)\log f(x)dx$, as an average of $\log f(x)$, being $f(x)$ an existing pdf. The estimation of $\widehat{\log f(x)}$ would allow the estimation of $\hat{H}(X) = -N^{-1}\sum_{i=1}^{N}\widehat{\log f(x)}$. For this purpose the probability distribution $P_k(\epsilon)$ of the distance between a sample x_i and its k-NN is considered. If a ball of diameter ϵ is centered at x_i and there is a point within distance $\epsilon/2$, then there are $k-1$ other points closer to x_i and $N-k-1$ points farther from it. The probability of this to happen is $P_k(\epsilon)d\epsilon = k\binom{N-1}{k}\frac{dp_i(\epsilon)}{d\epsilon}p_i^{k-1}(1-p_i)^{N-k-1}$ being p_i the mass of the ϵ-ball and $p_i(\epsilon) = \int_{||\xi-x_i||<\epsilon/2} f(\xi)d\xi$.

The expectation of of $\log p_i(\epsilon)$ is $E(\log p_i) = \int_0^\infty P_k(\epsilon)\log p_i(\epsilon)d\epsilon$ that is $= k\binom{N-1}{k}\int_0^1 p^{k-1}(1-p)^{N-k-1}\log p\cdot dp = \psi(k)-\psi(N)$, where $\psi(\cdot)$ is the well-known digamma function. If assumed that $f(x)$ is constant in the entire ϵ-ball, then the approximation $p_i(\epsilon) \approx \frac{V_d}{2^d}\epsilon^d\mu(x_i)$ can be formulated. Here d is the dimension and V_d is the volume of the unit ball $\mathcal{B}(0,1)$, defined as $V_d = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}$. From the previous approximation and using the expectation of $\log p_i(\epsilon)$, we have the approximation $\log f(\epsilon) \approx \psi(k)-\psi(N)-dE(\log\epsilon)-\log\frac{V_d}{2^d}$, and finally,

$$\hat{H}(X) = -\psi(k)+\psi(N)+\log\frac{V_d}{2^d}+\frac{d}{N}\sum_{i=1}^{N}\log\epsilon_i \tag{14}$$

is the estimation of $H(X)$, where $\epsilon_i = 2||x_i - x_j||$ is twice the distance between the sample x_i and its k-NN x_j. It is suggested that the error for Gaussian and uniform distributions is $\sim k/N$ or $\sim k/N\log(N/k)$.

4 Experiments

We present serveral experiments in order to show the performance of our method. We have tested the algorithm on both synthetic and real data.

4.1 Simple Densities

In this first experiment we have generated $2,500$ samples from five bidimensional Gaussians with different prior probabilities, averages and covariance matrices. However the distributions do not overlap and are well separated. Fig. 1 shows the estimations of the mixtures and the final Bayesian classification of the samples. The parameters of the mixture of this experiment are:

$$\Sigma_1 = \begin{bmatrix} 0.20 & 0.00 \\ 0.00 & 0.30 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 0.60 & 0.15 \\ 0.15 & 0.60 \end{bmatrix},$$

$$\Sigma_3 = \begin{bmatrix} 0.40 & 0.00 \\ 0.00 & 0.25 \end{bmatrix}, \Sigma_4 = \begin{bmatrix} 0.60 & 0.00 \\ 0.00 & 0.30 \end{bmatrix},$$

$$\Sigma_5 = \begin{bmatrix} 0.20 & 0.00 \\ 0.00 & 0.30 \end{bmatrix},$$

$$\begin{gathered} \pi_k = 0.2, \\ \mu_1 = [-1,-1]^T, \mu_2 = [6,3]^T, \mu_3 = [3,6]^T, \\ \mu_4 = [2,2]^T, \mu_5 = [0,0]^T. \end{gathered} \tag{15}$$

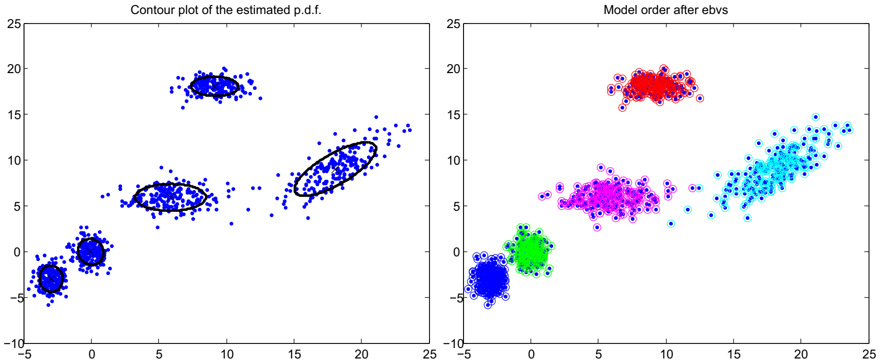


Fig. 1. First experiment: Easy density estimation

4.2 Overlapping Densities

This experiment presents the problem of having overlaping densities. The method show a sucessful density estimation and classification. We generated $1,000$ samples from four

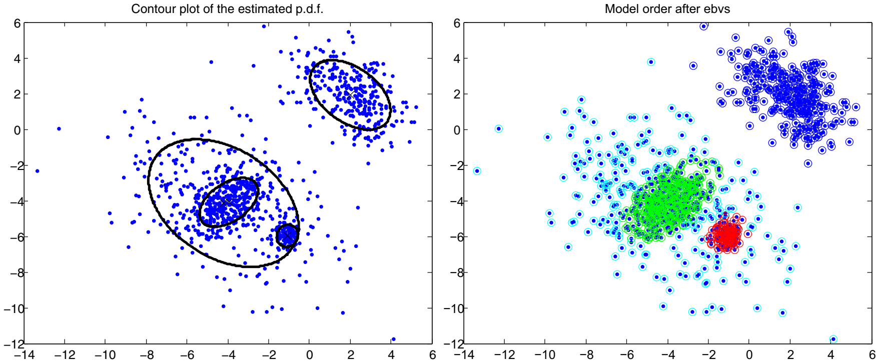


Fig. 2. Overlaped components experiment

bidimensional Gaussians with different prior probabilities, averages and covariance matrices. Fig. 2 shows the estimations of the mixtures and the final Bayesian classification of the samples. The mixture parameteres are the following:

$$\Sigma_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 6 & -2 \\ -2 & 6 \end{bmatrix},$$

$$\Sigma_3 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \Sigma_4 = \begin{bmatrix} 0.125 & 0 \\ 0 & 0.125 \end{bmatrix},$$

$$\begin{gathered} \pi_1 = \pi_2 = \pi_3 = 0.3, \\ \pi_4 = 0.1, \\ \mu_1 = \mu_2 = [-4,-4]^T, \mu_3 = [2,2]^T, \mu_4 = [-1,-6]^T. \end{gathered} \tag{16}$$

This experiment is used in [16] to argument that their proposed incremental improves the results obtained in [8] (in both cases a MDL criterion is used for model selection). Then our method produces also a good result in this case.

4.3 Symmetric Densities

This experiment presents the problem of having symmetric densities. One Gaussian is placed in the center and four symmetric Gaussians, with symmetric covariances, are placed around the center. We generated $1,000$ samples from this mixture. Fig. 3 shows the successful estimations of the mixtures and the final Bayesian classification of the samples. The mixture parameteres are the following:

$$\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \Sigma_2 = \Sigma_4 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \Sigma_3 = \Sigma_5 = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix},$$

$$\begin{gathered} \pi_k = 0.2, \\ \mu_1 = [0,0]^T, \mu_2 = [3,-3]^T, \mu_3 = [3,3]^T, \mu_4 = [-3,3]^T, \mu_5 = [-3,-3]^T. \end{gathered} \tag{17}$$

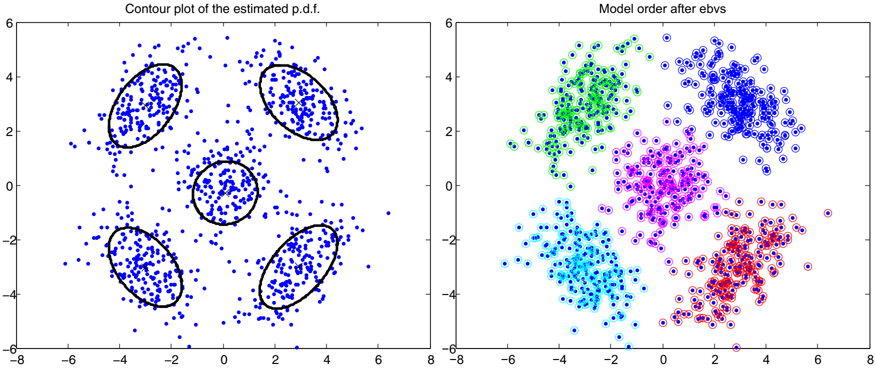


Fig. 3. Experiment with symmetric densities

4.4 Real Data

We have experimented with real data in the context of unsupervised pattern classification. The data set we tested has a relatively high number of dimensions. The well-known *Wine* data set contains three classes of 178 samples of 13 dimensions. This data set comes from chemical analysis of wines grown in different cultivars from the same region, in Italy. The dimensions correspond to 13 constituents found in each one of the three types (classes) of wines. The data set is preprocessed in order to have zero mean and unit variance in each one of the dimensions. The classification performance we obtain on this data set is 86%.

Altough experiments in higher dimensions can be performed, when the number of samples is not high enough, the risk of unbounded maxima of the likelihood function is higher, due to singular covariance matrices. The entropy estimation method, however, performs very well with thousands of dimensions.

5 Conclusions and Future Work

In this paper we have proposed a significant improvement, in terms of computational complexity, of the VBgmmSplit method. Such an improvement relies on the reduction of the quadratic number of splits per iteration to a linear one and this is key for reducing the complexity of the VB optimization method. Splits are reduced by selecting only the worse (lowest entropy) kernel to split and entropy estimation is addressed through a recently proposed bypass method. Our future work includes the extension to other kind of mixtures as well as the incorporation of a more robust/reliable split strategy.

References

1. Jain, A., Dubes, R., Mao, J.: Statistical pattern recognition: A review. IEEE Trans. Pattern Anal. Mach. Intell. 22(1), 4–38 (2000)
2. Titterington, D., Smith, A., Makov, U.: Statistical Analysis of Finite Mixture Distributions. John Wiley and Sons, Chichester (2002)

3. Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs (1988)
4. Hastie, T., Tibshirani, R.: Discriminant analysis by gaussian mixtures. Journal of The Royal Statistical Society(B) 58(1), 155–176 (1996)
5. Hinton, G., Dayan, P., Revow, M.: Modeling the manifolds of images of handwriting digits. IEEE Transactions On Neural Networks 8(1), 65–74 (1997)
6. Dalal, S., Hall, W.: Approximating priors by mixtures of natural conjugate priors. Journal of The Royal Statistical Society(B) 45(1) (1983)
7. Box, G., Tiao, G.: Bayesian Inference in Statistical Models. Addison-Wesley, Reading (1992)
8. Figueiredo, M., Jain, A.: Unsupervised learning of finite mixture models. IEEE Trans. Pattern Anal. Mach. Intell. 24(3), 381–399 (2002)
9. Husmeier, D.: The bayesian evidence scheme for regularizing probability-density estimating neural networks. Neural Computation 12(11), 2685–2717 (2000)
10. MacKay, D.: Introduction to Monte Carlo Methods. In: Jordan, M.I. (ed.) Learning in Graphical Models. MIT Press, MA (1999)
11. Ghahramani, Z., Beal, M.: Variational inference for bayesian mixture of factor analysers. In: Adv. Neur. Inf. Proc. Sys., MIT Press, Cambridge (1999)
12. Nasios, N., Bors, A.: Variational learning for gaussian mixtures. IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics 36(4), 849–862 (2006)
13. Nasios, N., Bors, A.: Blind source separation using variational expectation-maximization algorithm. In: Petkov, N., Westenberg, M.A. (eds.) CAIP 2003. LNCS, vol. 2756, pp. 442–450. Springer, Heidelberg (2003)
14. Figueiredo, M., Leitao, J., Jain, A.: On fitting mixture models. In: Hancock, E.R., Pelillo, M. (eds.) EMMCVPR 1999. LNCS, vol. 1654, pp. 54–69. Springer, Heidelberg (1999)
15. Figueiredo, M.A.T., Jain, A.K.: Unsupervised selection and estimation of finite mixture models. In: Proc. Int. Conf. Pattern Recognition, pp. 87–90. IEEE, Los Alamitos (2000)
16. Penalver, A., Escolano, F., Sáez, J.: Learning gaussian mixture models with entropy-based criteria. IEEE Transactions on Neural Networks 20(11), 1756–1772 (2009)
17. Constantinopoulos, C., Likas, A.: Unsupervised learning of gaussian mixtures based on variational component splitting. IEEE Transactions on Neural Networks 18(3), 745–755 (2007)
18. Watanabe, K., Akaho, S., Omachi, S.: Variational bayesian mixture model on a subspace of exponential family distributions. IEEE Transactions on Neural Networks 20(11), 1783–1796 (2009)
19. Attias, H.: Inferring parameters and structure of latent variable models by variational bayes. In: Proc. of Uncertainty Artif. Intell., pp. 21–30 (1999)
20. Corduneau, A., Bishop, C.: Variational bayesian model selection for mixture distributions. In: Artificial Intelligence and Statistics, pp. 27–34. Morgan Kaufmann, San Francisco (2001)
21. Richardson, S., Green, P.: On bayesian analysis of mixtures with unknown number of components (with discussion). Journal of the Royal Statistical Society B 59(1), 731–792 (1997)
22. Hero, A., Michel, O.: Estimation of rényi information divergence via pruned minimal spanning trees. In: Workshop on Higher Order Statistics, Caessaria, Israel. IEEE, Los Alamitos (1999)
23. Leonenko, N., Pronzato, L.: A class of rényi information estimators for multi-dimensional densities. The Annals of Statistics 36(5), 2153–2182 (2008)

Learning Graph Quantization

Brijnesh J. Jain, S. Deepak Srinivasan,
Alexander Tissen, and Klaus Obermayer

Berlin Institute of Technology, Germany
jbj@cs.tu-berlin.de

Abstract. This contribution extends learning vector quantization to the domain of graphs. For this, we first identify graphs with points in some orbifold, then derive a generalized differentiable intrinsic metric, and finally extend the update rule of LVQ for generalized differentiable distance metrics. First experiments indicate that the proposed approach can perform comparable to state-of-the-art methods in structural pattern recognition.

1 Introduction

Learning vector quantization (LVQ) as introduced by Kohonen [11] is a supervised learning algorithm for pattern classification. To classify patterns, LVQ applies the nearest neighbor rule using a condensed set of prototypes. Prototypes are learned by combining competitive learning with supervision. LVQ is easy to implement, runs efficiently, allows to control the complexity of the resulting classifier, naturally deals with multiclass problems, provides an interpretable rather than a block-box model, and in many cases provides state of the art performance.

LVQ and related methods have been originally devised for feature vectors equipped with the Euclidean metric. Extensions have been proposed, for example, for vectors with arbitrarily differentiable distance functions [8], for variable length and warped feature sequences [18], and for strings [12]. But there have been no efforts reported towards extending LVQ for the domain of attributed graphs, although there are a number of related unsupervised methods that extend competitive learning for central clustering to structured data [7,6,9],

In this contribution, we generalize LVQ to learning graph quantization (LGQ). The challenge consists in formulating an update rule for prototype adaption. For differentiable metrics on vectors such as the Euclidean metric, adaption amounts in moving prototypes along the line determined by the gradient of the underlying distance metric. Graph distance metrics, however, are not differentiable in general and therefore local gradient information is unavailable for prototype adaption. To overcome this problem, an appropriate approach to represent graphs is necessary. As such an approach, we suggest to represent graphs as points in some Riemannian orbifold. An orbifold is a quotient of a manifold by a finite group action. Using orbifolds, we derive an intrinsic metric that enables us to adopt concepts such as the derivative and gradient. Since the intrinsic metric

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 109–118, 2010.

of a graph orbifold is generalized differentiable, we can apply local gradient information for adapting graph prototypes almost everywhere. At this point, it is important to note that the intrinsic metric is not an artificial construction for analytical purposes but rather a common choice of graph distance metric in a number of applications [1,2,3,5,19,21]. Experiments on three data sets of the IAM graph database [15] show that the proposed LGQ approach returns state-of-the-art results.

The approach presented in this contribution can be applied to finite combinatorial structures other than graphs such as, for example, point patterns, sequences, trees, and hypergraphs. For the sake of concreteness, we restrict our attention exclusively to the domain of graphs. For graphs consisting of a single vertex with feature vectors as attributes, the proposed learning graph quantization (LGQ) reduces to LVQ.

This paper is organized as follows. Section 2 represents graphs as point in some orbifold. Section 3 extends LVQ to LGQ. In Section 4, we present and discuss experiments. Finally, Section 5 concludes.

2 Graph Orbifolds

Crucial for designing pattern classification algorithms is an appropriate representation of the data space. We suggest to represent attributed graphs as points in some graph orbifold. A graph orbifold is the simplest form of a Riemannian orbifold. For proofs of the statements in this section we refer to [10].

2.1 Representation of Attributed Graphs

Let $\mathbb{E} = \mathbb{R}^d$ be a Euclidean space. An *attributed graph* is a triple $X = (V, E, \alpha)$ consisting of a set V of *vertices*, a set $E \subseteq V \times V$ of *edges*, and an *attribute function* $\alpha : V \times V \rightarrow \mathbb{E}$, such that $\alpha(i, j) \neq \mathbf{0}$ for each edge and $\alpha(i, j) = \mathbf{0}$ for each non-edge. Attributes $\alpha(i, i)$ of vertices i may take any value from $\mathbb{E}$.

For simplifying the mathematical treatment, we assume that all graphs are of order n, where n is chosen sufficiently large. Graphs of order less than n, say $m < n$, can be extended to order n by including isolated vertices with attribute zero. For practical issues, it is important to note that limiting the maximum order to some arbitrarily large number n and extending smaller graphs to graphs of order n are purely technical assumptions to simplify mathematics. For pattern recognition problems, these limitations should have no practical impact, because neither the bound n needs to be specified explicitly nor an extension of all graphs to an identical order needs to be performed. When applying the theory, all we actually require is that the graphs are finite.

A graph X is completely specified by its *matrix representation* $\boldsymbol{X} = (\boldsymbol{x}_{ij})$ with elements $\boldsymbol{x}_{ij} = \alpha(i, j)$ for all $1 \leq i, j \leq n$. By concatenating the columns of $\boldsymbol{X}$, we obtain a *vector representation* $\boldsymbol{x}$ of X.

Let $\mathcal{X} = \mathbb{E}^{n \times n}$ be the Euclidean space of all $(n \times n)$-matrices with elements from $\mathbb{E}$ and let $\mathcal{T}$ denote a subgroup of all $(n \times n)$-permutation matrices. Two matrices $\boldsymbol{X}, \boldsymbol{X'} \in \mathcal{X}$ are said to be equivalent, if there is a permutation matrix $P \in \mathcal{T}$ such that $\boldsymbol{P}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{P} = \boldsymbol{X'}$. The quotient set

$$\mathcal{X}_{\mathcal{T}} = \mathcal{X}/\mathcal{T} = \{[\boldsymbol{X}] \,:\, \boldsymbol{X} \in \mathcal{X}\}$$

consisting of all equivalence classes $[\boldsymbol{X}]$ is a *graph orbifold* over the *representation space* $\mathcal{X}$. Its *orbifold chart* is the surjective continuous mapping

$$\pi : \mathcal{X} \to \mathcal{X}_{\mathcal{T}}, \quad \boldsymbol{X} \mapsto [\boldsymbol{X}]$$

that projects each point $\boldsymbol{X}$ to its orbit $[\boldsymbol{X}]$. A graph orbifold is the simplest form of a so called *Riemannian orbifold.*

In the following, we identify $\mathcal{X}$ with $\mathbb{E}^N$ $(N = n^2)$ and consider vector- rather than matrix representations of abstract graphs. We use capital letters $X, Y, Z, \ldots$ to denote graphs from $\mathcal{X}_{\mathcal{T}}$ and write $\boldsymbol{x} \in X$ if $\boldsymbol{x}$ is a vector representation that projects to X (i.e. if $\pi(\boldsymbol{x}) = X$). Since $\mathbb{E}$ is Euclidean so is $\mathcal{X}$. By $\|\cdot\|$ we denote the Euclidean norm defined on $\mathcal{X}$.

2.2 Generalized Differentiable Graph Metric

Next, we introduce and analyze an intrinsic metric structure on graph orbifolds. For graphs with discrete attributes the intrinsic metric is related to the concept of maximum common subgraph.

The Graph Metric. We consider graph metrics of the form

$$d(X, Y) = \min\left\{\|\boldsymbol{x} - \boldsymbol{y}\|^2 \,:\, \boldsymbol{x} \in X, \boldsymbol{y} \in Y\right\}.$$

A pair $\boldsymbol{x}, \boldsymbol{y} \in X \times Y$ is an *optimal alignment* if $d(X, Y) = \|\boldsymbol{x} - \boldsymbol{y}\|^2$. By $\mathcal{A}(X, Y)$ we denote the set of all optimal alignments of X and Y.

Since $\mathcal{T}$ is a group, we have

$$d_X(Y) = \min\left\{\|\boldsymbol{x} - \boldsymbol{y}\|^2 \,:\, \boldsymbol{y} \in Y\right\} = d(X, Y),$$

where $\boldsymbol{x} \in X$ is an arbitrary vector representation. Hence, the graph distance $d(X, Y)$ can be determined by fixing an arbitrary vector representation $\boldsymbol{x} \in X$ and then finding a vector representation $\boldsymbol{y}_* \in Y$ that minimizes $\|\boldsymbol{x} - \boldsymbol{y}\|^2$ over all vector representations $\boldsymbol{y} \in Y$. Note that we also have $d_X(Y) = d(Y, X)$ by symmetry.

Generalized Differentiability. The *lift* $\tilde{d}_X$ of the parametrized graph distance function d_X is defined by

$$\tilde{d}_X : \mathcal{X} \to \mathbb{R}, \quad \boldsymbol{y} \mapsto \min\left\{\|\boldsymbol{x} - \boldsymbol{y}'\|^2 \,:\, \boldsymbol{y}' \in Y\right\}.$$

where $\boldsymbol{x} \in X$ is an arbitrary vector representation. Certainly, the lift $\tilde{d}_X$ satisfies $\tilde{d}_X = d_X \circ \pi$ and is invariant under group actions of $\mathcal{T}$, that is $\tilde{d}_X(\boldsymbol{y}) = \tilde{d}_X\,(\gamma(\boldsymbol{y}))$ for all $\gamma \in \mathcal{T}$.

By lifting the distance function d_X to the Euclidean space $\mathcal{X}$, we are in the position to transfer analytical concepts such as differentiability and gradients to functions on graph orbifolds. We say, the function d_X is continuous (locally Lipschitz, differentiable, generalized differentiable) at point $Y \in \mathcal{X}_\mathcal{T}$ if its lift $\tilde{d}_X$ is continuous (locally Lipschitz, differentiable, generalized differentiable) at some vector representation $\boldsymbol{y} \in Y$.[1] This definition is independent of the choice of vector representations that project to X and Y.

As a minimizer of a set of continuously differentiable distance functions, the function d_X is generalized differentiable at any point Y. Though d_X is not differentiable, it is locally Lipschitz and therefore differentiable almost everywhere.

Gradients. Suppose that d_X is differentiable at Y and let $\boldsymbol{x} \in X$ be arbitrary. Then the lift $\tilde{d}_X$ is differentiable at any vector representation that projects to Y. The gradient $\nabla\tilde{d}_X(\boldsymbol{y})$ of $\tilde{d}_X$ at $\boldsymbol{y}$ is of the form

$$\nabla\tilde{d}_X(\boldsymbol{y}) = -2(\boldsymbol{x} - \boldsymbol{y}_*)$$

where $(\boldsymbol{x}, \boldsymbol{y}_*) \in \mathcal{A}(X, Y)$ is an optimal alignment. Since d_X is differentiable at Y, the optimal alignment $(\boldsymbol{x}, \boldsymbol{y}_*)$ is unique. From

$$\nabla\tilde{d}_X(\gamma(\boldsymbol{y})) = \gamma\left(\nabla\tilde{d}_X(\boldsymbol{y})\right)$$

for all $\gamma \in \mathcal{T}$ follows that the gradients of $\tilde{d}_X$ at $\boldsymbol{y}$ and $\gamma(\boldsymbol{y})$ are vector representations of the same graph. Hence, at differentiable points Y, the gradient of $d_X(Y)$ at Y is defined by the projection

$$\nabla d_X(Y) = \pi\left(\nabla\tilde{d}_X(\boldsymbol{y})\right)$$

of the gradient $\nabla\tilde{d}_X(\boldsymbol{y})$ at vector representation $\boldsymbol{y} \in Y$. Thus, the gradient of d_X at Y is a well-defined graph pointing to the direction of steepest ascent.

Generalized Gradients. Now suppose that d_X is generalized differentiable at Y. Then the lift $\tilde{d}_X$ is generalized differentiable at any vector representation that projects to Y. The subdifferential $\partial\tilde{d}_X(\boldsymbol{y})$ of $\tilde{d}_X$ at $\boldsymbol{y}$ is a convex set containing

$$-2(\boldsymbol{x} - \boldsymbol{y}_*) \in \partial\tilde{d}_X(\boldsymbol{y})$$

as generalized gradient, where $(\boldsymbol{x}, \boldsymbol{y}_*) \in \mathcal{A}(X, Y)$ is an optimal alignment. From

$$\partial\tilde{d}_X(\gamma(\boldsymbol{y})) = \gamma\left(\partial\tilde{d}_X(\boldsymbol{y})\right)$$

for all $\gamma \in \mathcal{T}$ follows that the subderivatives of $\tilde{d}_X$ at $\boldsymbol{y}$ and $\gamma(\boldsymbol{y})$ project to the same subset of graphs. Hence, at generalized differentiable points Y, the subderivative of $d_X(Y)$ at Y is defined by the projection

$$\partial d_X(Y) = \pi\left(\partial\tilde{d}_X(\boldsymbol{y})\right)$$

[1] Appendix A defines generalized differentiable functions.

of the subderivative $\nabla\tilde{d}_X(\boldsymbol{y})$ at an arbitrary vector representation $\boldsymbol{y} \in Y$. Thus, the subderivative of d_X at Y is well-defined and coincides with the gradient at differentiable points, that is $\partial d_X(Y) = \{\nabla d_X(Y)\}$.

3 Learning Graph Quantization

The task of Learning Graph Quantization (LGQ) is to construct a classifier $c : \mathcal{X}_\mathcal{T} \to \mathcal{C}$ that maps graphs from $\mathcal{X}_\mathcal{T}$ to class labels from a finite set $\mathcal{C}$. The classifiers are parameterized by a set of k prototypes $Y_1, \ldots, Y_k \in \mathcal{X}_\mathcal{T}$ with class labels $c_1, \ldots, c_k \in \mathcal{C}$. We predict the class label $c(X)$ of a new graph $X \in \mathcal{X}_\mathcal{T}$ by assigning it to the class label of the closest prototype according to the nearest neighbor rule. The goal of learning is to find a set of k prototypes that best predicts the class labels of graphs from $\mathcal{X}_\mathcal{T}$. In the following, we extend LVQ and LVQ2.1 to the domain of graph orbifolds.

3.1 LGQ

Suppose that $\mathcal{S} = \{(X_i, y_i)\}_{i=1}^n \subseteq \mathcal{X}_\mathcal{T} \times \mathcal{C}$ is a training set consisting of n input graphs $X_i \in \mathcal{X}_\mathcal{T}$ together with class labels $y_i \in \mathcal{C}$. The algorithm first chooses k prototypes $\mathcal{Y} = \{(Y_j, c_j)\}_{j=1}^k$ such that each class is represented by at least one prototype. Next, during adaption, the algorithm randomly choses an example $(X, y) \in \mathcal{S}$ from the training set and modifies the closest prototype Y_X in accordance with the current example. The input graph X attracts its closest prototype Y_X if the class labels y of X and c_X of Y_X agree. Otherwise, if the class labels differ, the input X repels the closest prototype Y_X. To determine the closest prototype, LGQ applies the nearest neighbor rule

$$Y_X = \arg\min_{Y \in \mathcal{Y}} \{d(X, Y)\}\,.$$

To update the closest prototype Y_X, the algorithm fist selects an optimal alignment $(\boldsymbol{x}, \boldsymbol{y_x}) \in \mathcal{A}(X, Y)$. Then it applies the standard LVQ update rule

$$\boldsymbol{y_x} \leftarrow \begin{cases} \boldsymbol{y_x} + \eta(\boldsymbol{x} - \boldsymbol{y_x}) & : \quad y = c_x \\ \boldsymbol{y_x} - \eta(\boldsymbol{x} - \boldsymbol{y_x}) & : \quad y \neq c_x \end{cases},$$

where η is a monotonically decreasing learning rate following the guidelines of stochastic optimization. The updated vector representation projects to the updated graph prototype. This process continues until the procedure satisfies a termination criterion. Algorithm 1 summarizes the LGQ procedure.

3.2 LGQ2.1

In contrast to LGQ the LGQ2.1 procedure updates the two closest prototypes Y_X^1 and Y_X^2 in accordance to the current training example $(X, y) \in \mathcal{S}$. The algorithm adapts the prototypes Y_X^1 and Y_X^2 if the following conditions hold:

Algorithm 1. Learning Graph Quantization

```
Input:
    training set $\mathcal{S} = \{(X_1, y_1), \dots, (X_n, y_n)\} \subseteq \mathcal{X}_\mathcal{T} \times \mathcal{C}$
Procedure:
    1. choose initial prototypes $\mathcal{Y} = \{(Y_1, c_1), \dots, (Y_k, c_k)\} \subseteq \mathcal{X}_\mathcal{T} \times \mathcal{C}$
    2. choose vector representations $\boldsymbol{y}_1 \in Y_1, \dots, \boldsymbol{y}_k \in Y_k$
    3. repeat until termination
        3.1. randomly select a training example $(X, y) \in \mathcal{S}$
        3.2. let $Y_X = \arg\min_{Y \in \mathcal{Y}} d(X, Y)$
        3.3 choose optimal alignment $(\boldsymbol{x}, \boldsymbol{y_x}) \in \mathcal{A}(X, Y_X)$
        3.4. determine learning rate $\eta > 0$
        3.5. update according to the rule

            $$\boldsymbol{y_x} \leftarrow \begin{cases} \boldsymbol{y_x} + \eta\,(\boldsymbol{x} - \boldsymbol{y_x}) & : \text{ if } y = c_X \\ \boldsymbol{y_x} - \eta\,(\boldsymbol{x} - \boldsymbol{y_x}) & : \text{ if } y \neq c_X \end{cases}$$

Return: set $\mathcal{Y}$ of prototypes
```

1. Exactly one of both prototypes Y_X^1 and Y_X^2 has the same class label as X
2. The input graph X falls in a window around the decision border defined by

$$\frac{d\left(X, Y_X^2\right)}{d\left(X, Y_X^1\right)} > \frac{1-w}{1+w},$$

where w is the relative width of the window.

For each prototype LGQ2.1 uses the same update rule as LGQ.

4 Experiments

To assess the performance of the proposed LGQ algorithms, we conducted first experiments.

4.1 Data

We selected four data sets described in [15]. Each data set is divided into a training, validation, and a test set. The description of the data sets are mainly excerpts from [15]. Table 1 provides a summary of the main characteristics of the data sets.

Letter Graphs (high distortion level). The letter data set compiles distorted letter drawings from the Roman alphabet that consist of straight lines only (A, E, F, H, I, K, L, M, N, T, V, W, X, Y, Z). The graphs are uniformly distributed over the 15 classes (letters). The letter drawings are obtained by distorting prototype letters at high distortion level. Lines of a letter are represented by edges and ending points of lines by vertices. Each vertex is labeled with a two-dimensional

Table 1. Summary of main characteristics of the data sets

data set	#(graphs)	#(classes)	avg(nodes)	max(nodes)	avg(edges)	max(edges)
letter	750	15	4.7	8	3.1	6
grec	528	22	11.5	24	11.9	29
fingerprint	900	3	8.3	26	14.1	48

vector giving the position of its end point relative to a reference coordinate system. Edges are labeled with weight 1.

GREC Graphs. This data set [4] consists of graphs representing symbols from architectural and electronic drawings. The images occur at five different distortion levels. Depending on the distortion level, either erosion, dilation, or other morphological operations are applied. The result is thinned to obtain lines of one pixel width. Finally, graphs are extracted from the resulting denoised images by tracing the lines from end to end and detecting intersections as well as corners. Ending points, corners, intersections and circles are represented by vertices and labeled with a two-dimensional attribute giving their position. The vertices are connected by undirected edges which are labeled as line or arc. An additional attribute specifies the angle with respect to the horizontal direction or the diameter in case of arcs.

Fingerprint Graphs. This data set represents fingerprint images of the NIST-4 database [20] from four classes *arch*, *left*, *right*, and *whorl*. Fingerprint images are converted into graphs by filtering the images and extracting regions that are relevant [13]. Relevant regions are binarized and a noise removal and thinning procedure is applied. This results in a skeletonized representation of the extracted regions. Ending points and bifurcation points of the skeletonized regions are represented by vertices. Additional vertices are inserted in regular intervals between ending points and bifurcation points. Finally, undirected edges are inserted to link vertices that are directly connected through a ridge in the skeleton. Each vertex is labeled with a two-dimensional attribute giving its position. Edges are attributed with an angle denoting the orientation of the edge with respect to the horizontal direction.

4.2 Experimental Setup

Setting of LGQ. We initialized the prototypes in a class-wise manner as follows: For every class, we applied the k-means algorithm for graphs proposed by [10]. To set the number k and initialize the clusters, we partitioned the graphs according to their number of vertices. Each cell of the partition forms a cluster, if it contains at least m graphs. The number m was optimized with respect to the validation set. After applying k-means, we used the resulting cluster centers as initial prototypes of the LGQ algorithms.

We terminated both algorithms after $\max_t = 100$ cycles through the training set. The learning rate was annealed according to $\eta_t = 0.01 \cdot (1 - t/\max_t)$, where

$0 \leq t$ refers to the t-th cycle. For LGQ2.1 the window width was set to 0.1 for the letter and GREC data set and to 0.2 for the fingerprint data set.

Graph Distance Calculations and Optimal Alignment. For graph distance calculations and finding optimal alignments, we applied the graduated assignment algorithm [5]. This algorithm returns an approximate double-stochastic match matrix. We applied Munkres algorithm to convert the match matrix to a permutation sub-matrix. Using the permutation sub-matrix, we aligned the first graph towards the second.

Protocol. Both LGQ algorithms have been applied to the training set of each data set 10 times. To assess the generalization performance on the test sets, we have chosen the model that best predicts the class labels on the respective validation set. We compared the LGQ algorithms with the kNN method [15], the similarity kernel in conjunction with the SVM (SK+SVM) and the family of Lipschitz embeddings in conjunction with SVM (LE+SVM) proposed by [16] as well as the topological embedding approach based on the signature of a graph (TESG) proposed by [17].

4.3 Results

Table 2 summarizes the results. For TESG no results on letter (high) and fingerprint have been reported. Since LE+SVM refers to a family of related methods rather than a single method, Table 2 presents the best result over all methods of the LE+SVM family for each data set. In doing so, the comparison is biased towards LE+SVM.

The first observation to be made is that LGQ2.1 performs slightly superior than LGQ. Thus, as for feature vectors, pairwise adjustments of two prototypes belonging to different classes apparently better approximates the Bayes rule whereas LGQ tends to repel prototypes from Bayes decision surfaces in the graph space. The second observation to be made is that LGQ2.1 is comparable with the family of LE+SVM methods on GREC and fingerprint. Performance of LE+SVM family is, however, clearly superior on the letter data set.

As the results indicate, classifiers that directly operate in the domain of graphs can perform comparable to methods that embed graphs into vector spaces in order to apply state-of-the-art machine learning methods. An advantage of LVQ is its simplicity and efficiency. A simple initialization heuristic is sufficient to

Table 2. Classification accuracy (in %) of LGQ and LGQ2.1

	Letter	GREC	Fingerprint
kNN	90.0	95.5	77.6
SK+SVM	79.1	94.9	41.0
LE+SVM	92.5	96.8	82.8
TE	nil	95.8	nil
LGQ	80.9	94.7	79.2
LGQ2.1	83.7	97.3	82.2

learn a relatively small number of prototypes, which, in addition, can be used to extract information about characteristic structural properties of a class.

5 Conclusion

Learning graph quantization generalizes LVQ by identifying attributed graphs with points in some Riemannian orbifold. The intrinsic metric of this orbifold turns out to be a generalized differentiable graph metric, which is widely used in a number of applications. The final step, to extend the update rule of LVQ for generalized differentiable distance function is straightforward and can be applied for distance spaces other than graphs. Despite its simplicity LGQ and LGQ2.1 performed comparable to state-of-the-art methods. The promising results suggest to extend generalized LVQ and soft LVQ to the domain of graphs.

A Generalized Differentiable Functions

Let $\mathcal{X} = \mathbb{R}^n$ be a finite-dimensional Euclidean space. A function $f : \mathcal{X} \to \mathbb{R}$ is *generalized differentiable* at $\boldsymbol{x} \in \mathcal{X}$ in the sense of Norkin [14] if there is a multi-valued map $\partial f : \mathcal{X} \to 2^{\mathcal{X}}$ in a neighborhood of $\boldsymbol{x}$ such that

1. $\partial f(\boldsymbol{x})$ is a convex and compact set;
2. $\partial f(\boldsymbol{x})$ is upper semicontinuous at $\boldsymbol{x}$, that is, if $\boldsymbol{y}_i \to \boldsymbol{x}$ and $\boldsymbol{g}_i \in \partial f(\boldsymbol{y}_i)$ for each $i \in \mathbb{N}$, then each accumulation point $\boldsymbol{g}$ of $(\boldsymbol{g}_i)$ is in $\partial f(\boldsymbol{x})$;
3. for each $\boldsymbol{y} \in \mathcal{X}$ and any $\boldsymbol{g} \in \partial f(\boldsymbol{y})$ holds $f(\boldsymbol{y}) = f(\boldsymbol{x}) + \langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{x} \rangle + o\,(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{g})$, where the remainder $o\,(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{g})$ satisfies the condition

$$\lim_{i \to \infty} \frac{|o\,(\boldsymbol{x}, \boldsymbol{y}_i, \boldsymbol{g}_i)|}{\|\boldsymbol{y}_i - \boldsymbol{x}\|} = 0$$

   for all sequences $\boldsymbol{y}_i \to \boldsymbol{y}$ and $\boldsymbol{g}_i \in \partial f\,(\boldsymbol{y}_i)$.

We call f *generalized differentiable* if it is generalized differentiable at each point $\boldsymbol{x} \in \mathcal{X}$. The set $\partial f(\boldsymbol{x})$ is the *subdifferential* of f at $\boldsymbol{x}$ and its elements are called *generalized gradients.*

Generalized differentiable functions have the following properties [14]: 1. Generalized differentiable functions are locally Lipschitz and therefore continuous and differentiable almost everywhere. 2. Continuously differentiable, convex, and concave functions are generalized differentiable. 3. Suppose that $f_1, \ldots, f_n : \mathcal{X} \to \mathbb{R}$ are generalized differentiable at $\boldsymbol{x} \in \mathcal{X}$. Then

$$f_*(\boldsymbol{x}) = \min(f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \quad \text{and} \quad f^*(\boldsymbol{x}) = \max(f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))$$

are generalized differentiable at $\boldsymbol{x} \in \mathcal{X}$. 4. Suppose that $f_1, \ldots, f_m : \mathcal{X} \to \mathbb{R}$ are generalized differentiable at $\boldsymbol{x} \in \mathcal{X}$ and $f_0 : \mathbb{R}^m \to \mathbb{R}$ is generalized differentiable at $\boldsymbol{y} = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \in \mathbb{R}^m$. Then $f(\boldsymbol{x}) = f_0(f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))$ is generalized differentiable at $\boldsymbol{x} \in \mathcal{X}$. The subdifferential of f at $\boldsymbol{x}$ is of the form

$$\partial f(\boldsymbol{x}) = \text{con}\Big\{\boldsymbol{g} \in \mathcal{X} : \boldsymbol{g} = \big[\boldsymbol{g}_1 \boldsymbol{g}_2 \ldots \boldsymbol{g}_m\big] \boldsymbol{g}_0, \boldsymbol{g}_0 \in \partial f_0(\boldsymbol{y}), \boldsymbol{g}_i \in \partial f_i(\boldsymbol{x}), 1 \leq i \leq m\Big\}.$$

where $[\boldsymbol{g}_1\boldsymbol{g}_2\ldots\boldsymbol{g}_m]$ is a $(N\times m)$-matrix. 5. Suppose that $F(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}}\left[f(\boldsymbol{x},\boldsymbol{z})\right]$, where $f(\cdot,\boldsymbol{z})$ is generalized differentiable. Then F is generalized differentiable and its subdifferential at $\boldsymbol{x}\in\mathcal{X}$ is of the form $\partial F(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}}\left[\partial f(\boldsymbol{x},\boldsymbol{z})\right]$.

References

1. Almohamad, H., Duffuaa, S.: A linear programming approach for the weighted graph matching problem. IEEE Transactions on PAMI 15(5), 522–525 (1993)
2. Caetano, T.S., Cheng, L., Le, Q.V., Smola, A.J.: Learning graph matching. In: ICCV (2007)
3. Cour, T., Srinivasan, P., Shi, J.: Balanced graph matching. In: NIPS (2006)
4. Dosch, P., Valveny, E.: Report on the second symbol recognition contest. In: Liu, W., Lladós, J. (eds.) GREC 2005. LNCS, vol. 3926, pp. 381–397. Springer, Heidelberg (2006)
5. Gold, S., Rangarajan, A.: Graduated Assignment Algorithm for Graph Matching. IEEE Transactions on PAMI 18, 377–388 (1996)
6. Gold, S., Rangarajan, A., Mjolsness, E.: Learning with preknowledge: clustering with point and graph matching distance measures. Neural Computation 8(4), 787–804 (1996)
7. Günter, S., Bunke, H.: Self-organizing map for clustering in the graph domain. Pattern Recognition Letters 23(4), 405–417 (2002)
8. Hammer, B., Strickert, M., Villmann, T.: Supervised neural gas with general similarity measure. Neural Processing Letters 21(1), 21–44 (2005)
9. Jain, B., Wysotzki, F.: Central Clustering of Attributed Graphs. Machine Learning 56, 169–207 (2004)
10. Jain, B., Obermayer, K.: Structure Spaces. Journal of Machine Learning Research 10, 2667–2714 (2009)
11. Kohonen, T.: Self-organizing maps. Springer, Heidelberg (1997)
12. Kohonen, T., Somervuo, P.: Self-organizing maps of symbol strings. Neurocomputing 21(1-3), 19–30 (1998)
13. Neuhaus, M., Bunke, H.: A graph matching based approach to fingerprint classification using directional variance. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 191–200. Springer, Heidelberg (2005)
14. Norkin, V.I.: Stochastic generalized-differentiable functions in the problem of nonconvex nonsmooth stochastic optimization. Cybernetics 22(6), 804–809 (1986)
15. Riesen, K., Bunke, H.: IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 287–297. Springer, Heidelberg (2008)
16. Riesen, K., Bunke, H.: Graph Classification by Means of Lipschitz Embedding. IEEE Transactions on Systems, Man, and Cybernetics 39(6), 1472–1483 (2009)
17. Sidere, N., Heroux, P., Ramel, J.-Y.: Vector Representation of Graphs: Application to the Classification of Symbols and Letters. In: Conference Proceedings on Document Analysis and Recognition, pp. 681–685 (2009)
18. Sumervuo, P., Kohonen, T.: Self-organizing maps and learning vector quantization for feature sequences. Neural Processing Letters 10(2), 151–159 (1999)
19. Umeyama, S.: An eigendecomposition approach to weighted graph matching problems. IEEE Transactions on PAMI 10(5), 695–703 (1988)
20. Watson, C., Wilson, C.: NIST Special Database 4, Fingerprint Database. National Institute of Standards and Technology (1992)
21. Van Wyk, M., Durrani, M., Van Wyk, B.: A RKHS interpolator-based graph matching algorithm. IEEE Transactions on PAMI 24(7), 988–995 (2002)

High-Dimensional Spectral Feature Selection for 3D Object Recognition Based on Reeb Graphs

Boyan Bonev[1], Francisco Escolano[1], Daniela Giorgi[2], and Silvia Biasotti[2]

[1] University of Alicante, Spain
{boyan,sco}@dccia.ua.es
[2] IMATI CNR, Genova, Italy
{daniela,silvia}@ge.imati.cnr.it

Abstract. In this work we evaluate purely structural graph measures for 3D object classification. We extract spectral features from different Reeb graph representations and successfully deal with a multi-class problem. We use an information-theoretic filter for feature selection. We show experimentally that a small change in the order of selection has a significant impact on the classification performance and we study the impact of the precision of the selection criterion. A detailed analysis of the feature participation during the selection process helps us to draw conclusions about which spectral features are most important for the classification problem.

1 Introduction

Although feature selection (FS) plays a fundamental role in pattern classification [1], there are few studies about this topic in structured patterns, mainly when graphs are not attributed (pure structure). One exception is the work of Luo et al. [2] where different spectral features are investigated, but for embedding purposes. Regarding application areas, graph-based descriptors have been used for 3D object retrieval and classification. In this paper we study Reeb graphs [3] obtained from different functions. What is the role of each function? What is the role of each spectral feature, beyond the ones studied so far? Answering these questions, through an information-theoretic [4] method, is the main contribution of this paper. Not less important is the successful multi-class classification of unattributed graphs, using only structural information.

2 Reeb Graphs

Given a surface $\mathcal{S}$ and a real function $f : \mathcal{S} \rightarrow \mathbb{R}$, the *Reeb graph* (RG) [5] represents the topology of $\mathcal{S}$ through a graph structure whose nodes correspond to the critical points of f. When f is differentiable, the critical points are located in correspondence of topological changes of S, such as birth, join, split and death of connected components of the surface. Hence, RGs describe the *global* topological structure of $\mathcal{S}$, while also coding *local* features identified by f. RGs are becoming popular in several application domains including shape comparison, segmentation and visualisation. A detailed overview of mathematical properties, computational techniques and applications

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 119–128, 2010.

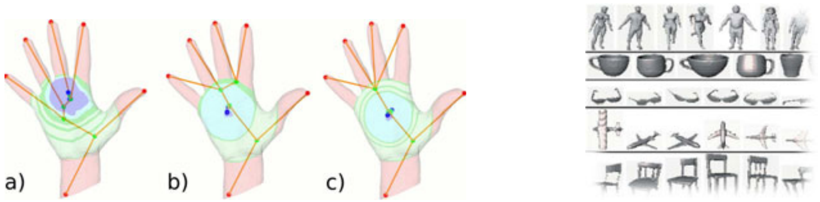


Fig. 1. Left: Extended Reeb graphs. Right: some samples of the 3D shapes database [9].

of Reeb graphs is presented in [6]. The graph representation we adopt in this paper is the *Extended Reeb Graph* (ERG) proposed in [7,3] for triangle meshes representing closed surfaces embedded in $\mathbb{R}^3$. The salient feature of ERG is the approximation of the RG by using a fixed number of level sets (63 in this paper) that divide the surface into a set of regions; critical regions, rather than critical points, are identified according to the behaviour of f along level sets; ERG nodes correspond to critical regions, while the arcs are detected by tracking the evolution of level sets.

The most interesting aspect of RGs is their parametric nature. By changing f, we have different descriptions of the same surface S that highlight different shape properties. Here we choose three alternative scalar functions f, namely the integral geodesic distance defined in [8] and the two distance functions $f(\mathbf{p}) = ||\mathbf{p} - \mathbf{b}||_2$, with $\mathbf{b}$ the center of mass and the center of the sphere circumscribing the triangle mesh respectively. Fig. 1 exemplifies our three ERG representations on a hand model, namely a) using geodesic distance [8], b) the distance from the mass center, and c) from the center of the circumscribing sphere (Fig. 1-left).

3 Features from Graph Spectra

The design of the feature extraction process is the most important part in a subsequent classification task. Concerning the characterization of a graph $G = (V, E)$, the degree distribution is a major source of statistical information. For instance, testing whether a graph is scale-free or not is posed in terms of checking whether its degree distribution follows the power law [10]. A more elaborate feature is the *subgraph node centrality* [11], which quantifies the degree of participation of a node i in structural subgraphs. It is defined in terms of the spectrum of the adjacency matrix $\mathbf{A}$, i.e. $C_S(i) = \sum_{k=1}^{n} \phi_k(i)^2 e^{\lambda_k}$, where $n = |V|$, λ_k the k-th eigenvalue of $\mathbf{A}$ and ϕ_k its corresponding eigenvector. In this regard, ϕ_n (the eigenvector corresponding to the largest eigenvalue) is the so called *Perron-Frobenius eigenvector*. The components of the latter vector denote the degree of importance of each node in a connected component and they are closely related to subgraph centrality [11]. Furthermore, the magnitudes $|\phi_k|$ of the (leading) eigenvalues of $\mathbf{A}$ have been been experimentally validated for graph embedding [2]. Besides the study of the adjacency matrix, it is also interesting to exploit the *spectrum of the Laplacian* $\mathbf{L} = \mathbf{D} - \mathbf{A}$ or the *spectrum of the normalized Laplacian* $\mathcal{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ is the diagonal degree matrix. These spectra encode significant structural information. For instance, $\lambda_2 \leq n/(n-1)$ $, n \geq 2$;

in addition, the multiplicity of the trivial eigenvalue yields the number of connected components. in the case of $\mathcal{L}$ we have $\lambda_k \leq 2,\ 2 \leq k \leq n$, and the Laplacian spectrum plays a fundamental role in the design of regularization graph kernels. Such kernels encode a family of dissimilarity measures between the nodes of the graph. Regarding the eigenvectors of the Laplacian, the *Friedler vector*, that is, the eigenvector corresponding to the first non-trivial eigenvalue, ϕ_2 in connected graphs, encodes the connectivity structure of the graph (actually its analysis is the core of graph-cuts methods) and it is related to the Cheeger constant. In addition, both the eigenvectors and eigenvalues of the Laplacian are key to defining a metric between the nodes of the graph, namely the *commute time*, $CT(i,j)$. It is the average time taken by a random walk starting at i to reach j and then returning. If we use the un-normalized Laplacian, we have that $CT(i,j) = vol \sum_{k=2}^{n}(1/\lambda_k)(\phi_k(i) - \phi_k(j))^2$, where vol is the volume of the graph, that is, the trace of $\mathbf{D}$. In the normalized case $CT(i,j) = vol \sum_{k=2}^{n}(1/\lambda_k)(\phi_k(i)/\sqrt{d_i} - \phi_k(j)/\sqrt{d_j})^2$, where d_i and d_j are the degrees of i and j respectively. Since the commute time is a metric, and because of its utility for graph embedding [12], the path-length structure of the graph is partially encoded. Finally, considering diffusion kernels on graphs, which belong to the family of regularization kernels, the analysis of the diffusion process itself yields a valuable source of information concerning the structure of the graph. A recent characterization of the diffusion process is the *the flow complexity trace* [13], a fast version of polytopal complexity [14]. The complexity trace encodes the amount of heat flowing through the edges of G for a set of inverse temperatures β: from $\beta = 0$ (no flow) to $\beta \rightarrow \infty$ (flow equal to $2|E|$) there is a phase-transition point. More precisely, the instantaneous flow for a given β is $F(G;\beta) = \sum_{i=1}^{n}\sum_{j\neq i}^{n} A_{ij}(\sum_{k=1}^{n}\phi_k(i)\phi_k(j)e^{-\beta\lambda_k})$ and the trace element for this inverse temperature is the instantaneous complexity $C(G;\beta) = \log_2(1 + F(G;\beta)) - \log_2(n)$ where the final term is for the purpose of size normalization.

4 Feature Selection

4.1 Mutual Information Criterion

In *filter feature selection* methods, the criterion for selecting or discarding features does not depend on any classifier. We estimate the mutual information (MI) between the features set and the class label, provided that we tackle a supervised classification problem: $I(\boldsymbol{S};\boldsymbol{C}) = H(\boldsymbol{S}) - H(\boldsymbol{S}|\boldsymbol{C})$. Here $\boldsymbol{S}$ is a matrix of size $m \times n$ and $\boldsymbol{C}$ of size $m \times 1$ where m is the number of samples and n the number of features of the feature subset. Traditionally the MI has been evaluated between a single feature and the class label. Here we calculate the MI using the entire set of features to select. This is an important advantage in FS, as the interactions between features are also taken into account [1]. The entropies $H(\cdot)$ of a set with a large n number of features can be efficiently estimated using the k-NN-based method developed by Leonenko [15]. Thus, we take the data set with all its features and determine which feature to discard in order to produce the smallest decrease of $I(\boldsymbol{S_{n-1}};\boldsymbol{C})$. We then repeat the process for the features of the remaining feature set, until only one feature is left. A similar information-theoretic selection approach is described in detail in [16]. They use minimal spanning trees for entropy estimation, while in this work we use the method of Leonenko which is simpler and allows us to vary the precision of the estimation by using the Approximate Nearest Neighbours algorithm [17].

4.2 Entropy Estimation

A simple way to understand the k-NN entropy estimation proposed by Leonenko [15] is to look at the Shannon entropy formula $H(X) = -\int f(x)\log f(x)dx$, as an average of $\log f(x)$, being $f(x)$ an existing pdf. The estimation of $\widehat{\log f(x)}$ would allow the estimation of $\hat{H}(X) = -N^{-1}\sum_{i=1}^{N}\widehat{\log f(x)}$. For this purpose the probability distribution $P_k(\epsilon)$ of the distance between a sample x_i and its k-NN is considered. If a ball of diameter ϵ is centered at x_i and there is a point within distance $\epsilon/2$, then there are $k-1$ other points closer to x_i and $N-k-1$ points farther from it. The probability of this to happen is $P_k(\epsilon)d\epsilon = k\binom{N-1}{k}\frac{dp_i(\epsilon)}{d\epsilon}p_i^{k-1}(1-p_i)^{N-k-1}$ being p_i the mass of the ϵ-ball and $p_i(\epsilon) = \int_{||\xi - x_i|| < \epsilon/2} f(\xi)d\xi$.

The expectation of of $\log p_i(\epsilon)$ is $E(\log p_i) = \int_0^\infty P_k(\epsilon)\log p_i(\epsilon)d\epsilon$ that is $= k\binom{N-1}{k}\int_0^1 p^{k-1}(1-p)^{N-k-1}\log p \cdot dp = \psi(k) - \psi(N)$, where $\psi(\cdot)$ is the well-known digamma function. If assumed that $f(x)$ is constant in the entire ϵ-ball, then the approximation $p_i(\epsilon) \approx \frac{V_d}{2^d}\epsilon^d\mu(x_i)$ can be formulated. Here d is the dimension and V_d is the volume of the unit ball $\mathcal{B}(0,1)$, defined as $V_d = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}$. From the previous approximation and using the expectation of $\log p_i(\epsilon)$, we have the approximation $\log f(\epsilon) \approx \psi(k) - \psi(N) - dE(\log\epsilon) - \log\frac{V_d}{2^d}$, and finally,

$$\hat{H}(X) = -\psi(k) + \psi(N) + \log\frac{V_d}{2^d} + \frac{d}{N}\sum_{i=1}^{N}\log\epsilon_i \tag{1}$$

is the estimation of $H(X)$, where $\epsilon_i = 2||x_i - x_j||$ is twice the distance between the sample x_i and its k-NN x_j. It is suggested that the error for Gaussian and uniform distributions is $\sim k/N$ or $\sim k/N\log(N/k)$.

4.3 Experimental Setup

In this work each sample is originally a 3D object represented by a triangle mesh. From each 3D object, three types of graphs (Sec. 2) are extracted (labeled in the figures as a) *Sphere*, b) *Baricenter* and c) *Geodesic*). Only the structural graph information is used for classification. For each graph, 9 different measures (listed in the area plots in Fig. 4) are calculated, as described in Sec. 3. They are transformed into histograms after normalizing them by the volume of the graph. Commute time is normalized twice, 1) linearly and 2) quadratically. Histograms are used in order to characterize the graph without dependence on the number and order of nodes. Only the complexity flow curve is not histogrammed, for the sake of order preservation. Since there is no optimal way to select the number of bins, we perform several different binnings on each measure (2, 4, 6 and 8 bins). All histograms form a bag of features, of size $9 \cdot 3 \cdot 20 = 540$ features (see Fig. 2). We let the FS process decide which binning from which measure and from which graph to discard.

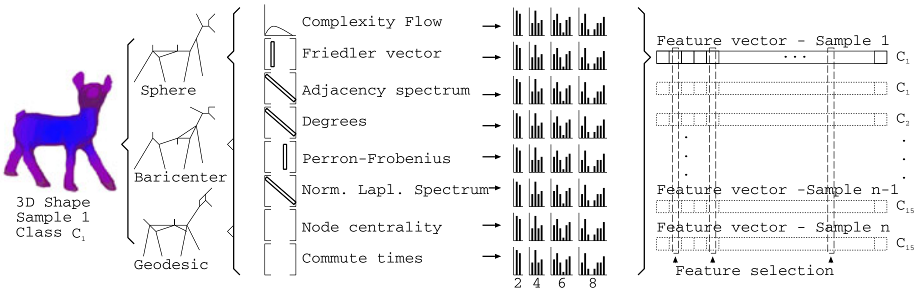


Fig. 2. The process of extracting from the 3D object the three graph representations, unattributed graph features, histogram bins, and finally selecting the features

5 Results and Discussion

The experiments are performed on the pre-classified 3D shapes database [9]. It consists of 15 classes × 20 objects. Each one of the 300 samples is characterized by 540 features, and has a class label $l \in \{$*human, cup, glasses, airplane, chair, octopus, table, hand, fish, bird, spring, armadillo, buste, mechanic, four-leg*$\}$; see Fig. 1-right.

5.1 Classification Error

The errors are measured by 10-fold cross validation (10-fold CV). In Fig. 3 we show how MI is maximized as the number of selected features grows, and its relation to the decrease in error. The figure shows how a high number of features degrades the classification performance. For the 15-class problem, the optimal error $(23, 3)$ is achieved with a set of 222 features. This error is lower for 10 classes $(15, 5\%)$, 5 classes (6%) and 2 classes problems (0%). These results depend on the classifier used for measuring the error. However the MI curve, as well as the selected features, do not depend on the classifier, as it is a purely information-theoretic measure.

5.2 Features Analysis

Several different unattributed graph measures are used in this work. We aim to determine which measures are most important and in which combinations. In Fig. 4-left we show the evolution of the proportion of selected features. The coloured areas in the plot represent how much a feature is used with respect to the remaining ones (the height on the Y axis is arbitrary). For the 15-class experiment, in the feature sets smaller than 100 features, the most important is the Friedler vector, *in combination* with the remaining features. Commute time is also an important feature. Some features that are not relevant are the node centrality and the complexity flow. Turning our attention to the graphs type, all three appear relevant. In Fig. 4-right we show the proportion of features selected for the 222-feature set, which yielded the lowest error in our experiments. (The dashed vertical line in Fig. 4-left also shows the 222-feature set) In the plot representing the selected binnings we can see that the four different binnings of the features do have importance for graph characterization.

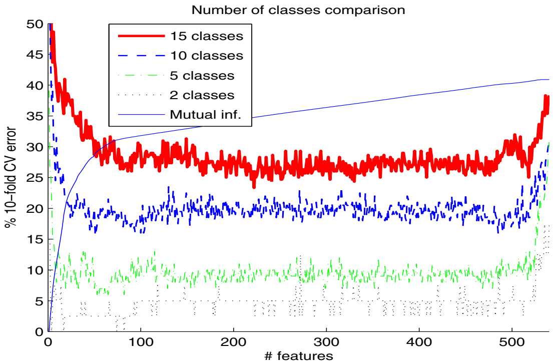


Fig. 3. Classification errors

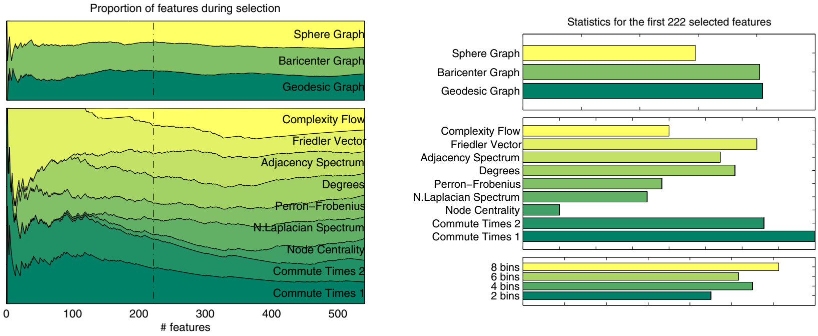


Fig. 4. Feature selection on the 15-class experiment (left) and the feature statistics for the best-error feature set (right)

These conclusions concerning the relevance of each feature cannot be drawn without performing some additional experiments with different groups of graph classes. For this purpose in Fig. 5 we present four different 3-class experiments. The classes share some structural similarities, for example the 3 classes of the first experiment have a head and limbs. Although in each experiment the minimum error is achieved with very different numbers of features, the participation of each feature is highly consistent with the 15-class experiment. The Friedler vector is always the most important for feature sets smaller than 100. On the other hand, the commute time measure is not important for feature sets smaller than 20, but then it becomes as important as the Friedler vector. The main difference among experiments (Fig. 5) is that node centrality seems to be more important for discerning among elongated sharp objects. Although all three graph types are relevant, the *sphere graph* performs best for blob-shaped objects.

5.3 The Impact of Feature Selection

In Fig. 3 it is obvious that, if the number of features used for classification is too low, the error is higher, and on the other hand if the number of features is too high, the

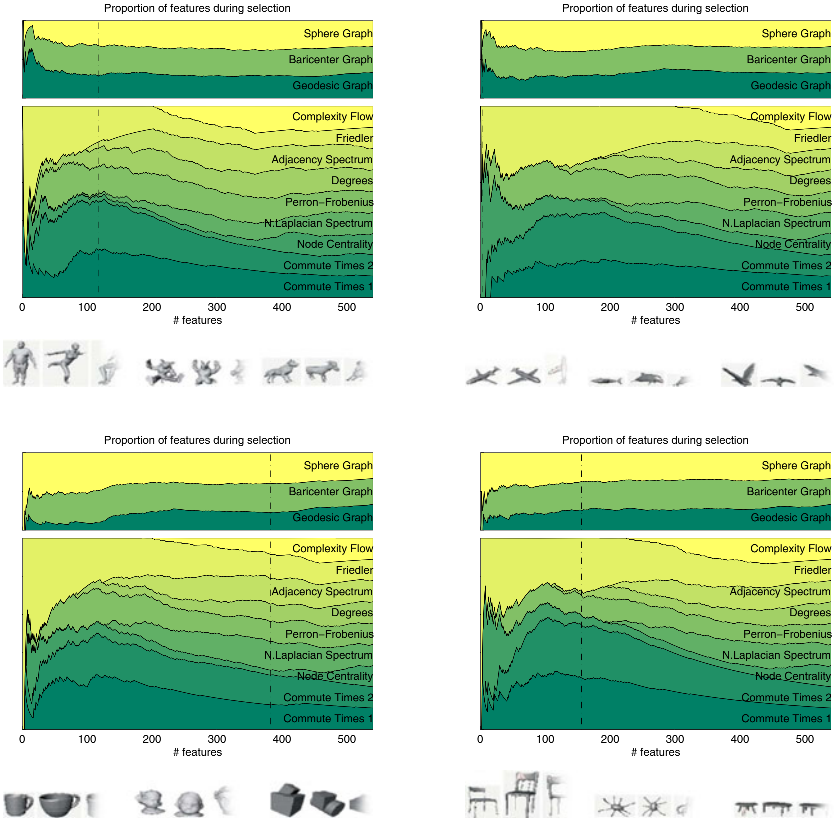


Fig. 5. Feature Selection on 3-class experiments: Human/Armadillo/Four-legged, Aircraft/Fish/Bird, Cup/Bust/Mechanic, Chair/Octopus/Table

error could also rise. However this depends on the order of features are added. In this work we use mutual information as the evaluation criterion because it is related to the minimization of the Bayesian error. What would happen if a worse criterion is used? To what extent the precision of the mutual information estimation is important? What is its impact on the final classification error?

Following we present some experiments which answer these questions. All of them refer to the 15-classes experiment. In order to vary the precision of the mutual information criterion we change the error bound ϵ of the ANN algorithm which is used for entropy estimation. ANN builds a kd-tree structure, whose cells are visited in increasing order of distance from the query point. A stop condition of the search algorithm occurs when the distance is closer than an error bound ϵ. This premature stop can save computational time, as shown in Fig. 6-right. It also causes a decrease in the precision of the k-NN computation. Thus, the entropy estimation, and so, the mutual information estimations, are degraded. To what extent? This is shown in Fig. 6-left. It is interesting to see that the error bound $\epsilon = 0$ yields significantly better feature selection results, in terms of 10-fold Cross Validation error. Also, the increment of the error bound is not linear with respect to the increment of the 10-fold CV error.

The differences in the classification performance are due to small differences in the feature sets. For example, the difference among the feature sets yielded by $\epsilon = 0$ and $\epsilon = 1$ are significant (see Fig. 7,-top-left). Then, before the error bound ϵ arrives the 0.5 value, the feature sets remain very similar. Other significant changes in the feature sets are plotted in Fig. 7. Each one of the figures compares two different feature selection processes, as a consequence of different ϵ values. The first process is represented as a coloured area plot, and the second one is represented with black solid lines.

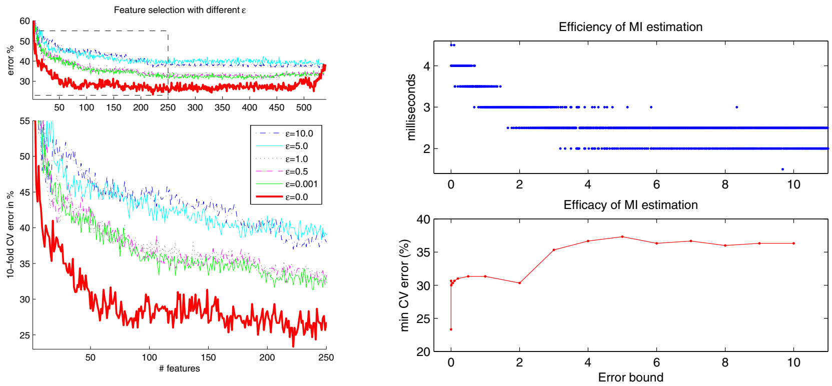


Fig. 6. Left: the 10-fold CV errors yielded by several feature selection runs, with different ANN error bound values (ϵ). Right-top: the milliseconds it takes (on a 1.6GHz Intel Centrino processor and DDR2 RAM) to evaluate the mutual information between the 300 samples with 540 features, and the 15 class labels. Right-bottom: the minimal errors achieved in the 15-class feature selection, for different Error bound (ϵ) values.

The most important differences are observed in the early stage of feature selection (before the first 200 features are selected). After that, the proportion among the different features selected converges, because there are no more features left for selecting. It is the early stage of the selection process which strongly conditions the maximum error which could be achieved, as shown in Fig. 6-left: a good run ($\epsilon = 0$) yields an error plot which decreases to $23,3\%$, and after that increases to $37,33\%$. A run which yields poor results is the case of $\epsilon = 0.5$, for instance. In this case the error decreases progressively until achieving $37,33\%$, but none of the feature subsets produces a lower error.

It is also worth observing that the node centrality and the Friedler vector features are always important in the beginning of the process, disregarding the precision of the feature selection criterion. Shortly after the beginning, commute times start to play an important role. Regarding the Reeb graph types, most of the features are selected from the "sphere graph" type.

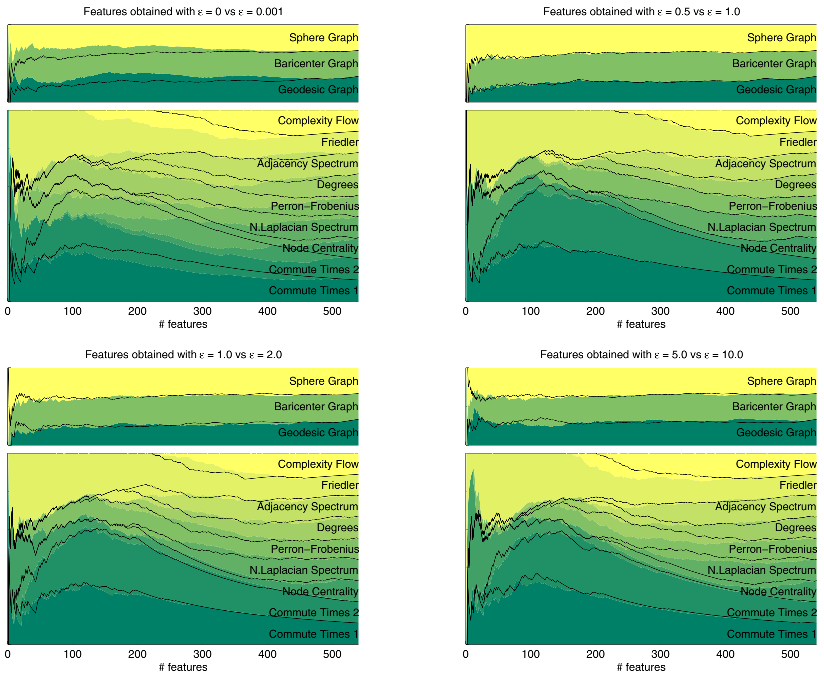


Fig. 7. Four feature selection comparisons of different pairs of ϵ values. The first feature selection process is represented as a coloured area plot, while the second one is plotted with black solid lines.

6 Conclusions

The contributions of this work to graph classification are twofold. Firstly, it demonstrates the feasibility of multi-class classification based on purely structural spectral features. Secondly, an information-theoretic feature analysis suggests that similar features are selected for very different sets of objects. Moreover, the feature selection experiments show that even if the precision of the selection criterion is degraded, the most important features are still the same.

On the other hand this paper demonstrates some important effects of feature selection. In the first place we prove that the precision of the mutual information estimation has a great impact on the final classification performance. The same experiments show how very small changes in the order of the selected features can also affect the classification result. Working with the maximum precision available is key to minimizing the classification error.

As future work we consider using attributed and directed graphs for improving classification accuracy. We also find it necessary to use a wider range of graph features, as well as other kinds of graph extraction methods.

References

1. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
2. Luo, B., Wilson, R., Hancock, E.: Spectral embedding of graphs. Pattern Recognition 36(10), 2213–2223 (2003)
3. Biasotti, S.: Topological coding of surfaces with boundary using Reeb graphs. Computer Graphics and Geometry 7(1), 31–45 (2005)
4. Escolano, F., Suau, P., Bonev, B.: Information Theory in Computer Vision and Pattern Recognition. Springer, New York (2009)
5. Reeb, G.: Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. Comptes Rendus 222, 847–849 (1946)
6. Biasotti, S., Giorgi, D., Spagnuolo, M., Falcidieno, B.: Reeb graphs for shape analysis and applications. Theoretical Computer Science 392(1–3), 5–22 (2008), doi:10.1016/j.tcs.2007.10.018.
7. Biasotti, S.: Computational Topology Methods for Shape Modelling Applications. PhD thesis, Universitá degli Studi di Genova (May 2004)
8. Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.L.: Topology matching for fully automatic similarity estimation of 3D shapes. In: SIGGRAPH 2001, Los Angeles, CA, pp. 203–212 (2001)
9. Attene, M., Biasotti, S.: Shape retrieval contest 2008: Stability of watertight models. In: SMI 2008, pp. 219–220 (2008)
10. Barabási, A.L., Bonabeau, E.: Scale-free networks. Scientific American 288, 50–59 (2003)
11. Estrada, E., Rodriguez, J.A.: Subgraph centrality in complex networks. Physical Review E 71(5) (2005)
12. Qiu, H., Hancock, E.R.: Clustering and embedding using commute times. IEEE Transactions on PAMI 29(11), 1873–1890 (2007)
13. Escolano, F., Giorgi, D., Hancock, E.R., Lozano, M.A., Falcidieno, B.: Flow complexity: Fast polytopal graph complexity and 3d object clustering. In: GbRPR, pp. 253–262 (2009)
14. Escolano, F., Hancock, E.R., Lozano, M.A.: Birkhoff polytopes, heat kernels and graph complexity. In: ICPR, pp. 1–5 (2008)
15. Leonenko, N., Pronzato, L., Savani, V.: A class of rényi information estimators for multidimensional densities. The Annals of Statistics 36(5), 2153–2182 (2008)
16. Bonev, B., Escolano, F., Cazorla, M.: Feature selection, mutual information, and the classification of high-dimensional patterns. Pattern Analysis and Applications (February 2008)
17. Mount, D., Arya, S.: Ann: A library for approximate nearest neighbor searching (1997)

Dissimilarity-Based Multiple Instance Learning

Lauge Sørensen[1], Marco Loog[1,2], David M.J. Tax[2], Wan-Jui Lee[2],
Marleen de Bruijne[1,3], and Robert P.W. Duin[2]

[1] The Image Group, Department of Computer Science,
University of Copenhagen, Denmark
lauges@diku.dk
[2] Pattern Recognition Laboratory, Delft University of Technology, The Netherlands
[3] Biomedical Imaging Group Rotterdam, Departments of Radiology & Medical Informatics, Erasmus MC, Rotterdam, The Netherlands

Abstract. In this paper, we propose to solve multiple instance learning problems using a dissimilarity representation of the objects. Once the dissimilarity space has been constructed, the problem is turned into a standard supervised learning problem that can be solved with a general purpose supervised classifier. This approach is less restrictive than kernel-based approaches and therefore allows for the usage of a wider range of proximity measures. Two conceptually different types of dissimilarity measures are considered: one based on point set distance measures and one based on the earth movers distance between distributions of within- and between set point distances, thereby taking relations within and between sets into account. Experiments on five publicly available data sets show competitive performance in terms of classification accuracy compared to previously published results.

Keywords: dissimilarity representation, multiple instance learning, bag dissimilarity measure.

1 Introduction

In multiple instance learning (MIL), complex objects are represented by sets of "sub-objects" where only the sets have an associated label, not the sub-objects. Following MIL terminology, the sets are termed bags and the sub-objects are termed instances. This kind of problem might, e.g., arise in medical image classification where a subject is known to suffer from a certain disease, but it is not clear exactly which regions in the associated medical image that contain the corresponding pathology. In this case, local image patches are the instances, the whole image is the bag, and the label of the bag is either ill or healthy.

The traditional approach to solving MIL problems involves explicit learning of a decision boundary in instance space that separates the instances capturing the concept from the remaining instances [1,2]. A bag is then classified based on whether it contains an instance falling in this area. An alternative instance space approach involves labeling all instances with the same label as the bag they belong to. The problem is then treated as a standard supervised learning

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 129–138, 2010.

problem where all instances are classified in instance space, ultimately disregarding the multiple instance aspect of the original problem, and a bag is classified by combining the individual instance classifications in that bag [3].

The above mentioned approaches treat instances in the same bag independently in the learning step thereby disregarding potentially useful information. In some MIL problems, instances from the same bag collectively constitute that bag and should as such all contribute to the classification of that bag. Several authors have looked into using this information by applying learning at bag level with kernel-based methods. To name a few: Andrews *et al.* reformulated a support vector machine (SVM) optimization problem to operate directly on MIL problems at bag level [4]. Gärtner *et al.*, Tao *et al.*, and Zhou *et al.* designed specialized kernels for MIL problems and used standard SVMs with these kernels [5,6,7]. Chen *et al.* represented bags in an n-dimensional space where each dimension was the similarity between one of the n instances in the training set and the closest instance in a bag. Then a 1-norm SVM was used to simultaneously select the relevant features, or instances, and train a bag classifier [8].

In this paper, we propose to use the dissimilarity representation approach to learning [9] for solving MIL problems at the bag level. Once the bag dissimilarity space has been constructed, the problem is turned into a standard supervised learning problem that can be solved with a general purpose supervised classifier. This is a proximity-based approach as are kernel-based methods, however, the dissimilarity representation approach does not require Mercer kernels as do kernel-based methods. A broader range of proximity measures, such as well known measures in pattern recognition like the Hausdorff distance and the single linkage distance, can therefore be used for solving MIL problems with this approach. We further propose, not only to consider all instances collectively in bag classification, but also to consider the relations among the instances within and between bags. This is similar in spirit to [7] where graphs capturing instance relations were constructed and used in a SVM with a graph kernel [5]. A novel non-Mercer bag dissimilarity measure that is based on the earth movers distance (EMD) between instance distance distributions is proposed for this purpose. Compared to the graph kernel approach used in [7], the proposed bag dissimilarity measure is less rigid since distributions of instance distances are considered instead.

Dissimilarity-based learning has previously been applied in MIL. Wang and Zucker applied the k nearest neighbor (kNN) classifier to MIL problems by using the Hausdorff distance between the instances in two bags as the distance between these bags [10]. They showed that this was not sufficient to get good performance on the classical MIL data sets MUSK1 and MUSK2 [1], due to noise in the presence of negative instances in the positive bags, and suggested two adaptations of kNN instead. A key observation is that kNN using Hausdorff distance between instances is working on dissimilarities between bags, and one way of arriving at a more global and robust decision rule when dissimilarities between objects are available is via a dissimilarity representation [9]. Building a global classifier like the Fisher linear discriminant classifier (Fisher) on such a representation leads to a global decision rule that uses a weighted combination of the dissimilarities to all training set objects in classification. This means better utilization of the available training data, with possibly increased accuracy and less sensitivity to noise.

The rest of the paper is organized as follows: Sections 2 and 3 briefly describe the MIL problem and the dissimilarity representation approach to learning. Section 4 presents two conceptually different types of dissimilarity measures between bags of instances. The first type is points set distance measures and the second type is based on EMD between distributions of instance distances within- and between bags. The proposed approach is evaluated by training and testing traditional supervised classifiers on dissimilarity representations of five publicly available MIL data sets. This is reported in Section 5. Finally, Section 6 provides a discussion and conclusions.

2 Multiple Instance Learning in Short

In MIL [1], an object x_i is represented by a set, or bag, $B_i = \{\mathbf{x}_{ij}\}_{n_i}$ of n_i instances $\mathbf{x}_{ij}$, and a label $Y_i = \{+1, -1\}$ is associated with the entire bag. There are no labels y_{ij} associated directly with the instances, only indirectly via the label of the bag. This is different from standard supervised learning where objects are represented by a single instance, i.e., $B_i = \mathbf{x}_i$ and all instances therefore are directly labeled. The bag labels are interpreted in the following way in the original MIL formulation [1]: if $Y_i = -1$, then $\forall \mathbf{x}_{ij} \in B_i : y_{ij} = -1$. If $Y_i = +1$, then $\exists \mathbf{x}_{ij} \in B_i : y_{ij} = +1$. In other words, if a bag is labeled as positive, then at least one instance in that bag is a positive example of the underlying concept. This formulation can be relaxed to cope with a large and noisy set of instances by requiring that a positive bag contains a number or fraction of positive instances instead. In this work, we only consider two-class problems, but MIL can also be generalized to multi-class problems.

3 Dissimilarity Representations in Short

Objects x are traditionally represented by feature vectors in a feature vector space, and classifiers are built in this space. Alternatively, one can represent the objects by their pair-wise dissimilarities $d(x_i, x_j)$ and build classifiers on the obtained dissimilarity representation [9]. From the matrix of pair-wise object dissimilarities $D = [d(x_i, x_j)]_{n \times n}$ computed from a set of objects $\{x_1, \ldots, x_n\}$, there are different ways of arriving at a feature vector space where traditional vector space methods can be applied. In this work, we consider the dissimilarity space approach [9].

Given a training set $T = \{x_1, \ldots, x_n\}$, a subset $R = \{p_1, \ldots, p_k\} \subseteq T$ called the representation set containing prototype objects p_i is selected. An object x is represented with respect to R by the vector $D(x, R) = [d(x, p_1), \ldots, d(x, p_k)]$ of dissimilarities computed between x and the prototypes in R. This k-dimensional vector space based on R is called a dissimilarity space, and it is in this space that we propose to solve MIL problems at the bag level. In this work, we apply learning in the full dissimilarity space, i.e., $R = T$.

4 Bag Dissimilarity Space

The idea we propose is to map the bags into a dissimilarity space $D(\cdot, R = \{B_i\}_k)$. Here the bags are represented as single objects, positioned with respect to their dissimilarities to the prototype bags in R. In this space, the MIL problem can be considered as a standard supervised classification problem where each object $x_i = B_i$ has label Y_i and general purpose supervised classifiers can be directly applied. The separation of the bags in the obtained dissimilarity space is very much dependent on the choice of bag dissimilarity measure $d(B_i, B_j)$. In the following, we present two conceptually different types of dissimilarity measures for bags of instances.

4.1 Point Set Distance Measures

The instances $\mathbf{x}$ reside in a common space and bags B can therefore be thought of as sets of objects in this space. In the case of vectorial instances, these objects are points in a vector space. This leads to the idea of computing dissimilarities between bags using point set distance measures. In this work, we experiment with the minimum distance

$$d_{min}(B_i, B_j) = \min_{p,q} ||\mathbf{x}_{ip} - \mathbf{x}_{jq}||_2 \tag{1}$$

and the Hausdorff distance

$$d_H(B_i, B_j) = \max\{d_{dir}(B_i, B_j), d_{dir}(B_j, B_i)\} \tag{2}$$

which is based on the directed distance $d_{dir}(B_i, B_j) = \max_p \min_q ||\mathbf{x}_{ip} - \mathbf{x}_{jq}||_2$. These point set distance measures were also used in a modified kNN classifier in [10].

Both point set distance measures (1) and (2) use the distance between two single instances in the end. These measures may therefore be sensitive to noisy instances, and they are in general insensitive to the number of positive instances in a positive bag. This may not be desirable when constructing a bag dissimilarity representation, and taking more information about the instances in a bag into account in the bag dissimilarity measure may lead to a better representation of the bags.

4.2 Measures Based on between- and within Bag Instance Distances

Zhou *et al.* conjectured that instances in a bag are rarely independently and identically distributed and that relations among the instances may convey important information when applying learning at bag level [7]. In a similar spirit, we propose two bag dissimilarity measures that take relations among instances

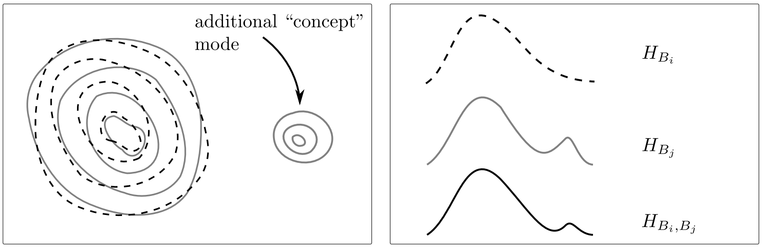


Fig. 1. Left: Illustration of two similar bag class distributions where one of the distributions, typically the positive bag distribution, has an extra mode corresponding to the positive instances. **Right**: Distributions of instance distances, from top to bottom: within bag instance distances in a bag from the class with no additional mode, typically the negative class; within bag instance distances in a bag from the class with an additional mode, typically the positive class. Notice the extra “bump” in the distribution; instance distances between two bags, one from each class.

into account, or more precisely, the distribution of instance distances within a bag and between bags. It is assumed that the instances in the two bag classes follow distributions in the common instance space that are very similar, with the slight difference that one distribution contains additional modes capturing the concept(s). This situation is illustrated, for one additional mode, to the left in Figure 1. This could, e.g., be the situation in a MIL problem in medical image classification where the positive medical images contain lesions surrounded by healthy tissue whereas the negative images only contain healthy tissue. The additional mode in one of the bag class distributions gives rise to an extra “bump” in the distribution of instance distances within bags from that class, compared to bags from the other class, as illustrated to the right in Figure 1. Further, the bump can also be seen in the histogram of instance distances computed between bags from the two classes.

We propose to use the within bag instance distance histograms H_{B_i} and H_{B_j}, computed from bag B_i and B_j, respectively, and the between bag instance distance histogram H_{B_i,B_j}, computed between bag B_i and B_j. The bag dissimilarity measure is then computed as the pair-wise histogram dissimilarity $d_{i,ij} = d(H_{B_i}, H_{B_i,B_j})$. $d_{i,ij}$ can be seen as the directed dissimilarity from B_i to B_j. The maximum and the mean of the directed dissimilarities from each of the two bags are proposed as two symmetric dissimilarity measures for bags

$$d_{BWmax}(B_i, B_j) = \max\{d_{i,ij}, d_{j,ij}\} \tag{3}$$

and

$$d_{BWmean}(B_i, B_j) = \frac{1}{2}(d_{i,ij} + d_{j,ij}). \tag{4}$$

The histogram dissimilarities are computed using EMD [11] between the normalized empirical distributions. For one-dimensional histograms $H = [h_1, \ldots, h_n]^T$ and $K = [k_1, \ldots, k_n]^T$ of equal number of bins n and equal mass, EMD can be computed as the L1-norm between the cumulative histograms of H and K: $d_{EMD}(H,K) = \sum_{i=1}^{n} | \sum_{j \leq i} h_j - \sum_{j \leq i} k_j |$.

4.3 A Second Dissimilarity Space

Initial experiments showed that linear classifiers performed poorly when built on the obtained bag dissimilarity representations whereas the nearest neighbor classifier (1NN) performed quite well. This indicates that the bags are separated in the obtained dissimilarity representations, but that the decision boundaries between the positive bags and the negative bags are complicated and non-linear, and/or that the class distributions are multi-modal in these new representations. An extra preprocessing step is therefore done before applying linear classifiers. From $D(\cdot, X)$ computed on the full data set X, a new dissimilarity representation $D2$ is constructed such that $D2(x_i, x_j) = ||D(x_i, X) - D(x_j, X)||_2, \forall_{x_i, x_j} \in X$. The linear classifiers are built on this representation. This is a transductive learning approach since all objects are used to construct the representation D2. It is, however, important to note that the labels of the objects are not considered in this construction. Tao *et al.* also used transductive learning to solve MIL problems [6].

5 Experiments and Results

The proposed approach is evaluated on the two standard data sets in MIL, namely MUSK1 and MUSK2 originally used in [1], and on three recently published image retrieval data sets [4].

5.1 MUSK1 and MUSK2

These are the standard MIL data sets, and they consist of descriptions of aromatic molecules that have been labeled according to whether they smell “musky” or not. A bag represents a molecule, and the instances in a bag are low energy shapes of the molecule described by 166-dimensional feature vectors. The MUSK1 data set comprises 47 positive bags and 45 negative bags, and each bag is represented by 2 to 40 instances. The MUSK2 data set comprises 39 positive bags and 63 negative bags, and each bag is represented by 1 to 1044 instances. The data was obtained from the UCI Machine Learning Repository [12], and we refer to this source as well as to [1] for further information about the data.

5.2 Image Retrieval

This data comprises three data sets that are subsets of the Corel data set. Each data set consists of 100 positive bags, or example images; elephant, fox, or tiger, and 100 negative bags, or background images, which are randomly drawn from a pool of photos of other animals. Each image is represented by 2-13 instances (apart from a single image in the tiger data set that is represented by a single instance), which are 230-dimensional feature vectors describing the color, texture and shape in subsegments of the image. The data was obtained from the homepage[1] associated with [4] and we refer to these sources for further information about the data.

5.3 Evaluation

The proposed dissimilarity representations are evaluated by training and testing three supervised classifiers on the bags in the given dissimilarity space. These classifiers are: 1NN; SVM with a linear kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$ where $\mathbf{x}_i = D2(B_i, X)$ and trade-off parameter $C = 1$; and Fisher. For 1NN and Fisher we use the pattern recognition toolbox PRTools [13], and for SVM we use LIBSVM [14].

Table 1. Classification accuracy on the MUSK1 and MUSK2 data set, reported as leave-one-out / ten-fold cross-validation. Accuracies reported in the literature are shown in the bottom part of the table. Cases in the literature where the classification accuracy is not reported using leave-one-out or ten-fold cross-validation are marked with "-". The highest accuracy among the dissimilarity representation-based classifiers as well as the highest accuracy in general is marked in boldface in each column.

Classifier	Bag dissimilarity measure	MUSK1	MUSK2
1NN (on D)	d_{min} (1)	90.2 / **91.3**	86.9 / 84.6
	d_H (2)	88.0 / 87.9	86.1 / 82.5
	d_{BWmax} (3)	85.8 / 86.9	82.8 / 77.7
	d_{BWmean} (4)	89.1 / 91.2	85.3 / 80.7
SVM (on $D2$)	d_{min} (1)	90.0 / 90.1	92.2 / 87.5
	d_H (2)	88.0 / 88.0	91.2 / 85.5
	d_{BWmax} (3)	89.1 / 89.0	82.2 / 88.3
	d_{BWmean} (4)	**91.2** / 89.0	85.3 / 85.0
Fisher (on $D2$)	d_{min} (1)	90.1 / 90.1	**93.5** / **92.7**
	d_H (2)	88.0 / 86.9	90.3 / 88.2
	d_{BWmax} (3)	90.1 / 87.9	87.7 / 87.4
	d_{BWmean} (4)	**91.2** / 91.2	89.8 / 90.3
citation-kNN [10]		**92.4** / -	86.3 / -
iterated discrim APR [1]		- / **92.4**	- / 89.2
diverse density [2]		- / 88.9	- / 82.5
mi-SVM [4]		- / 87.4	- / 83.6
MI-SVM [4]		- / 77.9	- / 84.3
SVM polynomial minimax kernel [5]		**92.4** / -	86.3 / -
SVM MI kernel [5]		87.0 / -	92.2 / -
MILES [8]		86.3 / 87.0	87.7 / **93.1**
$k_\wedge$ *emph* transduction [6]		- / 91.2	- / 90.3
$k_{\wedge/\vee}$ *emph* transduction [6]		- / 90.2	- / 92.2
MIGraph [7]		- / 90.0	- / 90.0
miGraph [7]		- / 88.9	- / 90.3

[1] http://www.cs.columbia.edu/˜andrews/mil/datasets.html

Classification accuracies are estimated using leave-one-out and 10-fold cross-validation, since these are commonly used performance measures in the MIL literature [1,2,10,4,5,3,7]. 10-fold cross-validation is sometimes performed once and sometimes the average of a repeated number of 10-fold cross-validation procedures is reported. Here we perform one 10-fold cross-validation. The results are presented in Table 1 and Table 2 where also previously published results are reported.

The classification accuracies of 1NN are quite close to the ones previously reported in the literature. The high 1NN classification accuracies on the MUSK1 and MUSK2 data set indicate that the bags are well separated in the obtained bag dissimilarity space defined by D. Fisher performs poorly when built on D with an average classification accuracy of 62.1% whereas SVM performs decent when built on D with an average classification accuracy of 78.4%. However, building them on a second dissimilarity representation $D2$ constructed from D, as described in Section 4.3, improves performance considerably for Fisher with an average absolute increase of 19.3% and slightly for SVM with an average absolute increase of 4%. 1NN performs slightly worse when applied to $D2$ compared to D, and the numbers reported in Table 1 and Table 2 for 1NN are therefore based on D. SVM and Fisher generally perform better than 1NN. We also tried kNN with k optimized using cross-validation on the training set in each fold which achieved similar performance to 1NN.

Across all five data sets, SVM and Fisher built on dissimilarity representations show excellent performance. On the MUSK1 and MUSK2 data set, the classifiers achieve accuracies close to the best reported accuracies in the literature. On the

Table 2. Classification accuracy on the image retrieval data. See the caption of Table 1 for further details.

Classifier	Bag dissimilarity measure	elephant	fox	tiger
1NN (on D)	d_{min} (1)	78.0 / 78.0	60.0 / 59.5	77.0 / 74.0
	d_H (2)	70.0 / 69.5	52.0 / 50.0	67.0 / 64.5
	d_{BWmax} (3)	75.0 / 77.5	57.5 / 57.0	68.0 / 66.0
	d_{BWmean} (4)	80.0 / 79.0	59.5 / 59.0	70.5 / 71.5
SVM (on $D2$)	d_{min} (1)	85.5 / 83.5	**67.5** / 65.0	77.5 / 78.0
	d_H (2)	84.0 / 84.5	37.5 / 49.0	73.5 / 73.5
	d_{BWmax} (3)	**89.0** / **89.0**	64.5 / 56.0	69.5 / 62.0
	d_{BWmean} (4)	87.0 / 87.0	62.5 / 58.5	78.0 / 76.5
Fisher (on $D2$)	d_{min} (1)	86.0 / 84.5	66.0 / **66.0**	78.5 / 78.0
	d_H (2)	84.5 / 85.0	59.0 / 59.0	73.5 / 72.0
	d_{BWmax} (3)	88.5 / 88.5	66.5 / 63.0	81.0 / 78.5
	d_{BWmean} (4)	**89.0** / 88.5	64.5 / 64.0	**81.5** / **79.5**
mi-SVM [4]		- / 82.2	- / 58.2	- / 78.9
MI-SVM [4]		- / 81.4	- / 59.4	- / 84.0
MIGraph [7]		- / 85.1	- / 61.2	- / 81.9
miGraph [7]		- / 86.8	- / 61.6	- / **86.0**

image retrieval data sets, SVM with a linear kernel, as well as Fisher, perform better than the SVM's adapted to MIL problems [4] in two out of three data sets. This indicates that taking instance relations into account is beneficial in this kind of problems, as is also seen in [7].

6 Discussions and Conclusions

The linear classifiers built on the proposed dissimilarity representations performed better than the best results in the MIL literature in some cases, and in the remaining cases close to the best published results [1,2,10,4,5,8,6,7]. It should be noted that the classifiers were applied "off the shelf" and that, e.g., the trade-off parameter C in SVM was not tuned by cross-validation but fixed to 1. Also, the classifiers were trained and tested in dissimilarity spaces of dimension equal to the number of training samples. This is no problem for SVM. For Fisher, the pseudo-inverse was used. It may be possible to obtain even better results than the ones reported in Table 1 and Table 2 by proper regularization or by reducing the dimensionality of the dissimilarity space, e.g., by prototype selection [15].

SVM shows worse than random performance on some of the image retrieval data sets, in particular when built on the dissimilarity representation obtained using the Hausdorff distance, d_H, on the fox data set. This could be caused by a strong class overlap in the dissimilarity space. This is also indicated by the fact that both 1NN and Fisher perform worse on this representation compared to the other representations.

The minimum point set distance, d_{min}, works well as bag dissimilarity measure. Similar results were reported in [10]. This is somewhat surprising since classes are expected to be overlapping in MIL due to positive bags also containing negative instances. The explanation is that the distribution of the positive instances is more dense compared to the negative instances in the used data sets, and therefore a bag containing at least one positive instance is more likely to be close to another bag containing at least one positive instance than to a bag containing only negative instances.

To conclude, we have shown that the dissimilarity representation approach can be used to solve MIL problems. Global decision rules in the form of general purpose supervised linear classifiers built in a bag dissimilarity space achieves excellent classification accuracies on publicly available MIL data sets. The approach is general, and we see this as a promising direction that allows for using a wider range of proximity measures between bags in solving MIL problems compared to the popular kernel-based approaches. Further, there are indications that taking relations among instances into account improves the performance on certain MIL problems, such as the image retrieval problems.

Acknowledgements. This work is partly funded by the Danish Council for Strategic Research (NABIIT); the Netherlands Organisation for Scientific Research (NWO); AstraZeneca, Lund, Sweden; the FET programme within the

EU FP7, under the SIMBAD project (contract 213250); and the ICT-2007.2.2 programme within the EU FP7, under the SSPNet project (contract 231287).

References

1. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artif. Intell. 89(1-2), 31–71 (1997)
2. Maron, O., Lozano-Pérez, T.: A framework for multiple-instance learning. In: Jordan, M.I., Kearns, M.J., Solla, S.A. (eds.) NIPS. The MIT Press, Cambridge (1997)
3. Cannon, A., Hush, D.: Multiple instance learning using simple classifiers. In: ICMLA, pp. 123–128 (2004)
4. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: Becker, S., Thrun, S., Obermayer, K. (eds.) NIPS, pp. 561–568. MIT Press, Cambridge (2002)
5. Gärtner, T., Flach, P.A., Kowalczyk, A., Smola, A.J.: Multi-instance kernels. In: Sammut, C., Hoffmann, A.G. (eds.) ICML, pp. 179–186. Morgan Kaufmann, San Francisco (2002)
6. Tao, Q., Scott, S.D., Vinodchandran, N.V., Osugi, T.T., Mueller, B.: Kernels for generalized multiple-instance learning. IEEE Trans. Pattern Anal. Machine Intell. 30(12), 2084–2098 (2008)
7. Zhou, Z.H., Sun, Y.Y., Li, Y.F.: Multi-instance learning by treating instances as non-i.i.d. samples. In: Danyluk, A.P., Bottou, L., Littman, M.L. (eds.) ICML. ACM International Conference Proceeding Series, vol. 382, pp. 1249–1256. ACM, New York (2009)
8. Chen, Y., Bi, J., Wang, J.Z.: MILES: Multiple-instance learning via embedded instance selection. IEEE Trans. Pattern Anal. Machine Intell. 28(12), 1931–1947 (2006)
9. Pekalska, E., Duin, R.P.W.: Dissimilarity representations allow for building good classifiers. Pattern Recog. Lett. 23(8), 943–956 (2002)
10. Wang, J., Zucker, J.D.: Solving the multiple-instance problem: A lazy learning approach. In: Langley, P. (ed.) ICML, pp. 1119–1126. Morgan Kaufmann, San Francisco (2000)
11. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision 40(2), 99–121 (2000)
12. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
13. Duin, R.P.W., Juszczak, P., Paclík, P., Pekalska, E., de Ridder, D., Tax, D.M.J.: PRTools, a Matlab toolbox for pattern recognition, version 4.0 (January 2004)
14. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001) Software available at, http://www.csie.ntu.edu.tw/~cjlin/libsvm
15. Pekalska, E., Duin, R.P.W., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recognition 39(2), 189–208 (2006)

A Game Theoretic Approach to Learning Shape Categories and Contextual Similarities

Aykut Erdem and Andrea Torsello

Dipartimento di Informatica
Universitá "Ca' Foscari" di Venezia
{erdem,torsello}@dsi.unive.it

Abstract. The search of a model for representing and evaluating the similarities between shapes in a perceptually coherent way is still an open issue. One reason for this is that our perception of similarities is strongly influenced by the underlying category structure. In this paper we aim at jointly learning the categories from examples and the similarity measures related to them. There is a chicken and egg dilemma here: class knowledge is required to determine perceived similarities, while the similarities are needed to extract class knowledge in an unsupervised way. The problem is addressed through a game theoretic approach which allows us to compute 2D shape categories based on a skeletal representation. The approach provides us with both the cluster information needed to extract the categories, and the relevance information needed to compute the category model and, thus, the similarities. Experiments on a database of 1000 shapes showed that the approach outperform other clustering approaches that do not make use of the underlying contextual information and provides similarities comparable with a state-of-the-art label-propagation approach which, however, cannot extract categories.

1 Introduction

The unsupervised learning of shape categories is a central problem in computer vision with significant theoretical and practical impacts. There are two interrelated aspects to the problem: The first is the discovery of the shape categories present, and this can be effectively addressed as a problem of clustering shapes, while the second is the generalization of the class properties, i.e., the ability to assign each newly encountered shape to one of the extracted classes, or to recognize it as an outlier. Fundamental to both tasks is the problem of determining how similar two shapes really are.

These issues have been extensively studied with geometric characterizations of shape using both simple descriptors such as landmark points on the boundary [4], or more complex ones such as curve descriptors [8]. Shape-classes can then be located by vectorizing the shape-attributes and applying standard central clustering techniques to the shape-vectors, while the problem of determining the membership to a class can be solved by performing principal components analysis. An alternative to the use of a single vectorial representation of the shape's geometry is to use a structural abstraction where the object is divided

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 139–148, 2010.

into atomic components whose arrangement is then represented using a relational graph [7,20]. Typically in this context, the similarity between two shapes is a measure of how well the primitives forming the shapes and/or their spatial organizations agree, and the assessment of whether a shape belongs to a particular class is performed by comparing in isolation the shape to one or more prototypes and by applying the nearest neighbor rule, while categories can be extracted using pairwise clustering [15].

One problem with these approaches is that they all assume the existence of a single *universal* measure of similarity between shapes, often requiring metric properties as well, while psychological experiments suggest that the human perception of similarity is not only non-metric [5], but also strongly dependent on the surrounding context [17,11]. In particular, the observed variation within a shape-class is fundamental for determining the perception of the similarities of the shapes belonging to that class. Recently, this issue has also been surfaced from a computational point of view [19,9].

In this paper, we propose a game theoretic approach to compute shape categories in an unsupervised way. There is a chicken and egg problem here: Class knowledge is required to determine perceived similarities, while the similarities are needed to extract class knowledge. We solve this problem using a EM-like approach where we iteratively estimate the class memberships and maximize for the parameters of our category representation. The expectation of class membership is obtained by adopting a game theoretic clustering framework presented in [16]. Then the similarities are computed as the edit-distance of a skeletal representation presented in [3] using the newly estimated cost coefficients. Central to the approach is the ability of the clustering framework to provide both the cluster information needed to extract the categories, and the relevance information (or the degree of membership) needed to compute the category model, and thus the similarities, in a robust way. Interestingly, the contextual similarity defined in [3] is not symmetric, making the ability of the game-theoretic approach to deal with asymmetric affinities particularly attractive.

2 Disconnected Skeletons and Category Influenced Matching

Skeletons are one of the most common representation scheme for generic shape recognition [20,14], as they capture part structure and provide insensitivity to articulations and occlusions. However, in practice, two visually very similar shapes might have structurally different skeletons, hence this instability issue should be resolved either in extracting the skeleton or in the matching process. In this regard, *disconnected skeletons* [2,1] provide an alternative solution as the method aims at obtaining a coarse yet a very stable skeleton representation from scratch.

Disconnected skeletons are defined in terms of a special distance surface (Aslan surface), the level curves of which are increasingly smoothed versions of the initial shape boundary, and which has a single extremum point that captures the center of a blob-like representation of the shape (Fig. 1(a)). Each branch extracted from this surface is classified as either *positive* or *negative*, identifying

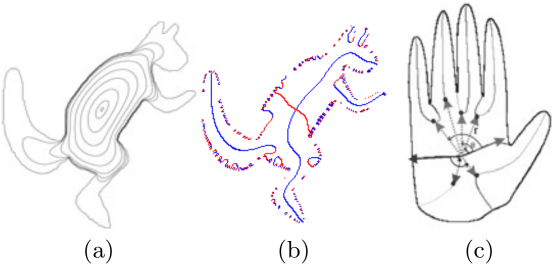


Fig. 1. Disconnected skeletons. (a) Level curves of Aslan surface (b) Positive and negative skeleton branches, respective drawn in blue and red (before pruning) (c) Spatial organization of skeleton branches (taken from [3]).

whether it originates from a positive curvature maxima (a protrusion) or a negative curvature minima (an indentation). Among the extracted branches, at least two positive and two negative branches reach the shape center, and these are called *major branches* since they represent the most prominent visual features of the shape. All the other branches terminate at some *disconnection point*s where a positive branch and a negative branch collide (Fig. 1(b)). It has been shown that these points are very stable under visual transformations such as articulation and deformation of parts. The skeletal attributes used to represent each skeleton branch are simply its type, the location of its disconnection points (r, θ), and its length l measured in the formed coordinate frame.

In [3], disconnected skeletons are represented as *rooted attributed depth-1 tree*s and tree-edit distance is used to match these structures. Moreover, Baseski and coworkers [3] used the category of one of the shapes to be matched to determine the edit-costs. The cost functions are computed on the basis of category specific statistics about the skeletal attributes that are stored in an auxiliary tree union structure. In this version, the cost function for the label `change` operation is defined in terms of a generic cost function. The idea resembles Mahalanobis distance in that when the distance within the observed range of skeletal attributes, but rapidly increases outside of that region.

3 Grouping Game

In [16], a novel framework for grouping and clustering was presented which was derived from a game-theoretic formalization of the competition between the hypotheses of group membership. The basic idea is as follows: Let the hypotheses that each element belongs to a group compete with one-another, each obtaining support from compatible elements and competitive pressure from all the others. Competition will reduce the population assuming hypotheses that do not receive strong support from the rest, while it will allow populations assuming hypotheses with strong support to thrive. Eventually, all inconsistent hypotheses will be driven to extinction, while all the surviving hypotheses will reach an equilibrium with all receiving the same average support. Clustering was thus formalized as a repeated non-cooperative game where competition for class membership selects elements belonging to a coherent cluster.

Specifically, let $O = \{1, \cdots, n\}$ be the set of available elements, for each pair of strategies $i, j \in O$, a_{ij} represents the payoff of an individual playing strategy i against an opponent playing strategy j. A *mixed strategy* is a probability distribution $\mathbf{x} = (x_1, \ldots, x_n)^T$ over the available strategies O.

$$\Delta = \left\{ \mathbf{x} \in \mathbb{R}^n \ : \ x_i \geq 0 \text{ for all } i \in O, \ \mathbf{1}^T \mathbf{x} = 1 \right\} ,$$

where $\mathbf{1} = (1, \ldots, 1)^T$, while the *support* of a mixed strategy $\mathbf{x} \in \Delta$, denoted by $\sigma(\mathbf{x})$, is defined as the set of elements chosen with non-zero probability: $\sigma(\mathbf{x}) = \{i \in O \mid x_i > 0\}$.

The expected payoff received by a player choosing element i when playing against a player adopting a mixed strategy $\mathbf{x}$ is $(A\mathbf{x})_i = \sum_j a_{ij} x_j$, hence the expected payoff received by adopting the mixed strategy $\mathbf{y}$ against $\mathbf{x}$ is $\mathbf{y}^T A \mathbf{x}$.

The *best replies* against mixed strategy $\mathbf{x}$ is the set of mixed strategies

$$\beta(\mathbf{x}) = \{\mathbf{y} \in \Delta \mid \mathbf{y}^T A \mathbf{x} = \max_{\mathbf{z}} (\mathbf{z}^T A \mathbf{x})\} .$$

A strategy $\mathbf{x}$ is said to be a *Nash equilibrium* if it is the best reply to itself, i.e.,

$$\forall \mathbf{y} \in \Delta \quad \mathbf{x}^T A \mathbf{x} \geq \mathbf{y}^T A \mathbf{x} . \tag{1}$$

Within this formalization, Nash equilibria abstracts the main characteristics of a group: internal homogeneity, that is, a high mutual support of all elements within the group, and external dishomogeneity, or low support from elements of the group to elements that do not belong to the group. Equilibria, and thus groups, are found using the replicator dynamics [18], a well-known formalization of a natural selection process.

The main characteristics of the framework are that it is generic, as it can deal with asymmetric as well as negative affinities; it does not require *a priori* knowledge of the number of clusters as it is inherently a multi-figure/ground discrimination process; and it provides immediate measures of both the cohesiveness of the cluster in the form of its average payoff $\mathbf{x}^T A \mathbf{x}$, and of the participation of an element to the cluster. In fact the value x_i can be interpreted as a degree of participation of element i to the cluster defined by the stable point $\mathbf{x}$.

4 The Proposed Method

In this study, we attempt to solve the interrelated problems of discovering shape categories and computing the corresponding contextual similarities using a EM-like approach where we iteratively estimate the class memberships and maximize for the parameters of our category representation. The expectation of class membership is obtained by adopting the game theoretic clustering framework summarized in Section 3. Then the similarities are computed as the edit-distance of a skeletal representation presented in [3] using the newly estimated cost coefficients. The details of these steps are as follows.

4.1 Discovering Shape Categories

We define the shape category in terms of a clustering game where shapes present in the training set compete for category membership. The outcome of the competition is determined by the payoff or utility matrix $A = (a_{ij})$ which represents the similarity of shape i with shape j. Initially, these payoffs simply correspond to the similarities among the given set of shapes obtained with $a_{ij} = \exp\left(-\frac{(\mathrm{dist}(i,j))^2}{\sigma^2}\right)$ where σ is a scaling factor, and $\mathrm{dist}(i, j)$ is the tree-edit distance between the disconnected skeletons of the shapes i and j.

Since no category information is available in the beginning, the initial similarities were computed in isolation without any context, thus A is a *symmetric* matrix. However, in the subsequent iterations, the category structure discovered in the previous step influences the similarity computations by differentiating the roles of the shapes in comparison. Now, each row index corresponds to a query shape whereas each column index is a shape which has a category label assigned by the previous grouping game (if it is not found to be an outlier), and the cost functions are determined by the context about the category of the second shape. Thus, this results in an *asymmetric* similarity matrix.

Given the payoff matrix A, we extract shape categories by applying a *peal-off* strategy. At first, we start with a grouping game that considers all the shapes and we extract a cluster by running the replicator dynamics. Following that, we define a new game on the set of remaining (unlabeled) shapes and reiterate the procedure until all groups are extracted. The game theoretic framework also provides us a direct way to evaluate the coherency of extracted clusters. Let $S \in \mathcal{S}$ be an extracted group, the coherency of S can be computed as its average payoff $\mathbf{x}_S^{\mathrm{T}} A \mathbf{x}_S \in [0, 1]$. By inspecting these values, we obtain an initial set $\mathcal{C}$ $(\subseteq \mathcal{S})$ of coherent shape categories which is formed by the clusters $S \in \mathcal{S}$ with $\mathbf{x}_S^{\mathrm{T}} A \mathbf{x}_S > \zeta_1$. This allows us to discard incoherent classes hence enforcing robustness in the extraction process.

The payoff information can also be used to assign additional members to the clusters in C. To compute the similarity between a shape i to a cluster S, we use the weighted similarity function $\gamma_S(i) = \frac{(A\mathbf{x}_S)_i}{\mathbf{x}_S^{\mathrm{T}} A \mathbf{x}_S}$. We evaluate this similarity measure for every unlabeled shape i and assign it to the most similar cluster if $\gamma_S(i) \geq \zeta_2$. Otherwise, it is considered as an outlier shape which does not belong to any of the extracted categories. The ability of assigning an unclustered object either to a category or to the *outlier* class is instrumental to the generalization capabilities. Note that the outlier class should be interpreted as a "don't know" label where the approach cannot say anything about the shape rather than recognizing the shape as a new class not seen in the other examples.

After reassigning the leftover elements, we re-examine the groups that were rejected by the first thresholding step and check whether they became more coherent with the removal of the reassigned elements. To evaluate their coherency we use an hysteresis strategy: we accept the groups with $|S| > 3$ whose average payoffs $\mathbf{x}_S^{\mathrm{T}} A \mathbf{x}_S > \zeta_3$, with $\zeta_3 < \zeta_1$. The purpose of this hysteresis is to reduce the effect the implicit change in scale induced by the peel-off strategy and to increase robustness with respect to the scaling factor σ.

4.2 Computing Contextual Similarities

To model the influence of the discovered category structure on the computation of shape similarities, we adopt the tree-edit distance based shape matching method proposed by Baseski *et al.* [3]. Here, however, we form the union in an unsupervised way, based on the clusters obtained with the game-theoretic approach. Further, in the computation of the edit-cost, we substitute the minimum and maximum values of the skeletal attributes in the category with soft bounds that make use of the membership information supplied by the clustering framework. In particular, we use the weighted mean $\mu_{\mathbf{x}}$ and weighted standard deviation $\sigma_{\mathbf{x}}$ (Eqn. 2) to determine the range $\mu_{\mathbf{x}} \pm 3\sigma_{\mathbf{x}}$ which has experimentally shown to account for the shape variability and provide a robust inference process.

$$\mu_{\mathbf{x}} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i}, \quad \sigma_{\mathbf{x}} = \sqrt{\frac{1}{1 - \sum_{i=1}^{n} x_i^2} \sum_{i=1}^{n} x_i (y_i - \mu_{\mathbf{x}})^2} \tag{2}$$

In obtaining the affinity matrix $A = (a_{ij})$ at time step $t > 0$, we introduce a soft indexing scheme where we propagate the information about the similarities to the extracted classes: When computing the similarity between the query shape i and the shape j, if j belongs to a cluster S extracted in the previous step, we multiply the similarity influenced by the new category information, with the similarity of the shape i to the cluster S normalized with respect to the most similar category. This allows us to bias the similarities towards the previously extracted clusters, thus propagating the membership throughout the iterations. Clearly, if j is an outlier shape, *i.e.* no category information is available about it, we keep the original distance which does not utilize any context information. Moreover, the corresponding multiplier b_{ij} is taken as 1.

$$a_{ij} = b_{ij} \times exp\left(-(dist(i,j))^2)/\sigma^2\right) \tag{3}$$

$$\text{where } b_{ij} = \begin{cases} 1 & \text{if } j \text{ is an outlier} \\ \dfrac{\gamma_S(i)}{\max\limits_{T \in \mathcal{C}} \gamma_T(i)} & \text{if } j \in S \end{cases}$$

Category discovery and similarity computation are iterated until the change in the ratio of unlabeled (outlier) shapes to the total number of shapes is smaller than a threshold ζ_4. Experimentally it was observed that the resulting group and distance information, as well as the query performance, are relatively stable after meeting this condition.

5 Experimental Results

In order to evaluate the performance of the proposed approach, we used the shape database provided in [3] which contains a total of 1000 shapes from 50 shape categories, each having 20 examples. We start by extracting the disconnected skeletons. After the descriptions are formed, we iteratively run the proposed method with the empirically set parameters $\sigma^2 = 24$, $\zeta_1 = 0.85$, $\zeta_2 = 0.95$ and

Table 1. The quantitative evaluation of the clustering results

The Method	Rand Index	Corrected Rand Index	NMI
Our method at t=0	0.9818	0.9929	0.8517
Our method (asymmetric case)	0.9854	0.9933	0.8722
Normalized Cut [13] (with # of classes=51)	0.9832	0.9833	0.8381
Normalized Cut [13] (with # of classes=61)	0.9848	0.9854	0.8380
Foreground Focus [9] (with # of classes=50)	0.9748		0.7329

Table 2. The final shape categories extracted from asymmetric affinities. The number of outlier shapes is 80.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Precision	1.00	0.67	1.00	0.95	1.00	0.95	1.00	1.00	0.87	0.70	0.67	1.00	1.00	0.91	0.95
Recall	0.90	0.70	0.85	0.90	0.90	0.95	0.90	0.90	0.65	0.35	0.60	0.95	0.80	1.00	1.00
Payoff	0.96	0.95	0.92	0.94	0.95	0.94	0.96	0.97	0.91	0.88	0.94	0.94	0.94	0.94	0.92
Entropy	2.75	2.77	2.69	2.72	2.81	2.77	2.77	2.89	2.47	1.60	2.75	2.63	2.39	2.94	2.87

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Precision	1.00	0.50	0.75	1.00	0.94	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Recall	0.65	0.35	0.15	1.00	0.75	0.70	0.45	0.20	0.90	0.60	0.40	0.45	0.25	0.25	0.80
Payoff	0.94	0.94	0.77	0.96	0.93	0.93	0.92	0.84	0.94	0.93	0.93	0.87	0.87	0.80	0.93
Entropy	2.39	2.55	1.34	2.92	2.59	2.46	2.03	1.38	2.78	2.44	2.03	2.13	1.61	1.57	2.71

	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
Precision	1.00	0.62	1.00	0.83	1.00	0.37	0.86	0.92	1.00	1.00	0.91	0.95	1.00	1.00	1.00
Recall	0.50	0.50	0.55	0.25	0.80	0.35	0.60	0.60	0.95	0.50	0.50	0.95	0.90	0.25	1.00
Payoff	0.91	0.87	0.92	0.89	0.94	0.94	0.92	0.93	0.95	0.94	0.92	0.91	0.93	0.80	0.95
Entropy	2.10	2.19	2.28	1.61	2.68	2.75	1.94	2.43	2.79	2.30	2.38	2.72	2.78	1.58	2.77

	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
Precision	1.00	0.75	1.00	1.00	1.00	1.00	0.74	1.00	1.00	1.00	0.95	0.94	1.00	1.00	0.55
Recall	0.80	0.15	0.95	0.90	0.35	1.00	0.70	0.75	0.90	0.60	0.95	0.75	0.20	0.85	0.55
Payoff	0.90	0.83	0.96	0.95	0.84	0.97	0.95	0.95	0.95	0.91	0.93	0.93	0.86	0.96	0.94
Entropy	2.56	1.38	2.92	2.69	1.85	2.93	2.63	2.70	2.81	2.33	2.66	2.55	1.39	2.83	2.85

	61	62	63	64
Precision	0.78	1.00	1.00	0.42
Recall	0.35	0.55	0.30	0.50
Payoff	0.90	0.92	0.85	0.91
Entropy	2.01	2.27	1.61	2.80

$\zeta_3 = 0.75$, and stop when $\zeta_4 \leq 0.005$. In this setting, the algorithm converges at the 2^{nd} iteration. The shape categories extracted are given in Table 2, where for each class we show the shape with the highest membership score.

Table 1 shows some cluster validity measures [6] on the classes extracted with our approach. The first measure is the standard Rand index, *i.e.* the ratio of agreements over all possible pairs. The second measure is a corrected version of the Rand index where the disagreements in the outlier class are not penalized, as this class is not supposed to form a coherent group. Note that the latter form of the Rand index favors more conservative approaches, where we prefer the approach not to label a shape when in doubt, while the former version favors bolder assignments where we prefer to make a few mistakes rather than not assign a shape to a class. Which version is to be preferred is clearly dependent on the application. The last measure is the normalized mutual information (NMI) which measures the closeness between the class distributions and the ground truth.

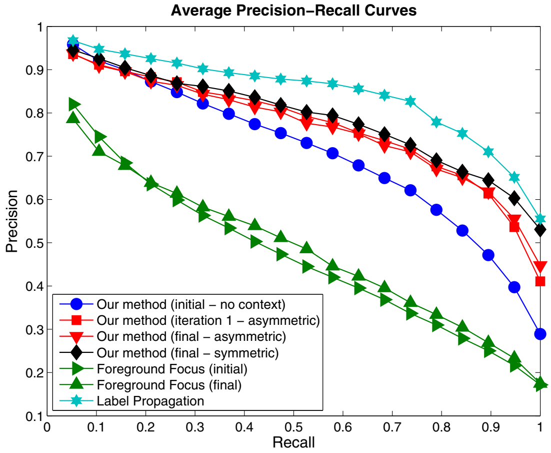


Fig. 2. Average precision-recall curves

In an attempt to assess information content in the asymmetry of the similarity matrix, we also perform the same experiment using the same parameters but rendering the affinities symmetric before applying pairwise clustering. In this case, the approach requires 3 iterations to converge. When the number of outlier shapes and the average precision recall values (Fig. 2) are considered alone, the symmetric case seems to work better than the asymmetric case. However, the difference between the plain and corrected Rand index show that the asymmetric approach is more conservative, i.e. it has a higher tendency to label shapes as belonging to an unknown class, but makes fewer misclassifications when it does assign shapes to a class, on the other hand the symmetric approach is more likely to assign shapes to a class, even when this results in more misclassifications.

We compared the results with several alternatives. The first, which should be seen as a baseline comparison, is performed by applying a pairwise clustering approach in order to extract the class structure, while assuming global, non-contextual similarities. Here we used Normalized Cut [13] as a baseline pairwise clustering approach. Note that the normalized cut approach requires the number of classes to be known *ab initio*. Here we choose two different values: 51 (the existing 50 semantic categories plus 1 for the outliers) and 61 (a number closer to the number of categories extracted with our approach). The additional number of classes is due to the fact that there can be a substantial semantic gap between appearance and categories, and allowing more freedom can result in better overall categorization. Indeed, as it can be seen in Table 1, normalized cut performs better with more degrees of freedom, but still performs significantly worse than the proposed approach.

The second approach we are comparing against is *Foreground Focus* [9]. This is an unsupervised algorithm proposed to learn categories from sets of partially matching image features. Just like our approach, it utilizes an EM-like algorithm to infer the categories. However, its goal is to learn relevant features rather than the actual contextual similarities. In order to compare with this method, we first form *Inner-Distance Shape Context* [10] descriptions of each shape by uniformly sampling 100 landmark points across the shape boundary and using a total of 5 inner-distance bins and 12 inner-angle bins. Earth Mover's Distance (EMD) algorithm [12] is then used to compute the matchings of shape features and similarities, and Normalized Cut [13] is used to determine the clusters. Here, the total number of extracted clusters is kept fixed at 50 (the actual number of shape categories exist in the database). Table 1 and Fig. 2 show that the performance of this approach is even significantly lower than the baseline normalized cuts over the skeletal distance. The huge difference can probably be explained by the lower descriptive power of the Inner-Distance Shape Context features with respect to disconnected skeletons, or bad performance of EMD matching algorithm.

The last comparison is with the label propagation method [19] and is limited to the retrieval performance of the contextual similarities. This method has three parameters which are used to construct the affinity matrix, the neighborhood size and the window size that are respectively set as $C = 0.275$, neighborhood size $K = 10$, window size $W = 250 \times 250$. When applied to the initial (symmetric) similarities, the approach offers a slightly better precision/recall (Fig. 2). However, note that the approach solves a slightly different problem; it concentrates only on improving retrieval rate and does not provide any category structure or an estimation of perceptually relevant similarities.

6 Summary and Conclusion

In this paper, we presented an approach for the simultaneous discovering of 2D shape categories and the corresponding contextual similarities. This was achieved by adopting the game theoretic clustering approach introduced in [16] and by modifying the shape retrieval system presented in [3] in order to account for the uncertainty in the category information. The game theoretic framework naturally provides us the membership information about the extracted categories which quantifies this uncertainty, and is capable of dealing with the asymmetric similarities obtained using the contextual information. We have demonstrated the potential of the proposed framework on a large shape database composed of highly varying 1000 shapes from 50 categories.

Acknowledgment

We acknowledge the financial support of the Future and Emerging Technology (FET) Programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open project SIMBAD grant no. 213250.

References

1. Aslan, C., Erdem, A., Erdem, E., Tari, S.: Disconnected skeleton: Shape at its absolute scale. IEEE Trans. Pattern Anal. Mach. Intell. 30(12), 2188–2203 (2008)
2. Aslan, C., Tari, S.: An axis-based representation for recognition. In: ICCV, vol. 2, pp. 1339–1346 (2005)
3. Baseski, E., Erdem, A., Tari, S.: Dissimilarity between two skeletal trees in a context. Pattern Recognition 42(3), 370–385 (2009)
4. Cootes, T.F., Taylor, C.J., Cooper, D.H.: Active shape models - their training and application. Comput. Vis. Image Underst. 61(1), 38–59 (1995)
5. Jacobs, D.W., Weinshall, D., Gdalyahu, Y.: Classification with nonmetric distances: Image retrieval and class representation. IEEE Trans. Pattern Anal. Mach. Intell. 22(6), 583–600 (2000)
6. Jain, A.K., Dubes, R.C.: Algorithms for clustering data. Prentice-Hall, Inc., Upper Saddle River (1988)
7. Kimia, B.B., Tannenbaum, A.R., Zucker, S.W.: Shapes, shocks and deformations i: The components of two-dimensional shape and the reaction-diffusion space. Int. J. Comput. Vision 15(3), 189–224 (1995)
8. Klassen, E., Srivastava, A., Mio, W., Joshi, S.H.: Analysis of planar shapes using geodesic paths on shape spaces. IEEE Trans. Pattern Anal. Mach. Intell. 26(3), 372–383 (2004)
9. Lee, Y.J., Grauman, K.: Foreground focus: Unsupervised learning from partially matching images. International Journal of Computer Vision (2009)
10. Ling, H., Jacobs, D.: Shape classification using the inner-distance. IEEE Trans. Pattern Anal. Mach. Intell. 29(2), 286–299 (2007)
11. Mumford, D.: Mathematical theories of shape: Do they model perception? In: Vemuri, B.C. (ed.) Geometric Methods in Computer Vision. SPIE, vol. 1570, pp. 2–10 (1991)
12. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision 40(2), 99–121 (2000)
13. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 22(8), 888–905 (2000)
14. Siddiqi, K., Kimia, B.B.: A shock grammar for recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1996)
15. Torsello, A., Robles-Kelly, A., Hancock, E.R.: Discovering shape classes using tree edit-distance and pairwise clustering. International Journal of Computer Vision 72(3), 259–285 (2007)
16. Torsello, A., Bulo, S.R., Pelillo, M.: Grouping with asymmetric affinities: A game-theoretic perspective. In: CVPR, pp. 292–299 (2006)
17. Tversky, A.: Features of similarity. Psychological Review 84, 327–352 (1977)
18. Weibull, J.W.: Evolutionary Game Theory. MIT Press, Cambridge (1995)
19. Yang, X., Bai, X., Latecki, L.J., Tu, Z.: Improving shape retrieval by learning graph transduction. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 788–801. Springer, Heidelberg (2008)
20. Zhu, S.C., Yuille, A.L.: Forms: A flexible object recognition and modeling system. Int. J. Comput. Vision 20(3), 187–212 (1996)

A Comparison between Two Representatives of a Set of Graphs: Median vs. Barycenter Graph

Itziar Bardaji, Miquel Ferrer, and Alberto Sanfeliu

Institut de Robòtica i Informàtica Industrial, UPC-CSIC, Spain
{ibardaji,mferrer,sanfeliu@iri.upc.edu}

Abstract. In this paper we consider two existing methods to generate a representative of a given set of graphs, that satisfy the following two conditions. On the one hand, that they are applicable to graphs with any kind of labels in nodes and edges and on the other hand, that they can handle relatively large amount of data. Namely, the approximated algorithms to compute the Median Graph via graph embedding and a new method to compute the Barycenter Graph. Our contribution is to give a new algorithm for the barycenter computation and to compare it to the median Graph. To compare these two representatives, we take into account algorithmic considerations and experimental results on the quality of the representative and its robustness, on several datasets.

1 Introduction

The straight advantages of the use of graphs for representation purposes appear to be useless in some applications due to the lack of mathematical structure in the graph domain. An illustrative example is the problem of finding a representative of a set of graphs. While in vector spaces it is easy to compute representatives such as medians and means with respect to a wide range of distances, in the graph domain the analogy turns out to be a highly non–trivial task.

In the literature we can distinguish different methodologies to tackle this problem, both probabilistic and deterministic. Random Graphs such as First-Order Random Graphs (FORGs) [1], Function-Described Graphs (FDGs) [2,3] and Second-Order Random Graphs (SORGs) [4]; a Maximally General Prototype [5]; the Median Graph [6] and the Barycenter Graph [7] have been proposed as representatives of a set of graphs, among others. Most of these methods suffer from a prohibitive computation time or are limited to a restricted family of graphs.

In this paper we aim to compare those algorithms that on the one hand are applicable to graphs with any kind of labels in nodes and edges and on the other can handle relatively large amount of data. Namely, the approximated algorithms to compute the Median Graph via graph embedding and a new method to compute the Barycenter Graph. It includes some algorithmical considerations and experiments on several real–world and artificial datasets.

This paper is organized as follows. Some basic definitions are given in Section 2, the computation of the Median Graph is discussed in Section 3 and the proposed computation for the barycenter graph is presented in Section 4. Section 5

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 149–158, 2010.

is devoted to comparing the Median and Barycenter Graphs. Finally, in Section 6 we draw some conclusions.

2 Definitions

Throughout the paper, let $S = \{g_1, g_2, ..., g_n\}$ be a set of graphs and let L be the set of labels of the nodes and edges of the graphs of S. Let U be the set of all graphs that can be constructed using labels from L, and observe that $S \subseteq U$. Also, let $d : U \times U \to \mathbb{R}$ be a distance over the set U.

Given a set of graphs S, the Set and Generalized Median Graphs [6] are defined as follows.

Definition 1. *The* set Median Graph $\hat{g}$ *and the* generalized Median Graph $\bar{g}$ *of S are defined as:*

$$\hat{g} = arg \ \underset{g \in S}{min} \sum_{g_i \in S} d(g, g_i) \qquad and \qquad \bar{g} = arg \ \underset{g \in U}{min} \sum_{g_i \in S} d(g, g_i).$$

The set Median Graph is a graph of the set S which minimizes the *sum of distances* (SOD) to all the graphs in S. The generalized Median Graph $\bar{g}$, is also a graph that minimizes the SOD to all the graphs in S, but the minimum is taken over U. Thus, the generalized Median Graph does not necessarily belong to the original set S. Since the minimum is taken over a larger set, the generalized Median Graph is expected to be a better representative for the set S of graphs. Notice that in general more than one set and generalized Median Graph may exist for a given set S.

Note that the median graph is analogous to the concept of median vector in a vector space. Similarly, the definition of Barycenter Graph is natural, by adapting the definition of barycenter of a set of points in $\mathbb{R}^n$.

Definition 2. *The* set Barycenter Graph $\hat{b}$ *and the* generalized Barycenter Graph *(or just* Barycenter Graph*)* $\bar{b}$ *of S are defined as:*

$$\hat{b} = arg \ \underset{g \in S}{min} \sum_{g_i \in S} d(g, g_i)^2 \qquad and \qquad \bar{b} = arg \ \underset{g \in U}{min} \sum_{g_i \in S} d(g, g_i)^2.$$

That is, the Barycenter Graph is the graph in U minimizing the *sum of squared distances* (SOSD) to all the graphs in S. The set barycenter is the argument minimizing the SOSD, when the search is limited to the given set S.

Although definitions 1 and 2 apply for any distance, we let d be the well known *graph edit distance* [8]. This choice makes it possible to apply the algorithms below to sets of graphs of different sizes and with any kind of labels.

Finally, we introduce the notion of *weighted mean*, first presented in [9].

Definition 3. *Let g, g' be graphs. Let $I = \{h \in U \mid d(g, g') = d(g, h) + d(h, g')\}$, be the set of* intermediate graphs*. Given $0 \leq a \leq d(g, g')$, the* weighted mean *of g and g' is a graph*

$$g'' = WM(g, g', a) = \arg\min_{h \in I} |d(g, h) - a|.$$

That is, given two graphs, g and g', and a parameter a, the weighted mean is an intermediate graph, not necessarily unique, whose distance to g is as similar as possible to a. Consequently, its distance to g' is also the closest to $d(g, g') - a$. Again, we let d be the graph edit distance.

Remark 1. Note that, the so called *error*, $\epsilon(a) = |d(g, g'') - a|$, is not necessarily null. This fact, regardless of the exactness of the computation, depends on the properties of the search space U.

3 Computation of the Median Graph

The most popular exact algorithm is called Multimatch [10], and was first presented by Münger and Bunke in 1995. This approach, as any exact Median Graph computation, suffers from a high computational complexity, and its application is very limited. The use of suboptimal methods is thus the unique feasible option to extend the use of the Median Graph to more realistic sets of graphs.

Approximate algorithms developed so far include a genetic based strategy [6,10] and one greedy-based algorithm [11]. Both solutions generally apply some kind of heuristics in order to reduce both the cost of the graph distance computation and the size of the search space. Finally, the most recent approach, which is based on the proposal by Riesen et al [12], consists on embedding the graphs into an auxiliary Euclidean space. Let us give a more thorough explanation of this last technique, since the algorithms that we have used in the experiments presented in this paper follow it.

3.1 Median Graph via Graph Embedding

The general embedding procedure is composed of three main steps, detailed in the following.

- **Step I: Graph Embedding in a Vector Space:** Each graph in the set S is embedded into an n-dimensional vector space. The vector representation p_i of a graph g_i of the set is obtained by computing its distance to all the graphs in the set. More precisely, the j-th coordinate of the vector corresponds to the distance to the j-th graph of the set.
- **Step II: Median Vector Computation:** This step consists in computing the median vector $\bar{p}$ of the points obtained in the first step. Although the Euclidean Median cannot be calculated in a straightforward way [13], an approximation, as good as desired, can be obtained by means of the Weiszfeld's algorithm [14]. It is an iterative procedure that converges to the solution.
- **Step III: Going Back to the Graph Domain:** The last step consists in going back to the graph domain converting the median vector into a graph $\tilde{g}$. This graph is taken as the Median Graph of the set. Different options on how to perform this last step have been proposed.

Linear Interpolation Procedure. In this algorithm from [15], once the median vector $\bar{p}$, is computed, the two closest points, p_1 and p_2 without lost of generality, are used to obtain the approximate median. The approximate generalized Median Graph, $\tilde{g}$, is then the weighted mean of g_1 and g_2, with $a = \frac{1}{2}d(g_1, g_2)$. We will refer to this algorithm as *linear embedding* (**MLE**).

Triangulation Procedure. In this case, the three closest points to $\bar{p}$ are selected for the approximated generalized Median Graph computation. This computation consists on generating an intermediate weighted mean using two of the three points followed by a second and definitive weighted mean which makes use of the third point and the previous weighted mean. The procedure, referred to as *triangulation embedding* (**MTE**), was proposed and explained in [15].

Recursive Procedure. A third option is to take into account all the points, this is, all the graphs in the set S in Step III. That is what the authors propose in [16]. We will refer to this algorithm as *recursive embedding* (**MRE**).

4 Computation of the Barycenter Graph

The algorithm that we propose to approximate the Barycenter Graph is based on the following geometrical property of the barycenter in Euclidean spaces.

Lemma 4. *Given a set* $P = \{p_1, p_2, \dots, p_m\}$ *of* m *points with* $p_i \in \mathbb{R}^n$ *for* $i = 1 \dots m$*, the barycenter*

$$Bar(P) = arg\ \min_{y \in \mathbb{R}^n} \sum_{i=1}^{m} ||p_i - y||^2,$$

of the set P *satisfies, for any* $1 \leq j \leq m$*,*

$$Bar(P) \ = \ \frac{1}{m} p_j + \frac{m-1}{m} Bar(P \setminus \{p_j\}). \tag{1}$$

As it is deduced from equation (1), $Bar(P)$ lies in the segment with ends p_j and $Bar(P \setminus \{p_j\})$ and

$$\|Bar(P \setminus \{p_j\}) - Bar(P)\| = (m-1)\|Bar(P) - p_j\|,$$

where $\| \cdot \|$ denotes the Euclidean distance. Therefore, in Euclidean spaces, the barycenter of m points can be recursively computed by subtracting a point in the set and computing the barycenter of the remaining ones. Then, the barycenter is easy to compute because it belongs to a segment with known ends and the distance to these ends is also known.

4.1 Algorithm

The procedure explained above can be easily adapted to the domain of graphs, since the last step corresponds to the computation of the weighted mean. The

Algorithm 1. Algorithm to approximate Barycenter Graph computation

```
input  : A set $S = \{g_1, \ldots, g_n\}$ of $n$ graphs
output: $\tilde{b}$ = Approximate Barycenter Graph of $S$
begin
    $B_2 = \mathrm{WM}(g_1, g_2, d(g_1, g_2)/2)$
    for $3 \leq m \leq n$ do
        $B_m = \mathrm{WM}(B_{m-1}, g_m, d(B_{m-1}, g_m)/m)$
    Return $\tilde{b} = B_n$.
end
```

resulting algorithm, **Algorithm 1**, may be considered an extension to the algorithm presented in [7]. The main contribution is that the graph edit distance, instead of the geometrically restricted distance function required in [7], can be used as the graph similarity measure of the graph domain.

The output of **Algorithm 1** is an approximation $\tilde{b} \approx \bar{b}$ to the barycenter graph. This inaccuracy is on the one hand due to the error $\epsilon(a)$, and a consequence of the suboptimal computation of distances and weighted means, which is unavoidable unless the number and size of the graphs of the set S is very small. On the other hand, it cannot be theoretically proved that the algorithm minimizes the SOSD. Nevertheless, the fact that our method gives results with small SOSD is supported by experimental results.

It is important to remark that there is no need to transform the graphs into vectors to apply our method. This means that the structural information of the graphs is preserved at every step in the process.

4.2 Different Sorting Schemes

In **Algorithm 1** the graphs are taken as they arrive, without any sorting. Then the question whether the ordering of the input plays a non-negligible part in the accuracy of the approximation arises. For this reason, we have developed and implemented two methods, the *Ascendent SOSD–based* sorting (**BSA**) and the *Descendent SOSD–based* sorting (**BSD**), to study the effect of the ordering.

In the BSA method the graphs of the input are ordered upwards, such that the first graph, g_1, is that with minimum SOSD: the set barycenter. In the BSD method, the ordering of the graphs is the inverse.

The method explained in Section 4.1, without preprocessing the data, will be referred to as *unordered* barycenter computation method (**BN**). We also compute the set barycenter (**SB**).

5 Comparison between Median and Barycenter Graphs

In this section we aim to compare the quality of the median and the Barycenter Graphs, as representatives of a set of graph. To do so, we compare experimental results on three real–data–based and five artificial datasets, some characteristics of which are displayed in Table 1. The LetterLOW, LetterHIGH, Molecules, Mutagenicity and Webpages datasets are from [17], where more information on them is available. The Synthetic datasets were created by the authors.

Table 1. Some dataset characteristics: size, number of classes (#c) and the average and maximum size of graphs

Database	**Size**	**# c**	$\emptyset\|g\|$	**max**$\|g\|$
LetterLOW	2,250	15	4.7	8
LetterHIGH	2,250	15	4.7	8
Molecules	2,000	2	15.7	95
Mutagenicity	4,337	2	30.3	417
Webpages	2,340	6	186.1	834
SyntheticSmall	2,000	10	10	13
SyntheticMedium	2,000	10	50	62
SyntheticLarge	2,000	10	100	122

The exact computation of both the generalized Median Graph and the generalized Barycenter Graph is unaffordable for these data. To carry on the experiments for this paper we have selected those suboptimal algorithms that can handle graphs with thousands of nodes. Table 2 shows the methods used.

Table 2. Methods to Approximate the Median and Barycenter Graphs and number of distances and weighted means that are computed for each of them, where n is the number of graphs in the given set S

Method	**Shortening**	**#distances**	**#WMs**
Medians			
Set Median	SM	$\mathcal{O}(n^2)$	0
Linear Interpolation Embedding	MLE	$\mathcal{O}(n^2)$	1
Triangulation Embedding	MTE	$\mathcal{O}(n^2)$	2
Recursive Embedding	MRE	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$
Barycenters			
Set Barycenter	SB	$\mathcal{O}(n^2)$	0
Unordered Barycenter Computation	BN	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Ascendent SOSD–Based Sorting	BSA	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$
Descendent SOSD–Based Sorting	BSD	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$

5.1 Algorithmical Considerations

Before showing the result of our experiments, we want to compare the different algorithms in terms of time complexity. Table 2 shows the number of distances and weighted means that need to be computed for each of the methods, where n is the number of graphs in the given set S.

The embedding procedure has been shown to be the only method for the Generalized Median Graph computation potentially applicable to real world problems, due to its lower computational demand [15]. Still, the embedding step (Step I) requires the computation of all the pairwise distances of the graphs in the given set S. That means that the number of distances computed is quadratic on the number n of graphs. In terms of computation time Step II is negligible, and Step III does not depend on n if MLE or MTE are used. The MRE procedure computes a linear, on n, number of weighted means.

In all the barycenter computation algorithms from Table 2, a linear number of weighted means must be computed. In the case of BSA and BSD, the computation of a quadratic number of distances is also needed. The number of distance computations that requires the unsorted algorithm BN, is linear on n, instead.

In this paper, we have chosen to follow [18] and [19] for the graph edit distance computation and [9] to compute the weighted mean. This makes the graph edit distance computation more time demanding than the weighted mean, and the BN method the fastest one.

It is important to remark, then, that BN may be an interesting choice from the computational point of view. At sight of conclusions drawn in [20], the loss of quality of the approximation in comparison with other barycenter computations is small when special robustness against outliers is not needed. Finally, let us note that the BN method is incremental, making it unnecessary to store all the information to be processed. The rest of the algorithms are not.

5.2 Stability

Some of the methods that we are considering, namely MRE, BSA and BSD compute $n - 1$ intermediate approximations, being the last one taken as the definitive one. In this section we study, experimentally, the evolution of the quality of the approximation along these $n - 1$ steps. Recall that the Median Graph aims to minimize the SOD while the Barycenter Graph approximates the graph with minimum SOSD. For this reason, SOD and SOSD will be our reference values.

In this experiments we compute the Median Graph of several graph sets for letter, molecule, mutagenicity and web databases. More precisely, we compute the median and the barycenter of sets of 50 and 100 randomly chosen graphs belonging to the same class, and we do so for all the classes in each database and using each of the methods we want to evaluate. Each of these experiments is repeated 10 times.

As an example, Figure 1 shows the evolution of the SOD and SOSD, correspondingly, for the different methods in the experiments carried out with the Webpages dataset with sets of 50 graphs. We want to remark that the methods to compute the barycenter show a convergent tendency, while the evolution of the recursive embedding method is more irregular. Similar result concerning other datasets are skipped due to space constraints.

5.3 Distance to Prototype

In this section we present a second experiment, devoted to comparing the median and the barycenter as representatives of a given set of graphs. To this end, we have performed experiments with LetterLOW, LetterHIGH and the three Synthetic databases. All this datasets have been created by distorting initial

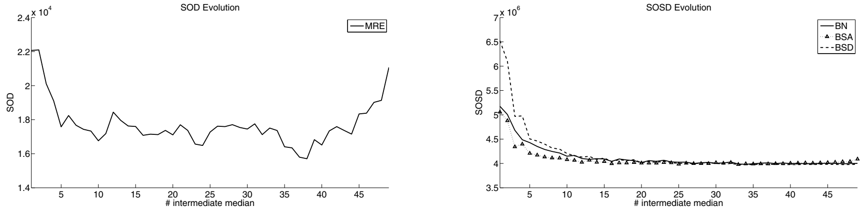


Fig. 1. Mean of the evolution of the SOD of the Median (left) and the SOSD of the Barycenter (right) for the **Web** dataset

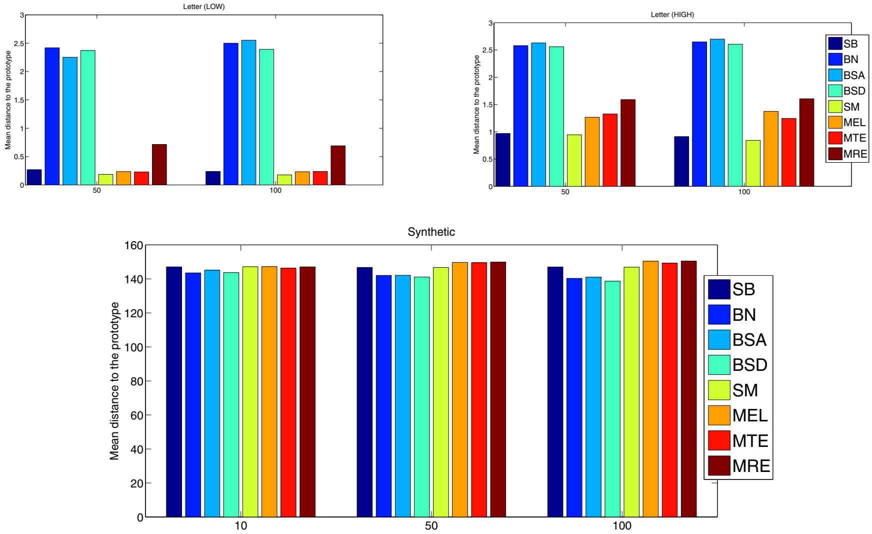


Fig. 2. Mean distance to prototype of several approximations to the barycenter and the median graphs, performed on letterLOW, letterHIGH, SyntheticSmall, SyntheticMedium and SyntheticLarge datasets

prototypes. This allows us to compare the medians and barycenters provided by the different methods, to the original prototype. Under the assumption that the best possible representative is the prototype itself, we consider that the smaller the distance to the prototype, the better the representative is.

For each of the datasets, we have computed the representative of sets of different sizes (50 and 100 for letters and 10, 50 and 100 for Synthetics) using each of the methods displayed in Table 2. In each database, 20 sets of graphs are considered for each class. Figure 2 shows the mean distance of the resulting approximation to the prototype, taken over all the classes and all the repetitions.

In the LetterLOW and LetterHIGH datasets we observe that the set median, followed by the set barycenter is the closest representative to the prototype. Recall that, by definition, the generalized Median Graph has lower or equal

SOD than the set median and similarly, the generalized barycenter has lower or equal SOSD than the set barycenter. This means that the set median and the set barycenter are expected to be worse representatives. In other words, they are the dummy approximations to beat.

We conclude that for the letter datasets, the approximated algorithms used for median and barycenter computation, although they have been experimentally validated [16,15], fail to give satisfactory results. Let us remark that these databases suffer from a high level of distortion, potentiated by the fact that the graphs have few nodes. The embedding technique for Median Graph computation gives better representatives than the barycenter techniques. We may conclude that the Median Graph shows a higher robustness against large distortion. That it behaves better in difficult datasets, in other words.

In the Synthetic databases, the set median and the set barycenter are outperformed by all the algorithms to compute the generalized barycenter. Three facts are to be underlined. First, that the Barycenter Graphs give representatives closer to the prototype than the Median Graphs computed via embedding. Secondly, that the BSD gives the closest representatives to the prototype in the three Synthetic datasets. Thirdly, that the BN method, which as we said before is faster to compute than the rest of the methods, gives similar results to the rest of the barycenter methods.

6 Conclusions

In the present paper we have compared two representatives of a set of graphs, the median graph and the barycenter graph. Since their exact computation is unaffordable, this comparison is carried out by means of several algorithms that provide approximate medians and barycenters.

By comparing these algorithms we have concluded that an approximation to the barycenter can be computed faster than an approximation to the median. Also, we have noted that the algorithms for barycenters show a high level of convergence in the process of computing intermediate solutions, the last of which is the definitive approximation.

Finally, we have designed an experiment to discuss whether, among the median and the barycenter graph, one is better than the other as a representative. We have observed that results are not uniform for different datasets, which makes us conclude that none of them can be said to be better than the other. Nevertheless, we remark that, for datasets for which the grade of distortion is not very high, barycenters give better representatives.

References

1. Wong, A., You, M.: Entropy and distance of random graphs with application to structural pattern recognition. IEEE TPAMI 7, 599–609 (1985)
2. Serratosa, F., Alquézar, R., Sanfeliu, A.: Synthesis of function-described graphs and clustering of attributed graphs. IJPRAI 16(6), 621–656 (2002)

3. Serratosa, F., Alquézar, R., Sanfeliu, A.: Function-described graphs for modelling objects represented by sets of attributed graphs. Pattern Recognition 36(3), 781–798 (2003)
4. Serratosa, F., Alquézar, R., Sanfeliu, A.: Estimating the joint probability distribution of random vertices and arcs by means of second-order random graphs. In: Caelli, T.M., Amin, A., Duin, R.P.W., Kamel, M.S., de Ridder, D. (eds.) SPR 2002 and SSPR 2002. LNCS, vol. 2396, pp. 252–262. Springer, Heidelberg (2002)
5. Cordella, L.P., Foggia, P., Sansone, C., Vento, M.: Learning structural shape descriptions from examples. Pattern Recognition Letters 23(12), 1427–1437 (2002)
6. Jiang, X., Münger, A., Bunke, H.: On median graphs: Properties, algorithms, and applications. IEEE Trans. Pattern Anal. Mach. Intell. 23(10), 1144–1151 (2001)
7. Jain, B., Obermayer, K.: On the sample mean of graphs. In: Proc. of IJCNN 2008, June 2008, pp. 993–1000 (2008)
8. Sanfeliu, A., Fu, K.: A distance measure between attributed relational graphs for pattern recognition. IEEE TSMC 13(3), 353–362 (1983)
9. Bunke, H., Günter, S.: Weighted mean of a pair of graphs. Computing 67(3), 209–224 (2001)
10. Münger, A.: Synthesis of prototype graphs from sample graphs. In: Diploma Thesis, University of Bern, in German (1998)
11. Hlaoui, A., Wang, S.: Median graph computation for graph clustering. Soft Comput. 10(1), 47–53 (2006)
12. Riesen, K., Neuhaus, M., Bunke, H.: Graph embedding in vector spaces by means of prototype selection. In: Escolano, F., Vento, M. (eds.) GbRPR. LNCS, vol. 4538, pp. 383–393. Springer, Heidelberg (2007)
13. Bajaj, C.: The algebraic degree of geometric optimization problems. Discrete Comput. Geom. 3(2), 177–191 (1988)
14. Weiszfeld, E.: Sur le point pour lequel la somme des distances de n points donnés est minimum. Tohoku Math. Journal (43), 355–386 (1937)
15. Ferrer, M., Valveny, E., Serratosa, F., Riesen, K., Bunke, H.: Generalized median graph computation by means of graph embedding in vector spaces. Pattern Recognition 43(4), 1642–1655 (2010)
16. Ferrer, M., Karatzas, D., Valveny, E., Bunke, H.: A recursive embedding approach to median graph computation. In: Torsello, A., Escolano, F., Brun, L. (eds.) GbRPR 2009. LNCS, vol. 5534, pp. 113–123. Springer, Heidelberg (2009)
17. Riesen, K., Bunke, H.: IAM graph database repository for graph based pattern recognition and machine learning. In: SSPR/SPR, pp. 287–297 (2008)
18. Neuhaus, M., Riesen, K., Bunke, H.: Fast suboptimal algorithms for the computation of graph edit distance. In: Yeung, D.-Y., Kwok, J.T., Fred, A., Roli, F., de Ridder, D. (eds.) SSPR 2006 and SPR 2006. LNCS, vol. 4109, pp. 163–172. Springer, Heidelberg (2006)
19. Riesen, K., Neuhaus, M., Bunke, H.: Bipartite graph matching for computing the edit distance of graphs. In: Escolano, F., Vento, M. (eds.) GbRPR. LNCS, vol. 4538, pp. 1–12. Springer, Heidelberg (2007)
20. Bardaji, I.: Graph representatives: Two different approaches based on the median and the barycenter graph. Master's thesis, UPC, Barcelona (2009)

Impact of Visual Information on Text and Content Based Image Retrieval

Christophe Moulin, Christine Largeron, and Mathias Géry

Université de Lyon, F-42023, Saint-Étienne, France
CNRS, UMR 5516, Laboratoire Hubert Curien, 42023, Saint-Étienne, France
Université de Saint-Étienne, Jean-Monnet, F-42023, Saint-Étienne, France
{christophe.moulin,christine.largeron,mathias.gery}@univ-st-etienne.fr

Abstract. Nowadays, multimedia documents composed of text and images are increasingly used, thanks to the Internet and the increasing capacity of data storage. It is more and more important to be able to retrieve needles in this huge haystack. In this paper, we present a multimedia document model which combines textual and visual information. Using a bag-of-words approach, it represents a textual and visual document using a vector for each modality. Given a multimedia query, our model combines scores obtained for each modality and returns a list of relevant retrieved documents. This paper aims at studying the influence of the weight given to the visual information relative to the textual information. Experiments on the multimedia ImageCLEF collection show that results can be improved by learning this weight parameter.

1 Introduction

In order to retrieve documents in multimedia collections, especially in the context of the Web, the development of methods and tools suitable to these data types is nowadays a challenging problem in Information Retrieval (IR). Most of the current IR systems handling multimedia documents can be classified into several categories, depending on their ability to exploit textual information, visual information, or a combination of both.

In the first category, namely *Text based Image Retrieval*, an image is indexed using only the textual information related to the image (file name, legend, text surrounding the image, etc.), without taking into account the image intrinsic features. This is the case, for example, of the main commercial search engines, and also of some systems specialized in images retrieval, such as Picsearch[1].

In the second category, namely *Content Based Image Retrieval* (CBIR), only the visual content of the image, represented by local color, shape or texture features, is used [1,2]. For example, QBIC, the IBM precursor system [3], proposes to retrieve images considering a query expressed using only those basic color, shape and texture features. The systems giving the best results are those handling a query image built by the user or an image example provided by the user

[1] Picsearch: http://www.picsearch.com

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 159–169, 2010.

("Search by image", e.g. QBIC or more recently the search engine TinEye[2]). So, some systems propose to the user to sketch the image sought ("Search by sketch", e.g. the Gazopa and Retrievr[3] search engines) while other propose to the user to arrange on a canvas the icons corresponding to concepts that have been previously identified in the image database. But one drawback of these systems is that users do not always have a reference image, and query languages based on visual features are not always very intuitive.

Finally, the last category deals with systems handling textual and visual features simultaneously. For example, the PicHunter system [4] aims at predicting users' goal given their actions while the Picitup system[4] proposes to define a textual query and then to filter results using visual elements (a picture, a category, a color, a shape, etc.). Recently, these approaches aiming at combining textual and visual information have been encouraging [5,6], but they have to fill the semantic gap between the objects and their visual representation [1]. A possible research direction deals with using visual ontology [7]; another one, proposed recently by Tollari, aims at associating keywords and visual information [8].

These previous works led us to propose a first approach which combines textual and visual information. Starting from a first set of documents returned for a given textual query, our system enriches the query, adding some visual terms to the original textual query in an automatic way or a semi-automatic way (i.e. asking the user for feedback on the first returned documents) [9].

Our preliminary experiments have shown the potential of combining visual and textual information. The first aim of the present work is to study how to estimate the weight of the visual information relative to the textual information. We propose to learn automatically this weight, using an IR collection as a learning set. The second aim is to check if the optimal weight accorded to each information type varies by the kind of queries, and if estimating a specific weight for each query can significantly improve the results. Indeed, the visual information is less important for concepts like e.g. "animal" or "vehicle", because these concepts can be described by very different visual features.

The next section describes the document model we proposed, combining text and images, then we present some experiments on an IR task using the ImageCLEF collection in section 3; we present the results in section 4.

2 Visual and Textual Document Model

2.1 General Framework

The figure 1 presents the global architecture of our multi-modal IR model. The first component aims at indexing the documents D and the queries Q, both composed by textual and visual information. The textual content, as well as the visual one, is represented by a bag-of-words. The second component estimates,

[2] TinEye: http://www.tineye.com/

[3] Gazopa: http://www.gazopa.com/, Retrievr: http://labs.systemone.at/retrievr/

[4] http://www.picitup.com/picitup

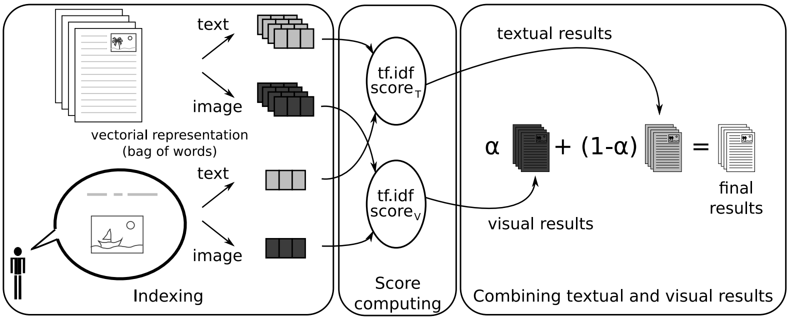


Fig. 1. Multi-modal IR model

given a query, a score for each document and for each modality (textual and visual). Finally, the last component combines linearly the score obtained for each modality, in order to retrieve the most relevant documents given a query.

2.2 Textual Document Model

Given a collection of documents D and $T = \{t_1, ..., t_j, ..., t_{|T|}\}$ the set of words occurring in the documents, each document $d_i \in D$ is represented as a vector of weights $w_{i,j}$ (vector space model [10]): $\boldsymbol{d_i} = (w_{i,1}, ..., w_{i,j}, ..., w_{i,|T|})$, with $w_{i,j}$, the weight of the term t_j in the document d_i, computed by a *tf.idf* formula ($w_{i,j} = tf_{i,j} * idf_j$). $w_{i,j}$ is high when the term t_j is frequent in the document d_i but rare in the others.

$tf_{i,j}$ is the *term frequency* that characterizes the representativeness of the term t_j in the document d_i. We use the variant of the BM25 weighting function defined in Okapi [11] and implemented by the Lemur system [12]:

$$tf_{i,j} = \frac{k_1 \times t_{i,j}}{t_{i,j} + k_1 \times (1 - b + b * \frac{|d_i|}{d_{avg}})}$$

where $t_{i,j}$ is the number of occurrences of the term t_j in the document d_i, $|d_i|$ the size of the document d_i, d_{avg} the average size of all documents and k_1 and b two constants.

idf_j is the *inverse document frequency* which estimates the importance of the term t_j over the corpus of documents. We use also the BM25 variant implemented by Lemur:

$$idf_j = \log \frac{|D| - df_j + 0.5}{df_j + 0.5}$$

where $|D|$ is the size of the corpus and df_j the number of documents where the term t_j occurs at least one time.

If we consider a query q_k in the same way (i.e. as a short document), we can also represent it as a vector of weights. A score is then computed between the query q_k and a document d_i:

$$score_T(q_k, d_i) = \sum_{t_j \in q_k} tf_{i,j} idf_j * tf_{k,j} idf_j$$

2.3 Visual Document Model

In order to combine the visual and the textual information, we also represent images as vectors of weights. It is possible to use the *tf.idf* formula in the same way, provided we are able to extract visual words from images. It requires a visual vocabulary $V = \{v_1, ..., v_j, ..., v_{|V|}\}$, which is built in two steps using a bag of words approach [13]. In the first step, each image of the collection D is segmented into a regular grid of 16×16 cells, with at least 8×8 pixels by cell. Then, each cell is described by the visual descriptor SIFT (*Scale-Invariant Feature Transform*) based on histograms of gradient orientation [14]. SIFT converts each cell into 128-dimensional vector in such a way that each image is a collection of vectors. We have evaluated other visual descriptors, like *meanstd* [9], but only the best results, provided by SIFT, are presented in this article.

In the second step, the visual words are built by performing a k-means clustering over the visual vectors. The words of the visual vocabulary V are then defined as the centers of the clusters and the size of the visual vocabulary corresponds to the number of clusters.

This bag of visual words is analogous to the bag of textual words inasmuch as an image can then be represented by an histogram of visual words. Indeed, an image, belonging to a document or a query, can be segmented into cells described by SIFT vectors and, each vector can be assigned to the nearest cluster (i.e. visual word) according to the Euclidean distance. This way, it is possible to count the number $v_{i,j}$ of occurrences of the visual word v_j in the image, in other words the number of cells $v_{i,j}$ assigned to the cluster with the center v_j. Like in the textual model, an image is represented by a vector where the weights for the visual words are given by the *tf.idf* formula in which $t_{i,j}$ is replaced by $v_{i,j}$ and t_j by v_j.

Finally, a visual score $score_V(q_k, d_i)$ is then computed between a document d_i and a query q_k by:

$$score_V(q_k, d_i) = \sum_{v_j \in q_k} tf_{i,j}idf_j * tf_{k,j}idf_j$$

2.4 Combining Textual and Visual Informations

The global score for a document d_i given a query q_k is computed, combining linearly the scores computed for each modality:

$$score(q_k, d_i) = \alpha \times score_V(q_k, d_i) + (1 - \alpha) \times score_T(q_k, d_i)$$

where α is a parameter allowing to give more or less importance to the visual information relative to the textual information.

3 Experiments

In order to experiment our model, we have used the IR collection ImageCLEF [15]. Our aim is to evaluate the impact of visual information on multimedia IR: this requires to study the influence of the fusion parameter α.

3.1 ImageCLEF: IR Collection

The ImageCLEF collection is composed by 151,519 XML documents extracted from Wikipedia, composed by one image (photos, drawings or painting) and a short text, which describes the image but which can also give some information related to the owner or to the copyright.

Each year, a different set of queries is delivered: in ImageCLEF 2008, used as a training collection, there are 75 queries. 42 queries contain both a textual part (a few words) and a visual part. The 33 others queries are provided with only a textual part. In order to have a visual information obtained in a similar way for all queries, the two first images ranked by a preliminary textual querying step have been used as a visual query part for all the 75 queries. In ImageCLEF 2009, used as a testing collection, there are 45 queries, containing both a textual part and a visual part (1.84 images per query).

3.2 Evaluation Measures

Several evaluation measures have been used, such as MAP, $P10$ and $iP[0.1]$. Let $Q = \{q_1, ..., q_k, ..., q_{|Q|}\}$ be the set of queries and $D_k = \{d_{k,1}, ..., d_{k,i}, ..., d_{k,|D_k|}\}$ the set of relevant documents given q_k. The N_k retrieved documents for the query q_k is a list of documents ranked according to their score. In ImageCLEF competition, N_k equals to 1000. The rank r corresponds to the r^{th} document ranked by the system. Precision $P_k(N)$ is defined as the number of relevant retrieved documents given q_k divided by the N retrieved documents. Recall $R_k(N)$ is defined as the number of relevant retrieved documents divided by the number of relevant documents. AP_k is the average precision for q_k.

$$P_k(N) = \frac{\sum_{r=1}^{N} \mathrm{rel}_k(r)}{N} \quad R_k(N) = \frac{\sum_{r=1}^{N} \mathrm{rel}_k(r)}{|D_k|} \quad AP_k = \frac{\sum_{r=1}^{N_k} (P_k(r) \times \mathrm{rel}_k(r))}{|D_k|}$$

where $\mathrm{rel}_k(r)$ is a binary function which equals 1 if the r^{th} document is relevant for the query q_k and 0 otherwise.

Three evaluation measures have been used to evaluate our model. The first one (MAP: Mean Average Precision) corresponds to the average for all queries of the average precision AP_k. The second one ($P10$) is the precision at 10^{th} rank. The last one ($iP[0.1]$) is the interpolated precision at 10% recall.

$$MAP = \frac{\sum_{k=1}^{|Q|} AP_k}{|Q|} \quad P10 = \frac{\sum_{k=1}^{|Q|} P_k(10)}{|Q|} \quad iP[0.1] = \frac{\sum_{k=1}^{|Q|} iP_k[0.1]}{|Q|}$$

with:

$$iP_k[0.1] = \begin{cases} \max_{1 \le r \le N_k} (P_k(r) | R_k(r) \ge 0.1) & \text{if } 0.1 \le R_k(N_k) \\ 0 & \text{otherwise} \end{cases}$$

3.3 Experimental Protocol

Many experiments were conducted in order to evaluate the interest of considering visual information on an IR task, and to study the α's influence.

Learning the $\boldsymbol{\alpha}$ parameter: firstly, queries from the ImageCLEF 2008 (resp. ImageCLEF 2009) collection are used as training set in order to calculate α_g^{2008} (resp. α_g^{2009}), the α value that globally optimize results on ImageCLEF 2008 (resp. ImageCLEF 2009). The optimal value of α correspond to the value of α that gives the best results for a given criterion, such as the MAP measure, obtained using a stepped search on the training set. We have used the MAP measure which is the main one used in the ImageCLEF competition. The learned α_g^{2008} value has been used by our system to process all the queries from the ImageCLEF 2009 collection. Our first question concerns the possibility of learning the parameter of the model on a set of queries and using it on a new set of queries: is it possible to estimate the optimized value α_g^{2009} using the ImageCLEF 2008 collection? The comparison of α_g^{2008} and α_g^{2009} will allow to conclude on the effectiveness of learning α.

Robustness of $\boldsymbol{\alpha}$ with regard to evaluation measures: the second aim is to determine the importance of visual information relative to the textual information, depending on the use case: 1) recall-oriented (exhaustive search), retrieving a lot of documents more or less relevant, 2) precision-oriented (focused search), retrieving a smaller set of documents mostly relevant. For this purpose, we have studied the parameter α_g regarding several evaluation measures: in the first hand MAP, which focus on recall, and in the other hand $P10$ and $iP[0.1]$, which focus on precision.

Optimizing $\boldsymbol{\alpha}$ parameter depending on the query: thirdly, we study the behavior of our model depending on the query type. Some queries seem to mainly depend on the textual information, such as *"people with dogs", "street musician"*, while others require more visual information, such as *"red fruit", "real rainbow"*. Studying how the performance of the system change depending on the kind of query is thus interesting. This local approach aims at calculating α_k, the α value optimized given a query q_k. The mean and the standard deviation of α_k will let us conclude on the variation of the α parameter depending on the query and on the interest of methods that aim at estimating the optimal α_k value for a new query. We will also study the optimization of α depending on the evaluation measures and thus, we will calculate the α_k optimized for the MAP, $P10$ and $iP[0.1]$ measures.

Global vs local approach: in the global approach, we study the variation of the α parameter in order to optimize the evaluation measure MAP_α (resp. $P10_\alpha$, $iP[0.1]_\alpha$). Let α_g be the optimal global value of the α paramater that maximizes MAP_α (resp. $P10_\alpha$, $iP[0.1]_\alpha$) on the training set:

$$\alpha_g = \alpha | MAP_{\alpha_g} = max\{MAP_\alpha, \alpha \in [0,1]\}$$

α_g is then used for all queries of the test set. During the ImageCLEF 2009 competition, α_g was obtained using all the queries of the 2008 collection and it was then used for processing the queries of the 2009 collection.

The local approach that uses a specific α per query should be the best solution. However, in practice, this local approach can not be performed since a training set is not available for a new query. Nevertheless, in order to compare our global approach with this local approach, we have searched the α_k value that optimizes the AP_k, $P_k(10)$ and $iP_k[0.1]$ measures for each query q_k using the test set. Then the MAP_{α_l} measure, corresponding to the average of the optimized average precision AP_k, is defined by:

$$MAP_{\alpha_l} = \frac{\sum_{k=1}^{|Q|} AP_k|\alpha = \alpha_k}{|Q|}$$

3.4 Setting Up of Our Model

The lemur software has been used with the default parameters as defined in [12]. The k_1 parameter of BM25 is set to 1. As $|d_k|$ and d_{avg} are not defined for a query q_k, b is set to 0 for the $tf_{k,j}$ computation. When the $tf_{i,j}$ is computed for a document d_i and a term t_j, this paramater b is set to 0.5. Moreover, stop-words have not been removed and a Porter stemming algorithm have been applied. The number of visual words, corresponding to the parameter k of the k-means, has been empirically set to 10,000.

4 Results

4.1 Learning Parameter α

MAP is a global measure corresponding to the average of the average precision for each query. This is the official ImageCLEF measure. Table 1 summarizes the results obtained, depending on which modality is used (text, visual, text + visual), and also on the optimizing method that is used. According to the MAP measure, the visual information leads to poor results ($MAP = 0.0085$) compared to those obtained using only the text ($MAP = 0.1667$).

However, figure 2 shows that giving more importance to the visual information significantly improves the results obtained only with text, especially with α close to 0.1. Nevertheless, giving too much importance to α (i.e. $\alpha > 0.1$) reduces the results quality. The α values are not normalized: thus it is difficult to interpret them directly, and only the improvement of IR results should allow to evaluate the interest of integrating visual information.

The parameter α_g^{2008} computed with the 2008 learning collection improves the results obtained using only the text on 2009 collection (+14.16%, MAP

Table 1. Results on the ImageCLEF 2009 collection (MAP measure)

Run	MAP	Gain / text only
Text only	0.1667	
Visual only	0.0085	-94.90%
Text+Visual (α_g^{2008})	0.1903	+14.16%
Text+Visual (α_g^{2009})	0.1905	+14.28%

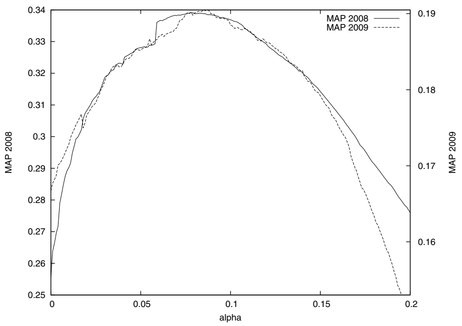


Fig. 2. MAP measure vs. α (2008 and 2009)

$= 0.1903$). This result is very interesting, particularly when it is compared to the optimal result ($MAP = 0.1905$) obtained using the α_g^{2009} value optimized on the 2009 collection itself. The MAP curves according to α, which look similar, and the values of $\alpha_g^{2008} = 0.084$ and $\alpha_g^{2009} = 0.085$, show a good robustness of the α_g parameter while changing collection (w.r.t. the MAP). Thus we think that learning α_g is possible.

4.2 Stability of Parameter α_g regarding the Evaluation Measure

Regarding more specific evaluation measures, as for example the precision oriented measures $P10$ and $iP[0.1]$, the α parameter seems less stable than regarding the MAP measure, especially on the 2009 collection, as shown by figure 3 (note that $P10$ and $iP[0.1]$ are averages, while MAP is an average of averages).

For these measures, the value of the α parameter learned on 2008 ($P10$: $\alpha_g^{2008} = 0.140$; $iP[0.1]$: $\alpha_g^{2008} = 0.108$) is quite different than the optimal α value for 2009 ($P10$: $\alpha_g^{2009} = 0.095$; $iP[0.1]$: $\alpha_g^{2009} = 0.078$). Nevertheless, the weighting of the visual information through the parameter α_g^{2008}, even if relatively different than the optimal value α_g^{2009}, still allows to significantly improve the results

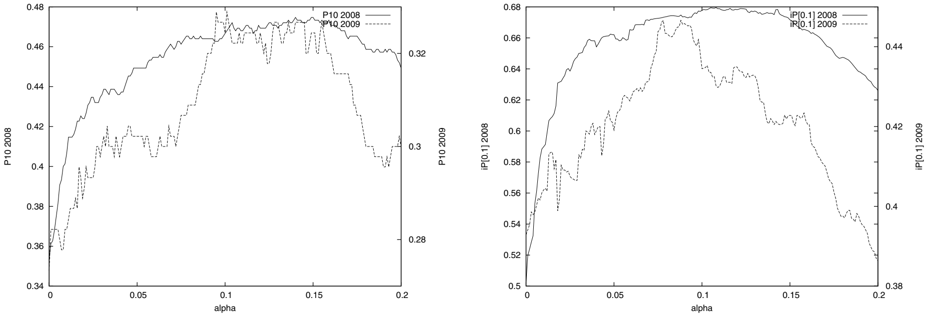


Fig. 3. $P10$ and $iP[0.1]$ measures vs. α (2008 and 2009)

Table 2. Results on the collection ImageCLEF 2009 ($P10$ and $iP[0.1]$ measures)

Run	$P10$	Gain / text only	$iP[0.1]$	Gain / text only
Text only	0.2733		0.3929	
Visual only	0.0178	-93.49%	0.0160	-95.93%
Text+visual (α_g^{2008})	0.3267	+19.54%	0.4302	+9.49%
Text+visual (α_g^{2009})	0.3289	+20.34%	0.4466	+13.67%

regarding $P10$ as well as $iP[0.1]$, as shown by table 2. We observe an improvement of 19.54% regarding $P10$, and of 9.49% regarding $iP[0.1]$.

4.3 Global Approach vs. Local Approach: Optimizing α w.r.t. a Query

The local approach, i.e. using a specific α_k parameter for each query q_k, is more challenging than the global approach, because it needs to compute a priori the value of α_k for each new query; this is an open problem. However, this approach would allow to dramatically improve the results: the potential gain is +29.99% (resp. +52.87%, +39.14%) reagrding the MAP measure (resp. $P10$, $iP[0.1]$), as shown by table 3. But implementing this local approach seems very difficult as it exists an important disparity of α_k regarding to the queries, as shown by μ_{α_l} (mean of α_k) and σ_{α_l} (standard deviation) observed for the 3 evaluation measures.

Table 3. Optimizing α_k for each query

	Run		Gain / text only	μ_{α_l}	σ_{α_l}
MAP	Text only	0.1667			
	Text+visual (α_l)	0.2167	+29.99%	0.080	0.063
$P10$	Text only	0.2733			
	Text+visual (α_l)	0.4178	+52.87%	0.055	0.058
$iP[0.1]$	Text only	0.3929			
	Text+visual (α_l)	0.5467	+39.14%	0.083	0.072

5 Conclusion and Future Work

In this paper, we have presented a multimedia IR model based on a bag-of-words approach. Our model combines linearly textual and visual information of multimedia documents. It allows to weight the visual information relative to the textual information using a parameter α.

Our experiments show that it is possible to learn a α_g^{2008} value for this parameter (using the ImageCLEF 2008 collection as a learning collection) and then to use it successfully on the ImageCLEF 2009 collection. This value sometimes differs compared to the optimal value α_g^{2009} (computed on the collection ImageCLEF 2009) regarding $P10$ and $iP[0.1]$, but remains relatively stable regarding MAP. However it allows to significantly improve the results regarding MAP as well as $P10$ and $iP[0.1]$.

According to our results, using a specific α_k for each query seems to be an interesting idea. In order to learn this parameter, a first approach could be to classify the queries: visual, textual and mixed queries. Maybe it is possible for this purpose to use the length of the textual queries, which seems to be related to the queries' class. Another direction could be to analyze some visual words extracted from the first set of textual results given the query, hypothesizing that they carry some visual information about the query. Their distribution should allow to estimate a specific α_k for each query.

References

1. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(12), 1349–1380 (2000)
2. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval: State of the art and challenges. ACM Transactions on Multimedia Computing, Communications, and Applications 2(1), 1–19 (2006)
3. Flickner, M., Sawhney, H.S., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.: Query by image and video content: The QBIC system. IEEE Computer 28(9), 23–32 (1995)
4. Cox, I.J., Miller, M.L., Minka, T.P., Papathomas, T.V., Yianilos, P.N.: The bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments. IEEE Transactions on Image Processing 9(1), 20–37 (2000)
5. Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D.M., Jordan, M.I.: Matching words and pictures. The Journal of Machine Learning Research 3, 1107–1135 (2003)
6. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys 40(2) (2008)
7. Snoek, C.G.M., Worring, M., Gemert, J.C.V., Mark Geusebroek, J., Smeulders, A.W.M.: The challenge problem for automated detection of 101 semantic concepts in multimedia. In: ACM Conference on Multimedia, pp. 421–430 (2006)
8. Tollari, S., Detyniecki, M., Marsala, C., Fakeri-Tabrizi, A., Amini, M.R., Gallinari, P.: Exploiting visual concepts to improve text-based image retrieval. In: European Conference on Information Retrieval, ECIR (2009)
9. Moulin, C., Barat, C., Géry, M., Ducottet, C., Largeron, C.: UJM at ImageCLEFwiki 2008. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) Evaluating Systems for Multilingual and Multimodal Information Access. LNCS, vol. 5706, pp. 779–786. Springer, Heidelberg (2009)

10. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communations of the ACM 18(11), 613–620 (1975)
11. Robertson, S.E., Walker, S., Hancock-Beaulieu, M., Gull, A., Lau, M.: Okapi at trec-3. In: Text REtrieval Conference, pp. 21–30 (1994)
12. Zhai, C.: Notes on the lemur TFIDF model. Technical report, Carnegie Mellon University (2001)
13. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV 2004 workshop on Statistical Learning in Computer Vision, pp. 59–74 (2004)
14. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
15. Tsikrika, T., Kludas, J.: Overview of the wikipediaMM task at ImageCLEF 2009. In: 10th Workshop of the Cross-Language Evaluation Forum, Corfu, Greece (2009)

Automatic Traffic Monitoring from Satellite Images Using Artificial Immune System

Mehrad Eslami[1] and Karim Faez[2]

[1] Computer Engineering Department, Azad University of Qazvin, Qazvin, Iran
Mehrad.Eslami@gmail.com
[2] Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran
kfaez@aut.ac.ir

Abstract. Automatic and intelligence Road traffic monitoring is a new research issue for high resolution satellite imagery application in transportation. One of the results of this research was to control the traffic jam in roads and to recognize the traffic density quickly and accurately. This article presents a new approach for recognizing the vehicle and the road in satellite high-resolution images in non-urban areas. For road recognition, they used feature extraction and image processing techniques like Hough transform, Gradient, and thresholding operation and they presented an artificial immune approach to extract vehicle targets from high resolution panchromatic satellite imagery. The average of results is about 94 percent and it shows that the used procedure has the suitable efficiency.

Keywords: Road Extraction, vehicle Detection, Hough transform, Artificial Immune System, Intelligence Traffic monitoring, Satellite images.

1 Introduction

With the growth of urban traffic and the necessity to control it, the attention has been paid to intelligent traffic control systems. Nowadays urban traffic is controlled by the cameras which are installed in highways located at a long distance from each other. With the development of technology, this procedure seems so slow and deficient. According to the satellite images considered recently, the existence of an intelligent system for road and vehicle recognition to control road's traffic will be so useful.

Hence, area-wide images of the entire road network are required to complement these selectively acquired data. Since the launch of new optical satellite systems like IKONOS and NAVTEQ, this kind of imagery is available with 0.6-1.0 meter resolution. Vehicles can be observed clearly on these high resolution satellite images. Thus new applications like vehicle detection and traffic monitoring are raising up.

Traffic, road and vehicle recognition in satellite images with very high resolution are very new discussions in machine vision science which makes many projects and other operations depend on it. Main factors having the most influence on the topic is the number of different objects in the image, the amount of their interconnections and the features that can distinguish them from other objects.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 170–179, 2010.

This paper proposes the technique using the artificial immune network concept to extract the vehicle targets and using Hough transform and parallel lines detection to extract roads and then recognize traffic in space imagery. First we detect road and its boundaries, then we start to detect vehicle in it, because after edge detection, only the objects within the boundaries of that road are processed. To do this, extraction of road and vehicle features is necessary. One of the most useful features of the road is that the road appeared in satellite images will usually be referred to as a direct confinement with a different color [7-8]; so the linear feature can be a proper one to detect road confinement. The other feature of roads is the lines drawn on them, these white lines which exist on almost all roads and help a lot in road detection. The lines close to roadsides and in the middle count as one of the proper features. The other feature used is road's color while in high resolution satellite images, distinguishes completely the roads from the roadside. With thresholding, the roads can be distinguished from background so that other operations can be performed on the image. This colorful threshold can be extracted by averaging images existing in the data. After using road's features to detect it, the vehicle will be detected in the roads. After road detection, vehicle detection is easier, because only the objects within the road limits are to be studied.

We used, the technique using the artificial immune network concept to extract the vehicles. The immune system is one of the highly evolved biological information processing systems and is capable to learn and memorize. Many kinds of immune systems have been studied by mathematical methods. In recent years, applications of artificial immune system have been proposed in many engineering problems [9-10]. In this study, we attempt to use them for target recognition. Observed targets are regarded as foreign antigens, and a template is regarded as an antibody. Complementary template matching is considered to be exactly the same as the binding of the Paratope and Epitope. The algorithm implements morphology operations on images to enhance vehicle features. Some of sub-images in the processed images are selected as the vehicle and non-vehicle training samples for antibody learning. The learned template antibodies are tested on real road segments.

2 Correlated Issues

Since satellite images with very high resolution include lots of information and elaborations of recognition process, the choice of appropriate features for recognition is one of the main operations in processing and analyzing satellite images.

The studied satellite images with very high resolution in this paper are caught by xerographic satellites. These images have high resolution which makes it easier to extract their features with high accuracy.

These images are fully colored taken in the same distance from Earth. The distance between cameras and the Earth is very important in taking satellite images, because some of the features which have been used have direct relationship with distance and size dimensions, and if the distance changes noticeably, we will need a new features measurement requiring standard modules.

Here it is assumed that the images are taken from a specific distance and resolution. Processing was done on color images in RGB, HSV. It should be considered that these images are taken of the non-urban roads with low probability of crossroads.

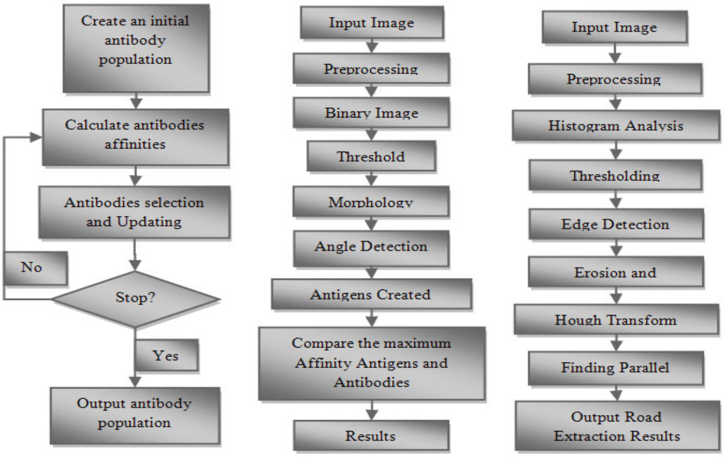


Fig. 1. Left: Antibodies learning flowchart. Middle: Vehicle Detection flowchart. Right: Road Detection Flowchart.

3 Road Detection

In this section, the represented method for road detection is implemented on satellite images (Figure 1). To do this, first we applied proper filters of image processing on the image, so that the image's edges will be clearer. One of these filters is sharpening filter which increases the image clearness and makes the image's edges more vivid.

Detection is applied on the image. Road's color is distinguished from other parts of the image with the help of a filter and through thresholding. First, the thresholding is performed to discriminate road from other parts of the image. This operation is done on the base of color difference between the road and other parts of the image.

This thresholding can be measured by the histogram of the road's satellite image. According to the histogram in Figure (2), it can be seen that, this histogram is divided into many areas. The areas which show roads in the image, is ranges 5 to 9 of horizontal axis of histogram which assign the greater amount of image color to itself. According to the histogram of satellite image and analysis of different images values red, green and blue which have the most similarities to the road's background color.

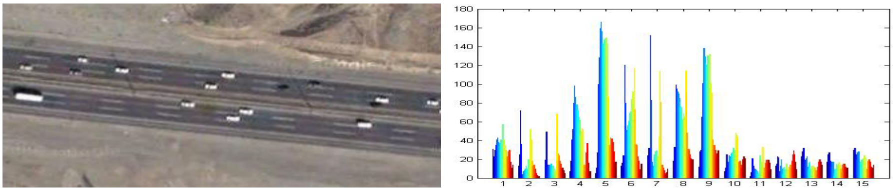


Fig. 2. Left: Satellite image of Karaj-Qazvin Highway. Right: Histogram of Satellite image of a Karaj-Qazvin Highway.

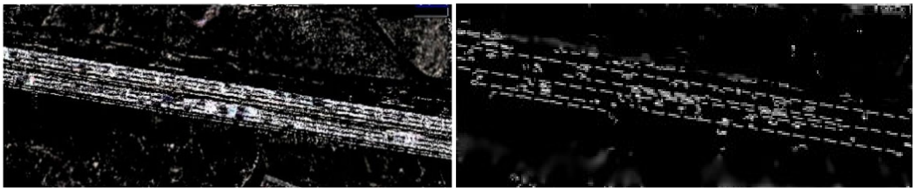


Fig. 3. Left: Road image after thresholding. Right: Road image after canny edge detection operation.

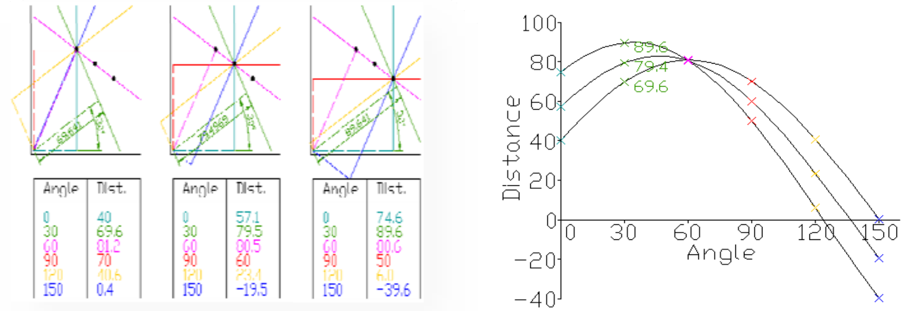
Now the main operation, the road detection, will be done, as it is shown in figures (3) (Road's color-linear of the white lines in sides and the middle of the road).

As it is shown in Figure (3), thresholding has been done, and the spots where the probability of the existence of road is less are in black. So other operations would be more functional. As you can see, there are spots, which are not road, but they are not black and it is because their color is the same as the road and that the spots must be consequently corrected. After thresholding, edge detection will function with better results. For edge detection Canny method will be used. This method has a better efficiency than other methods used in road detection. The result is shown in figure (5).

In digital images where the spots have been shown along with edges, there is a difference in color. So sharpening operations with the use of a filter will increase the color difference in edges and the clearness of the image; also, next operations such as edge detection will be performed with much proper attention. Hough transform is used after edge detection and thresholding in the last phase.

If the point (X_i,Y_i) and equation of a line be like Y_i=aX_i+b there are lots of lines coming from (X_i, Y_i) that any of them for the values of a, b make the equation Y_i=aX_i+b. we can recognize with Hough transform the spot that can be in line equation and make a line in the image [13-14, 17].

With the use of Figure (4), all the spots which are located on one line can be obtained. The spot where all curves meet shows a joint line on the specific spots. The spots which are located in parts of the chart that have same angles are parallel lines in the image that show road boundaries.



Angle	Dist.
0	40
30	69.6
60	81.2
90	70
120	40.6
150	0.4

Angle	Dist.
0	57.1
30	79.5
60	80.5
90	60
120	23.4
150	-19.5

Angle	Dist.
0	74.6
30	89.6
60	80.6
90	50
120	6.0
150	-39.6

Fig. 4. Left: several lines that crossed a point. Right: the spot where all curves meet make a joint line for all points.

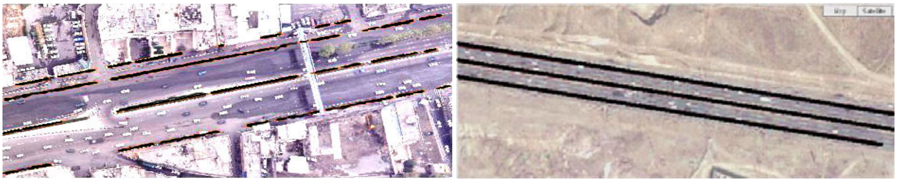


Fig. 5. Road and Highway after detection operation

Hough transform is done with a change of its parameters to improve the efficiency to detect road boundaries, and then vehicle detection in roads will be continued.

4 Vehicle and Traffic Detection

4.1 Definition of Immunological Terms

In this section, the immunological terms are defined in the following manner:

- *Antigen*: Vehicle targets.
- *Antibody*: Vehicle template images extracted from processed images by the morphology transform.

The used morphology transform is to enhance vehicle features. It is defined by

$$G(f) = f \oplus g - f \tag{1}$$

Where g is a structuring element, f is a gray scale image, f ⊕ g means dilate operation, i.e. Dilation:

$$(f \oplus g)\,(x) = max\{\, f(z)\; g^{*}(z) : \; z \in D\; [g^{*}\,]\} \tag{2}$$

Where $g_x(z) = g(z - x)$, $g^*(z) = -g(-z)$ and D[g] is the domain of g.

Figure (6), shows an original image and its morphology processing result. It can be clearly seen that all vehicle bodies or contours are enhanced. These enhanced features can be used to discriminate vehicle targets and non-vehicle targets. Figures (6) and (7) show some antibody examples collected from the morphology processed image, and each example image has same size.

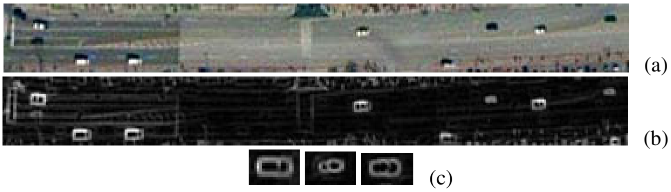


Fig. 6. (a) An original image. (b) The morphology preprocessing result. (c) Antibody examples.

- *Affinity:* Matching index. It is inspired from image correlation concept. It is defined by

$$R = \frac{\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}(w(x,y)-\bar{w})(f(x,y)-\bar{f})}{\sqrt{\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}(w(x,y)-\bar{w})^2}\sqrt{\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}(f(x,y)-\bar{f})^2}} \tag{3}$$

$w(x, y)$ is the template antibody image of size $K. L$ *and* $f(x, y)$ is the antigen image of size $K \cdot L$, W is the average intensity value of the pixels in template antibody image w, f is the average intensity value of the pixels in template antigen image f.

The greater the value of R has the higher the antibody's affinity.

4.2 Antibody Learning

For antibody learning (Figure 1), we setup an image database which includes vehicle samples and non-vehicle samples. In the database, all samples are collected from morphology processed images using same sampling window (Figure 7). To compare Antibodies with Antigens, chosen samples of both of them should be in the same direction. It should be considered that the movement orientation of the vehicles in the road, is the same as road's orientation, which was obtained in the previous section from the equation Y=αX+β. With rotation the vehicles as much as it is desire, all the extracted samples from the roads, will have same orientation. We randomly select N vehicle samples from the database as the initial antibody population, the rest samples are regarded as training sets. According to the immune network theory, antibodies interact with each other and with the environment (antigens). The interaction property leads to the establishment of a network. When an antibody recognizes an epitope or an idiotope, it can respond either positively or negatively to this recognition signal.

A positive response would result in antibody activation, antibody proliferation and antibody secretion, while a negative response would lead to tolerance and suppression.

According to these antibody properties, we develop an immune network for vehicle detection. A set of rules are proposed for antibody selection and updating in the immune network. These rules are as follows.

Rule 1. Eliminate the antibody if the maximum affinity of the antibody to vehicle samples is under the threshold (<0.6).

Rule 2. Eliminate the antibody that has high similarity over the threshold (>0.9) to other antibodies.

Rule 3. Eliminate the antibody if the affinity of the antibody to any non-vehicle sample is over the threshold (>0.6).

Rule 4. Add a vehicle sample from training sets into the antibody population as a new antibody if the affinity of the vehicle sample to any antibody is under the threshold (<0.6).

Based on above rules, the antibody learning procedure in the immune network is described as follows:

Step 1: Randomly select N vehicle samples from the database as the initial antibody population.

Step 2: Evaluate the affinity of each antibody in the population with Eq. (3).
Step 3: Eliminate the antibody according to Rule 1-3.
Step 4: Update antibody population according to Rule 4.
Step 5: Repeat Steps 2–4 until none of antibodies is eliminated and none of new antibodies is added.
Step 6: Save final antibody population for vehicle detection use.

Fig.3 shows the flowchart for antibody learning.

4.3 Strategy of Detection

After learning the antibody population, the learned antibody population can be used to detect vehicles in the imagery (Figure 1).

Firstly, according to Eq. (1), implement morphology transform on the original image (Figure 7). Secondly, calculate the maximum affinity to all template antibodies at each pixel point (i, j) by Eq. (3). Thirdly, compare the maximum affinity value R at every point with the given threshold. If the R is greater than the threshold, the point belongs to a vehicle target and is set as 255. Otherwise, it belongs to a non-vehicle target and is set as 0. Finally, a post processing based on morphology dilation and erode operations is employed to merge neighborhood vehicle target pixels and locate the center of a vehicle [21].

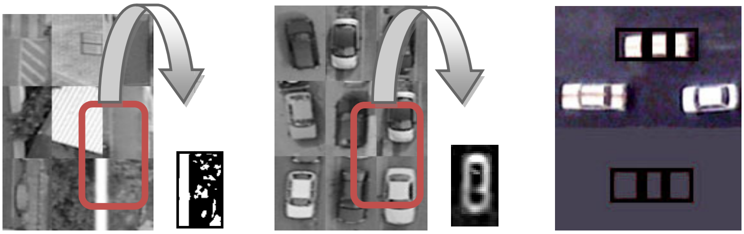


Fig. 7. Left: No-Vehicle, Middle: Vehicle Pattern, Right: Prototype of Vehicle Pattern

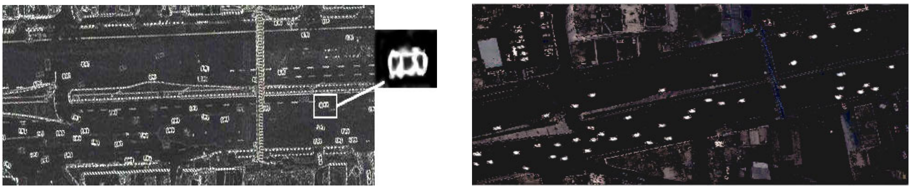


Fig. 8. Left: Morphology operation on image. Right: Vehicles after detection.

Fig. 4 shows the flowchart of the proposed vehicle detection based on the antibody learning.

After vehicle and road detection (Figure 8), we can distinguish the traffic density on the road. Because all images have been photographed from a same distance and resolution and with the number of vehicles in a specific area [12] and the use of thresholding method, we can distinguish the traffic density on the road according to the number of vehicles in a specific area and can be sent to the stations. This model also has the same deficiencies. Most important of them is that it cannot distinguish multiple roads in the image.

5 Results and Conclusions

In this research, we tried to detect road and vehicle in satellite images with very high resolution. This model can help drivers who want to pass the mentioned road, and also can help policemen to control the traffic on roads. At the present time, these images can be taken by xerography satellite and special airplanes. NAVTEQ panchromatic data set used in our study was collected from Space Imaging Inc. web site [23]. The data set contains different city pictures. A total of 6 roads segments containing over 200 vehicles were collected. Most vehicles in the images are around 8 to 10 pixels in length and around 3 to 5 pixels in width.

Since the vehicles are represented by a short number of pixels, their detection is very sensitive to the surrounding context. Accordingly, the sample database consists of vehicle and non-vehicle samples in a variety of conditions, such as road intersections, curved and straight roads, roads with lane markings, road surface discontinuity, pavement material changes, trees' shadow on the roads, etc. This represents most of the typical and difficult situations for vehicle detection.

Table 1. Results of present method

	No. of vehicles	No. of detected vehicles	No. of missing vehicles	No. of false alarm	Missing detection rate %	False detection rate %
Road1	5	5	0	0	0	0
Road2	8	8	0	0	0	0
Road3	10	10	0	1	0	10
Road4	15	15	0	1	0	6
Road5	19	17	1	1	5.3	5.2
Road6	16	15	1	0	6.2	0
Road7	23	23	2	2	8	8
Road8	60	56	6	5	10	8.3
Road9	52	47	5	4	11.5	7.6

For each selected image, roads were extracted in advance and vehicle detection was performed only on the extracted road surfaces. To build the vehicle example database, manually delineated the rectangular outer boundaries of vehicles in the image [15-16]. A total of 200 vehicles were delineated in this manner from 5 road segments. Sub-images of 10×5 pixels centered at vehicle were built in. In addition, 200 non-vehicle sub-image samples covering different road surfaces were also collected to build the non-vehicle example database (Figure (7)). The vehicle example database and used for features extraction. Results are shown in table (1).

Kuthadi Sumalatha, in his master's thesis "*Detection of Objects from High-Resolution Satellite Images*" has worked on road, vehicle and urban areas. He had used thresholding and color extraction methods in his thesis [20] (Figure (9)).

According to the results obtained, Kuthadi had carried out his research with higher accuracy in comparison with others. With the help of four images which Kutadi used for his work, we compared Kuthadi's method with the recommended method. This comparison is shown in table (2).

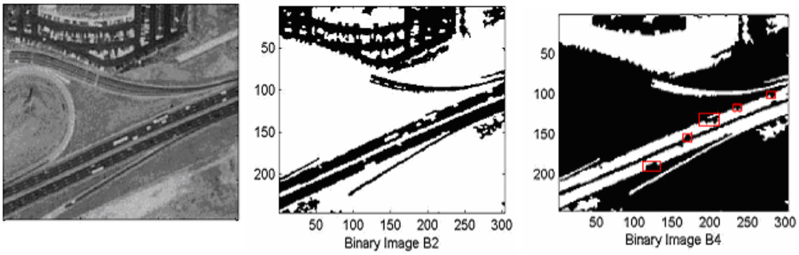


Fig. 9. Kuthadi's result for road and vehicle detection (A highway in France). Left: original image, Middle: road detection, Right: vehicle detection.

Table 2. Comparison of present method and Kuthadi method

	No. of vehicles	Kutadi's detected vehicles	Kutadi's missing vehicles	Kutadi Performance %	Present Method detected Vehicles	Present Method Performance %
Road 1	21	12	9	57.1	19	90.5
Road 2	14	10	4	71.5	13	93
Road 3	41	32	9	78	39	95.1
Road 4	9	9	0	100	9	100

References

[1] Yamazaki, F., Liu, W.: Vehicle Extraction And Speed Detection From Digital Aerial Images., IGARSS 978-1-4244- IEEE (3-08-2008)

[2] Juozapavicius, A., Blake, R., Kazimianec, M.: Image Processing in Road Traffic Analysis. Nonlinear Analysis: Modelling and Control (2005)

[3] Masakatsu, H., Toshio, H., Kouhei, T.: Traffic Queue Length Measurement Using an Image Processing Sensor (2005-12-10)

[4] Wai, H.L., Lindsay, K.: Real Time Object Tracking using Reflectional Symmetry and Motion. IEEE, Los Alamitos (2006)

[5] Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Prentice Hall, New Jersey (2002)

[6] Gonzalez Rafael, C., Woods Richard, E., Eddins_Steven, L.: Digital Image Processing Using MATLAB (Hardcover), 1st edn. Prentice Hall, Englewood Cliffs (2003)

[7] Hinz, S.: Automatic Road Etraction in Urban Scenes and Beyond (2005)

[8] Geoffrey, A.: Hollinger, Design and Construction of an Indoor Robotic Blimp for Urban Search and Rescue Tasks (2005)

[9] Zheng, H.: An Artificial Immune Approach for Vehicle Detection from High Resolution Space Imagery (2007)

[10] Arpad, B., Heipke, C.: Artificial Neural Network for The Detection of Road Junction in Aerial Images, ISPRS Archives. Part 3/W8, Munich, September 17–19, vol. XXXIV (2003)

[11] Erhan, B.: Road And Traffic Analysis from Video. In: A Thesis Submitted to the Graduate School of Engineering for the Degree of Master of Science, August 2007, Koc University (2007)

[12] Rosenbaum, D., Charmettea, B., Kurza, F., Suria, S., Thomasa, U., Reinartza, P.: "Automatic Traffic Monitoring From An Airborne Wide Angle Camera System. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Beijing, vol. XXXVII. Part B3b (2008)

[13] Stefan, H.: Automatic Road Extraction in Urban Scenes-and Beyond, Remote Sensing Technology, TU München (2004)

[14] Hu, J., Razdan, A., Femiani, J.C., Cui, M., Wonka, P.: Road Network Extraction and Intersection Detection From Aerial Images by Tracking Road Footprints. IEEE Transactions Geoscience and Remote Sensing 45(12), 4144–4157 (2007)

[15] Kim, Z., Malik, J.: High-Quality Vehicle Trajectory Generation from Video Data Based on Vehicle Detection and Description. In: Proc. IEEE Intelligent Transportation Systems Conference, pp. 176–182 (2003)

[16] Zhenfeng, Z., Hanqing, L., James, H., Keiichi, U.: Car Detection Based on Multi-Cues Integration. IEEE, Los Alamitos (2004)

[17] Albert, B., Steger, C., Mayer, H., Eckstein, W., Ebner, H.: Automatic Road Extraction, German in Photogrammetrie, Fernerkundung, Geoinformation (PFG), No. 1/99 (1999)

[18] Smith Mike, J., Chandler, J., Rose, J.: High Spatial Resolution Data Acquisition for the Geosciences: Kite Aerial Photography. Earth Surface Processes and Landforms (2008)

[19] Hong, Z., Li, L.: An Artificial Immune Approach for Vehicle Detection from High Resolution Space Imagery. IJCSNS International Journal of Computer Science and Network Security 7(2) (February 2007)

[20] Sumalatha, K.: Detection of Object from High-Resolution Satellite Images, A Thesis Submitted to the Graduate School of Engineering for the Degree of Master of Science. University of Minesota Duluth, USA (May 2005)

[21] Serra, J.: Image Analysis and Mathematical Morphology, vol. 2. Academic Press, New York (1988)

[22] National Geographical Organization, `http://www.ngo-iran.ir/Dphoto/index.html`

[23] NAVTEQ Satellite, `http://www.navteq.com/`

Graduated Assignment Algorithm for Finding the Common Labelling of a Set of Graphs[*]

Albert Solé-Ribalta and Francesc Serratosa

Universitat Rovira i Virgili (Tarragona, Spain)
{francesc.serratosa,albert.sole}@urv.cat

Abstract. In pattern recognition applications, it is useful to represent objects by attributed graphs considering their structural properties. Besides, some graph matching problems need a Common Labelling between vertices of a set of graphs. Computing this Common Labelling is an NP-complete problem. State-of-the-art algorithms are composed by two steps: in the first, they compute all pairwise labellings among the graphs and in the second, they combine this information to obtain a Common Labelling. The drawback of these methods is that global information is only considered in the second step. To solve this problem, by reducing the Common Labelling problem to the quadratic assignment one, all graphs nodes are labelled to a virtual structure whereby the Common Labeling is generated using global information. We tested the algorithm on both real-world and synthetic data. We show that the algorithm offers better performance than a reference method with same computational cost.

Keywords: Graduated Assignment, Multiple graph matching, graph common labelling, inconsistent labelling, softassign.

1 Introduction

From 80's, graphs have increase its importance in Pattern Recognition, being one of the most powerful characteristics the abstraction they achieve. Therefore, the same structure is able to represent a wide sort of problems from image understanding to interaction networks. Consequently, algorithms based on graph models are suitable in a very large problem space. There is an interesting review of graph representation models, graph matching algorithms and its applications in [1].

Sometimes in graph based Pattern Recognition applications, given a set of graphs, which all represent equivalent or related structures, it is required to find global consistent correspondences among all those graphs. These correspondences are called a Common Labelling (CL). Reference applications could be found in [2], where representations obtained from Infra-red, Optical, Cartographic and SAR images must be combined or in [3] where a prototype has to be synthesized from noisy data representing the same object.

[*] This research is supported by Consolider Ingenio 2010 (CSD2007-00018), by the CICYT (DPI 2007-61452) and by the Universitat Rovira I Virgili through a PhD research grant.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 180–190, 2010.

Unfortunately, only a few techniques to compute these correspondences have been developed when the elements are represented by Attributed Graphs (AGs). Among them we could name: [4] where optimal pairwise labelings are required or [5] and [6] where, in this case, the CL computation is based on sub-optimal pairwise labelings. Although [5] is quite more effective than [6], both share the same weakness: the use of pairwise labelings, where a simple labeling error taken at initial stages could derive in a bad global result. Moreover [5] have tendency to add extra nodes in the final CL, which might be not desired in some applications. In [2], Williams et al. introduce a method, which could induce a solution for this problem. However, this method is not extensible to N graphs. Another method, which seems to solve both problems, was published in [7]. Nevertheless, its high computational complexity makes its use infeasible with large graphs sets.

In this article, we present an energy function that represents the global cost of a given CL. Moreover, we present an algorithm, similar to the Graduated Assignment algorithm presented in [8] that iteratively seeks for a CL that maximizes this energy.

The document is structured as follows. In Section 2, we present some theoretical basis of the CL problem. In Section 3 and 4, the Graduated Assignment algorithm for graph matching [8] and our new algorithm are presented. The evaluation of our method is presented in Section 5. Finally, Section 6 finalizes the article with some conclusions.

2 Definitions

Definition 1. Attributed Graph: Let Δ_v and Δ_e denote the domains of possible values for attributed vertices and arcs, respectively. An attributed graph AG over $(\Delta_v$ and $\Delta_e)$ is defined by a tuple $AG=(\Sigma_v, \Sigma_e, \gamma_v, \gamma_e)$, where $\Sigma_v= \{v_k \mid k = 1,\ldots,R\}$ is the set of vertices (or nodes), $\Sigma_e \in \{e_{ij} \mid i,j \in \{1,\ldots,R\}, i \neq j\}$ is the set of arcs (or edges) and $\gamma_{v:}\Sigma_v \rightarrow \Delta_v$, $\gamma_{e\,:}\,\Sigma_e \rightarrow \Delta_e$ assign attribute values to vertices and arcs respectively. In case it is required, any AG can be extended with null nodes. A null node is a special AG node which has special attribute $\emptyset \in \Delta$v

Definition 2. Isomorphism between AGs: Let $G^p = (\Sigma_v^p, \Sigma_e^p, \gamma_v^p, \gamma_e^p)$ and $G^q=(\Sigma_v^q, \Sigma_e^q, \gamma_v^q, \gamma_e^q)$ be two AGs. If the selected graphs have initially different node size or it is desired to permit some extra null to vertex labelings, G^p and G^q can be extended with any number of null nodes. Besides, let T be a set of isomorphisms between two vertex sets Σ_v. The isomorphism $f^{p,q}:\Sigma_v^p \rightarrow \Sigma_v^q$, $f^{p,q} \in T$, assigns each vertex from G^p to only one vertex of G^q. There is no need to define the arcs isomorphism since they are mapped accordingly to the node isomorphism of their terminal nodes.

Definition 3. Cost and Distance between AGs: Let $f^{\,pq}$ be the isomorphism $f^{p,q}:\Sigma_v^p \rightarrow \Sigma_v^q$ that assigns each vertex from G^p to a vertex of G^q. The cost of this isomorphism, $C^G(G^p,G^q, f^{p,q})$ is a function that represents how similar are the AGs and how correct is the isomorphism. We consider this cost to be:

$$C^G\left(G^p,G^q,f^{p,q}\right)=\sum_{a=1}^{R}\sum_{b=1}^{R}\sum_{i=1}^{R}\sum_{j=1}^{R} F^{p,q}[a,i]\cdot F^{p,q}[b,j]\cdot C_{ai,bj}^{p,q} \tag{1}$$

where $F^{p,q}[a,i]$ is a permutation matrix which values are *1* if $f^{pq}(v_a^p)=v_i^q$ and $C_{ai,bj}^{p,q}$ represents the cost of matching nodes v_a^p to v_i^q and v_b^p to v_j^q plus the cost of matching the corresponding edge e_{ab}^p to e_{ij}^q.

Usually, $C^G=0$ represents that both AGs are identical and that the isomorphism captures this similarity. The distance *D* between two AGs is defined to be the minimum cost of all possible isomorphisms $f^{p,q}$. That is, $D\left(G^p,G^q\right)=\min\limits_{f^{p,q}\in T} C^G\left(G^p,G^q,f^{p,q}\right)$. We say that the isomorphism $f^{p,q}$ is *optimal* if it is the one used to compute the distance.

Definition 4. Multiple Isomorphism (MI) of a set of AGs: Let $\Gamma=\{G^1, G^2, \ldots, G^N\}$ be a set of *N* AGs. We say that the set φ is a Multiple Isomorphism of Γ if it contains one and only one isomorphism between elements, $\varphi = \{f^{1,2}, \ldots, f^{2,1}, \ldots, f^{N,N}\}$.

We assume that the AGs have *R* nodes. If it is not the case, the AGs would have to be extended with null nodes. We say that a multiple isomorphism is *consistent* if concatenating all the isomorphisms we can define disjoint *partitions* [5] of vertices. Every *partition* is supposed to contain one and only one vertex per each AG and, in addition, every vertex must belong to only one partition. **Fig 1a** shows a *Consistent Multiple Isomorphism* between three AGs, being *R=2*. We can distinguish two partitions, *P1* and *P2*. **Fig 1b.** shows the same AGs with an *Inconsistent Multiple Isomorphism*, consequently partitions can not be defined.

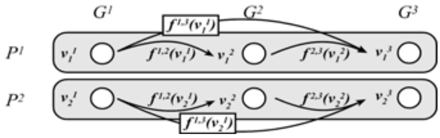


Fig. 1a. Consistent MI

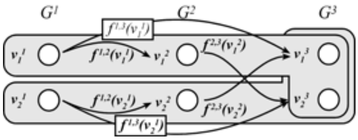


Fig. 1b. Inconsistent MI

We define the cost of a MI as the addition of the costs of all isomorphisms in φ:

$$C^{MI}(\varphi)=\sum_{p=1}^{N}\sum_{q=1}^{N}\sum_{a=1}^{R}\sum_{i=1}^{R}\sum_{b=1}^{R}\sum_{j=1}^{R}F^{p,q}[a,i]\cdot F^{p,q}[b,j]\cdot C_{ai,bj}^{p,q} \tag{2}$$

Definition 5. Consistent Multiple Isomorphism of a set of AGs (CMI): Let φ be a Multiple Isomorphism of Γ. φ is a CMI of Γ if it fulfils that:

$$f^{q,k}\left(f^{p,q}\left(v_i^p\right)\right)=f^{p,k}\left(v_i^p\right),\ 0<p,q,k\le N, 0<i\le R \tag{3}$$

We define the cost of a CMI as the cost of the related MI. The Optimal Consistent Multiple Isomorphism (OCMI) is the CMI with the minimum cost. Note that, the cost of the OCMI may be obtained by non-optimal isomorphisms since it is restricted to be consistent.

Definition 6. Optimal Consistent Multiple Isomorphism of a set of AGs (OCMI): Let φ be a CMI of Γ. φ is an Optimal Consistent Multiple Isomorphism (OCMI) of Γ if it fulfils that $\varphi = \underset{f^{pq} \in T}{\arg\min} \sum_{\forall G^p, G^q} C^G\left(G^p, G^q, f^{p,q}\right)$.

Given Γ, we can define a Common Labelling (CL) which is a bijective mapping between all graph nodes in the AGs to a virtual structure. We construct this CL through a CMI. The CMI requirements are mandatory due to if not, the CL would not be a bijective since an AG node would have to be labeled to several nodes of the virtual structure.

Definition 7. Common Labelling of a set of AGs (CL): Let φ be a CMI of Γ and let L be a vertex set, $L \in \Sigma_v$. The Common Labelling $\psi = \{ h^1, h^2, \ldots, h^n \}$ is defined to be a set of bijective mappings from the vertices of AGs to L as follows:

$$h^1(v_i^1)=i,\ h^p(v_i^p)=h^{p-1}(v_j^{p-1}),\ 1 \leq i,j \leq R,\ 2 \leq p \leq N, \text{ being } f^{p-1,p}(v_j^{p-1})=v_i^p. \tag{4}$$

Fig. 2 illustrates this definition.

Finally, the Optimal Common Labelling of a set is a CL computed through an OCMI. The prototype or representative of the set synthesized using this CL would be the best representative, from the statistical point of view, since the sum of the costs of each pair of AGs, considering the global consistency requirement, is the lowest among all possible CL.

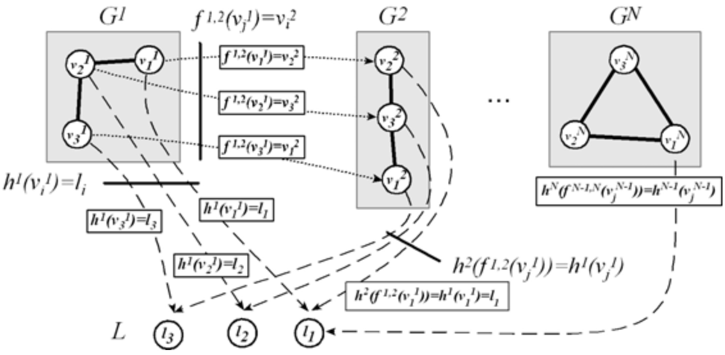


Fig. 2. From a CMI to a CL

Definition 8. Optimal Common Labelling of a set of AGs (OCL): Let ψ be a CL of Γ computed by a CMI φ. We say that ψ is an Optimal Common Labelling (OCL) of Γ if φ is an OCMI of Γ.

3 Common Labelling Framework

Given two graphs G^p and G^q, there are several error-tolerant graph matching algorithms that return the best isomorphism $f^{p,q}$ between them, given a minimization criteria. Considering that these graphs have a degree of disturbance and also the exponential complexity of the problem, some of these algorithms [8], [9], [10], [11] do not return exactly isomorphism $f^{p,q}$ but a probability matrix related to it. We represent this matrix by $P_f^{p,q}$ where each cell contains:

$$P_f^{p,q}[i,j] = Prob(f^{p,q}(v_i^p) = v_j^q) \tag{5}$$

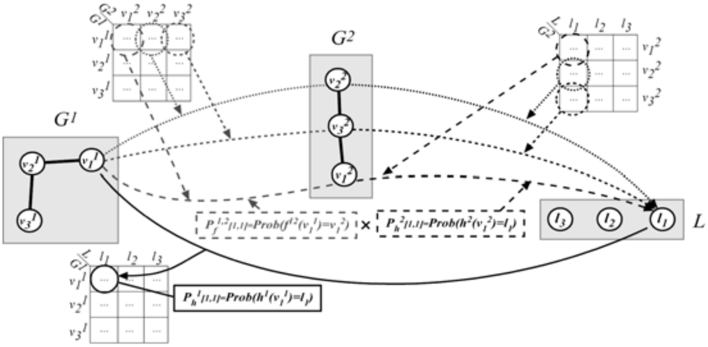


Fig. 3. Probability of matching $v_1^{\,1}$ to l_1.

To adapt the CL problem to the matching problem, we define the probability of match between a graph node v_i^p and a virtual node l_j as $P_h^p[i,j]=Prob(h^p(v_i^p)=l_j)$ (Fig. 3). Both, $P_f^{p,q}$ and $P_h^{\,p}$ are stochastic matrices [12], note that $F^{p,q}$ in (1) and (2) is a special case of $P_f^{p,q}$, when $P_f^{p,q}$ is composed by zeros and ones. At the end of the proposed algorithm, it is necessary to convert $P_f^{p,q}$ into $f^{p,q}$ and $P_h^{\,p}$ into h^p. There are several techniques to find these isomorphisms, e.g. [8], [13], which are out of the scope of this paper. We will indentify this discretization process as Λ.

We consider, as Fig 3 depicts, that the probability of matching a vertex v_i^p of graph G^p, to a vertex l_j of the virtual structure L is the probabilistic union of all the paths that goes through the nodes of a third graph G^q. That is,

$$Prob\left(h^p\left(v_i^p\right)=l_j\right) = Prob\left\{\begin{array}{c}\left[f^{p,q}\left(v_i^p\right)=v_1^q \wedge h^q\left(v_1^q\right)=l_j\right] \vee \left[f^{p,q}\left(v_i^p\right)=v_2^q \wedge h^q\left(v_2^q\right)=l_j\right] \vee \\ \vee \ldots \vee \left[f^{p,q}\left(v_i^p\right)=v_R^q \wedge h^q\left(v_R^q\right)=l_j\right]\end{array}\right\}, \tag{6}$$

Combining (6) with P_f and P_h definitions and assuming independence of events we have:

$$P_h^p[i,j] = \sum_{k=1}^{R} P_f^{p,q}[i,k] \cdot P_h^q[k,j] \quad \text{from where} \quad P_h^p = P_f^{p,q} \cdot P_h^q \tag{7}$$

In a similar way, we could infer that $P_f^{p,q} = P_h^{\,p} \cdot (P_h^{\,q})^T$.

Hence, following (7) we could obtain $P_h^{\,p}$ in several equivalent ways if Λ(φ) is a CMI,

$$\begin{aligned} &h^1 = \Lambda(P_h^1) = \Lambda(identity\_matrix) \\ &h^2 = \Lambda(P_h^2) = \Lambda(P_f^{2,N} \cdot P_h^N) = \Lambda(P_f^{2,N-1} \cdot P_h^{N-1}) = \ldots = \Lambda(P_f^{2,1} \cdot P_h^1) = \Lambda(P_f^{2,1}) \\ &\ldots \\ &h^N = \Lambda(P_h^N) = \Lambda(P_f^{N,N-1} \cdot P_h^{N-1}) = \Lambda(P_f^{N,N-2} \cdot P_h^{N-2}) = \ldots = \Lambda(P_f^{N,1} \cdot P_h^1) = \Lambda(P_f^{N,1}) \end{aligned} \tag{8}$$

However, in real data (due to distortion on the object representation and distortion induced by sub-optimality of the matching algorithms), it is usual that [2]:

$$\begin{aligned} &(P_h^p)' = P_f^{p,1} \cdot P_h^1 \,, (P_h^p)'' = P_f^{p,2} \cdot P_h^2 \,, \ldots, (P_h^p)''^{\ldots'} = P_f^{p,N} \cdot P_h^N \rightarrow \\ &\rightarrow \Lambda((P_h^p)') \neq \ldots \neq \Lambda((P_h^p)''^{\ldots'}) \end{aligned} \tag{9}$$

For this reason, probabilities P_h^p cannot be computed directly through matrices $P_f^{p,q}$, as in (8), when we cannot assume that $P_f^{p,q}$ will compose a CMI.

In this article, we propose and algorithm for the computation of a suboptimal solution to the CL problem, the algorithm is inspired in the Graduated Assignment. For ease of understanding, we first present an overview of the Graduated Assignment algorithm to later introduce the proposed algorithm.

4 The Graduated Assignment Algorithm

The Graduated Assignment algorithm is probably the most popular algorithm to compute a suboptimal solution for the graph isomorphism problem. Its cornerstone is how it reduces the referenced problem to the quadratic assignment problem. The proposed development starts by defining the energy of an isomorphism as:

$$E^{G^p,G^q} = -\sum_{a=1}^{R}\sum_{i=1}^{R}\sum_{b=1}^{R}\sum_{j=1}^{R} P_f^{p,q}[a,i] \cdot P_f^{p,q}[b,j] \cdot C_{ai,bj}^{p,q} \tag{10}$$

They approximate E^{G^p,G^q}, at point $\left(P_f^{p,q}\right)^0$, using Taylor series expansion as:

$$\begin{aligned} &E^{G^p,G^q} \approx (E^{G^p,G^q})' = -\sum_{a=1}^{R}\sum_{i=1}^{R}\sum_{b=1}^{R}\sum_{j=1}^{R} \left(P_f^{p,q}[a,i]\right)^0 \cdot \left(P_f^{p,q}[b,j]\right)^0 \cdot C_{ai,bj}^{p,q} \\ &-\sum_{a}^{R}\sum_{i}^{R}\left[\sum_{b}^{R}\sum_{j}^{R}\left(P_f^{p,q}[b,j]\right)^0 \cdot C_{ai,bj}^{p,q}\right]\left(P_f^{p,q}[a,i]-\left(P_f^{p,q}[a,i]\right)^0\right) \end{aligned} \tag{11}$$

analyzing the approximation it is seen that:

$$\arg\min\{E'\} \equiv \arg\max\left\{\sum_{a=1}^{R}\sum_{i=i}^{R} Q_{a,i}^{p,q} \cdot P_f^{p,q}[a,i]\right\} \,,\ Q_{a,i}^{p,q} = \left[\sum_{b}^{R}\sum_{j}^{R}\left(P_f^{p,q}[b,j]\right)^0 C_{ai,bj}^{p,q}\right] \tag{12}$$

```
Program Graduated_Assignment input
$G^p$,$G^q$ returns f
 Initialise $P_f^{p,q}$
 Begin A: (Do A until β ≥ βf)
  Begin B: (Do B until $Q_{a,i}^{p,q}$ converges)
   $Q_{a,i}^{p,q} = \sum_{b=1}^{R} \sum_{j=1}^{R} \left(P_f^{p,q}[b,j]\right)^0 \cdot C_{ai,bj}^{p,q}$
   $P_f^{p,q}[a,i] = \exp(\beta \cdot Q_{a,i}^{p,q})$
   Begin C:
    $P_f^{p,q}[a,i] = P_f^{p,q}[a,i] \Big/ \sum_{a=1}^{R} P_f^{p,q}[a,i]$
    $P_f^{p,q}[a,i] = P_f^{p,q}[a,i] \Big/ \sum_{i=1}^{R} P_f^{p,q}[a,i]$
   End C
  End B
 End A
 $f = \Lambda(P_f^{p,q})$
End Program
```

Algorithm 1. GA algorithm

```
Program CL input Γ returns ψ
 Initialise $P_h$
 Begin A: (Do A until β ≥ βf)
  Begin B: (Do B until $Q_{a,\omega_1}^{p}$ converges)
   Compute $Q_{a,\omega_1}^{p}$ (Alg. 3 or Alg. 4)
   $P_h^p[a,i] = \exp(\beta \cdot Q_{a,\omega_1}^{p})$   1<p<=N
   Begin C:
    $P_h^p[a,i] = P_h^p[a,i] \Big/ \sum_{a=1}^{R} P_f^p[a,i]$        1<p<=N
    $P_h^p[a,i] = P_h^p[a,i] \Big/ \sum_{i=1}^{I} P_h^p[a,i]$       1<p<=N
   End C
  End B
 End A
 $\Psi = \Lambda(P_h^p)$
End Program
```

Algorithm 2. CL algorithm

The algorithm proposed in [8] minimizes (10) under the assumption that it minimizes at same point as (12) is maximized. In this way, the problem is equivalent to the quadratic assignment one, where Q represents a cost matrix, and P_f represents a stochastic matrix [12] which contains the desired assignation probability.

The Graduated Assignment algorithm proceeds in the following way: start with a valid P_f, compute cost matrix Q given by (12), apply softassign to compute P_f and start again. A pseudo code of the Graduated Assignment is listed in Algorithm 1.

5 N-Graduated Assignment for the CL Problem

The methodology that we present applies a similar procedure as the Graduated Assignment methodology to solve the CL problem. The proposed algorithm instead of computing an isomorphism of two graphs, it computes the isomorphism of a set of graphs Γ and in addition, it imposes to those isomorphisms to be consistent (3).

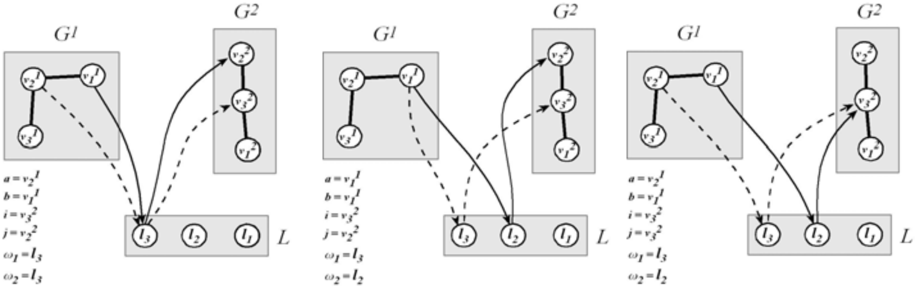


Fig. 4a. non valid: ω_1=ω_2 **Fig. 4b.** non valid: a=b **Fig. 4c.** non valid: i=j

Due to our objective is to compute a CL, our new energy function depends on the probabilities P_h instead of P_f. Nevertheless, the CL has to represent consistent and bijective isomorphisms between the involved graphs and the virtual structure, for this reason, we impose the restrictions *a≠b, i≠j and* $\omega_1 \neq \omega_2$. Fig. 4a,b,c shows non valid isomorphisms. Our new energy function is,

$$E^{CL} = \sum_{\forall p\in\Gamma}\sum_{\forall q\in\Gamma}^{q\neq p}\sum_{a=1}^{R}\sum_{i=1}^{R}\sum_{b=1,b\neq a}^{R}\sum_{j=1,j\neq i}^{R}\left(\sum_{\omega_1=1}^{R} P_h^p[a,\omega_1]\cdot P_h^q[i,\omega_1]\cdot\left[\sum_{\omega_2=1,\omega_2\neq\omega_1}^{R} P_h^p[b,\omega_2]\cdot P_h^q[j,\omega_2]\right]\right)\cdot C^{p,q}_{ai,bj} \qquad (13)$$

From (13) we compute the Taylor series expansion deducing that in our case *Q* is given by:

$$\begin{aligned} Q^p_{a,\omega_1} &= \frac{\partial E^{CL}}{\partial P_h^p[a,\omega_1]} = \\ &= \sum_{\forall q\in\Gamma}^{q\neq p}\sum_{i=1}^{R}\left(\sum_{b=1,b\neq a}^{R}\sum_{j=1,j\neq i}^{R} P_h^q[i,\omega_1]\cdot\left[\sum_{\omega_2=1,\omega_2\neq\omega_1}^{R} P_h^p[b,\omega_2]\cdot P_h^q[j,\omega_2]\right]\cdot C^{p,q}_{ai,bj}\right) \end{aligned} \qquad (14)$$

Finally, Algorithm 2 obtains a *CL* ψ given a set of graphs Γ. Note that in (4), we impose $h^1(v_i^1)=l_i$. For this reason, in the algorithm we present, we impose P_h^1 to be the identity matrix throughout the iterative process. This requirement is due to the fact that the virtual structure does not contain any type of attributes nor structure. Forcing nodes of G^1 to concrete nodes of the virtual structure *L*, we force the other graphs to label each other according to this prior labeling. The other probability matrices can be initialized to any stochastic matrix.

Function *Exact_Q* computes *Q* (14) with a cost of $O(N^2\cdot R^6)$(Algorithm 3). But, with the aim of reducing this cost, we have relaxed constraint $\omega_1 \neq \omega_2$ in (14) (Fig. 4.a) which allows to compute an approximation of (14) with a cost of $O(N^2\cdot R^4)$. Algorithm 4 shows the pseudocode. We don't show evaluation results of the *Exact_Q* due to the results are equivalent to the approximation ones. Moreover, it can be proven that, with large size of Γ, the noise introduced by the non valid isomorphism when $\omega_1=\omega_2$ in the approximation algorithm is not significant.

```
Function Exact_Q input $P_h$, Γ returns Q
for $\forall p \in \Gamma$
 for $a = 1..R$
  for $\omega_1 = 1..R$
   $Q^p_{a,\omega_1} = 0$
   for $\forall q \in \Gamma, q \neq p$
    for $i = 1..R$
     for $b = 1..R, b \neq a$
      for $j = 1..R, j \neq i$
       $v_2 = 0$
       for $\omega_2 = 1..R, \omega_2 \neq \omega_1$
        $v_2 = v_2 + P_h^p[b,\omega_2]\cdot P_h^q[j,\omega_2]$
       end
       $Q^p_{a,\omega_1} = Q^p_{a,\omega_1} + P_h^q[i,\omega_1]\cdot v_2 \cdot C^{p,q}_{ai,bj}$
      end
     end
    end
   end
  end
end Function
```

Algorithm 3. Calculus of Q

```
Func Approx_Q input $P_h$, Γ returns Q
for $\forall p \in \Gamma$
  $Q^p = [0]$
  for $\forall q \in \Gamma, q \neq p$
    $P_f^{p,q} = P_h^p \cdot (P_h^q)^T$
    for $a = 1..R$
      for $i = 1..R$
        $v_1 = 0$
        for $b = 1..R, b \neq a$
          for $j = 1..R, j \neq i$
            $v_1 = v_1 + P_f^{p,q}[b,j]\cdot C^{p,q}_{ai,bj}$
          end
        end
        for $\omega_1 = 1..R$
          $Q^p_{a,\omega_1} = Q^p_{a,\omega_1} + v_1 \cdot P_h^q[i,\omega_1]$
        end
      end
    end
end Function
```

Algorithm 4. Calculus of approx Q

6 Evaluation

To evaluate the presented algorithm we have compared to the algorithm presented by Bonev *et al.* [5]. We consider it is the only one generic enough. The method applies the Graduated Assignment algorithm [8] to compute the N^2 possible probabilistic assignation matrices P_f between the graphs. Next, the $N^2 \cdot R^2$ probabilities $P_f^{p,q}$ are sorted and processed in descending order to compute what they call a Super-graph. The cost of the algorithm is $O(N^2 \cdot (\#iterations \cdot R^4))$.

We evaluate both algorithms using two datasets composed by AGs that represent objects embedded in the plane. In both cases, nodes are defined over a two-dimensional domain that represents its plane position (x, y). Edges have a binary attribute that represents the existence of a line between two terminal points. The former dataset, created synthetically, is composed by 35 classes. The number of graphs per class is $N \in [3, 5, 7, 9, 11]$ and the noise level between graphs is $v \in [10, 20, 40...80]$. Therefore, we defined 5 x 7 = 35 different classes: seven classes with four graphs (with different noise levels), seven classes with five graphs (with different noise levels), and so on. Each class was created as follows. We randomly generate a base graph composed of $R=10$ nodes with random attributes in the range $\Delta_v=[0..100, 0..100]$. Edges are defined by the Delaunay triangulation. Then, with this base graph, we created N other graphs by: 1, generating Gaussian noise at every node with standard deviation $\sigma= v/100$. 2, removing $v\%$ nodes randomly. 3, inserting $v\%$ nodes (with random attributes) and 4, changing the state of $v\%$ edges. The latter dataset, created at the University of Bern [14], is called Letter. It is composed of 15 classes and 150 graphs per class representing the Roman alphabet i.e. A, E, F, …, X, Y, Z. From each class, we randomly selected $N \in [3, 5, 7, 9, 11]$ graphs, to generate the CL. To compute the cost C in equations (2), (12) and (14) we used the Edit Distance [15] applied to the sub-graphs induced by $\{v_a^p, v_b^p\}$ and $\{v_i^q, v_j^q\}$. Finally, with the aim of obtaining non-biased results, each experiment was performed 7 times.

The ground truths of our experiments are the MIs in which each isomorphism has been computed through Algorithm 1. Note that these MIs are not restricted to be consistent (3) and so, do not compose a CL (Def. 7). Their costs are computed through C^{MI} (2) and they are supposed to be the lowest ones due to the consistency restriction are not imposed. The results of the evaluation procedure are presented in Fig. 5 for the Letter dataset and in Table 2 for the Synthetic dataset. Each point in Fig. 5 shows the mean cost minus the cost of the ground truth of an experiment set performed by both algorithms. Each set is constructed by 7 random experiments with letter 'A', 7 with letter 'B', … and 7 with letter 'Z' given a concrete size of Γ. Besides, we present the results using the synthetic dataset in Table 1. In this case, each cell of the table represents the percentage of increment of the proposed algorithm in comparison with [5]. Each value is computed using the mean of 7 random experiments using a concrete size of Γ and a concrete noise level.

We see in Fig. 5 that the presented algorithm achieve better CLs than [5], we observe that as the size of Γ increases the performance of the presented algorithm tens also to increase respect [5]. In the results performed over the synthetic dataset (Table 1), we can observe that with noises greater than 10, the percentage of increment is considerable and also tends to increase together with the size of Γ and the noise level.

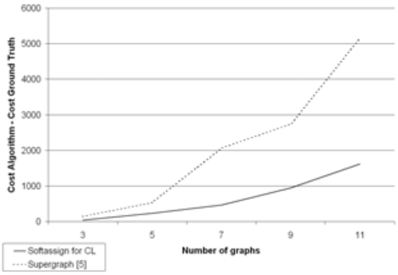
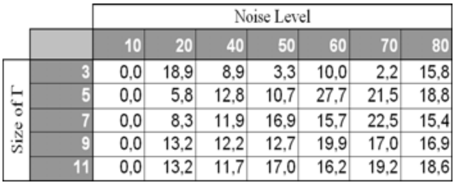


Fig. 5. Results of Letter dataset

Table 1. Results of Synthetic dataset



		Noise Level						
		10	20	40	50	60	70	80
Size of Γ	3	0,0	18,9	8,9	3,3	10,0	2,2	15,8
	5	0,0	5,8	12,8	10,7	27,7	21,5	18,8
	7	0,0	8,3	11,9	16,9	15,7	22,5	15,4
	9	0,0	13,2	12,2	12,7	19,9	17,0	16,9
	11	0,0	13,2	11,7	17,0	16,2	19,2	18,6

7 Conclusions and Further Work

Graphs are a very flexible representation of data capable of representing a large sort of problems related to pattern recognition. Examples could be found in image databases, video analysis, biomedical and biological applications and so on. A nice review can be found in [1]. In some of these applications, it is usual the need of finding a structure that represents a set of graphs. This structure is used as a representative of the set. The first step to generate this structure is to find a Common Labeling between the vertices of all the graphs such that a general cost is minimized. It is crucial to find a good common labeling to generate a good representative. In addition some works [2] deduce that, due to the noise, in some applications it is more useful to find a CL (of three graphs) instead of just pairwise labellings.

Known algorithms to compute a Common Labeling consist on first finding the labeling between any pairs of graphs and then combining this information to compute the Common Labeling. The presented algorithm differs from others because it computes the Common Labeling at the same time as the pairwise labelings, mixing the local and global knowledge at each step of the algorithm.

We have compared our algorithm with the most popular one in the literature and we present and evaluation which shows that our method finds better common labelings with similar computational cost. This means that, the approaches that need a Common Labeling between graphs would perform better. Moreover, the proposed iterative approach allows using the current Common Labeling at each step of the algorithm, in comparison with [5] which must wait for all pairwise computations before concluding a solution.

As a future work, we will apply this new technique to the representative of a set of graphs called Structurally-Defined Random Graph [3] and we will analyze its ability to keep the structural and semantic knowledge of the Γ set.

References

[1] Conte, D., et al.: Thirty Years Of Graph Matching In Pattern Recognition. IJPRAI 18(3), 265–298 (2004)

[2] Williams, M.L., Wilson, R.C., Hancock, E.R.: Multiple Graph Matching with Bayesian Inference. PRL 18(11-13), 1275–1281 (1997)

[3] Solé-Ribalta, A., Serratosa, F.: A structural and semantic probabilistic model for matching and representing a set of graphs. In: Torsello, A., Escolano, F., Brun, L. (eds.) GbRPR 2009. LNCS, vol. 5534, pp. 164–173. Springer, Heidelberg (2009)

[4] Wong, A.K.C., et al.: Entropy and distance of random graphs with application to structural pattern recognition. IEEE TPAMI 7, 599–609 (1985)

[5] Bonev, B., et al.: Constellations and the unsupervised learning of graphs. In: Escolano, F., Vento, M. (eds.) GbRPR. LNCS, vol. 4538, pp. 340–350. Springer, Heidelberg (2007)

[6] Serratosa, F., et al.: Synthesis of function-described graphs and clustering of attributed graph. IJPRAI 16(6), 621–655 (2002)

[7] Solé-Ribalta, A., Serratosa, F.: On the Computation of the Common Labelling of a set of Attributed Graphs. In: Bayro-Corrochano, E., Eklundh, J.-O. (eds.) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. LNCS, vol. 5856, pp. 137–144. Springer, Heidelberg (2009)

[8] Gold, S., Rangarajan, A.: A Graduated Assignment Algorithm for Graph Matching. IEEE TPAMI 18(4), 377–388 (1996)

[9] Christmas, W.J., Kittler, J., Petrou, M.: Structural matching in computer vision using probabilistic relaxation. IEEE TPAMI 17(8), 749–764 (1995)

[10] Rosenfeld, A., Hummel, R.A., Zucker, S.W.: Scene labeling by relaxation operators. IEEE Transactions on Systems, Man and Cybernetics 6, 420–443 (1976)

[11] O'leary, D.P., Peleg, S.: Analysis of relaxation processes: The two-node label case. IEEE Transactions on Systems, Man and Cybernetics 13, 618–623 (1983)

[12] Sinkhorn, R.: A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices. The Annals of Mathematical Statistics 35(2), 876–879 (1964)

[13] Kuhn, H.W.: The Hungarian method for the assignment problem Export. Naval Research Logistics Quarterly 2(1-2), 83–97 (1955)

[14] Riesen, K., Bunke, H.: IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 287–297. Springer, Heidelberg (2008)

[15] Sanfeliu, A., Fu, K.S.: A Distance Measure Between Attributed Relational Graphs for Pattern Recognition. Trans. Systems, Man, and Cybernetics 13, 353–362 (1983)

Affinity Propagation for Class Exemplar Mining

Shengping Xia[1,⋆], Rui Song[1], and Edwin R. Hancock[2,⋆⋆]

[1] ATR Lab, School of Electronic Science and Engineering, National University of Defense Technology, Changsha, Hunan, P.R. China 410073

[2] Department of Computer Science, University of York, York YO1 5DD, UK

Abstract. This paper focusses on the problem of locating object class exemplars from a large corpus of images using affinity propagation. We use attributed relational graphs to represent groups of local invariant features together with their spatial arrangement. Rather than mining exemplars from the entire graph corpus, we prefer to cluster object specific exemplars. Firstly, we obtain an object specific cluster of graphs using a similarity propagation based graph clustering (SPGC) method. Here a SOM neural net based tree clustering method is used to incrementally cluster a large corpus of local invariant descriptors. The popular affinity propagation based clustering algorithm is then individually applied to each object specific cluster. Using this clustering method, we obtain object specific exemplars together with a high precision for the data associated with each exemplar. The strategy adopted is one of divide and conquer, and this greatly increases the efficiency of mining exemplars. Using the exemplars, we perform recognition using a majority voting strategy that is weighted by nearest neighbor similarity. Experiments are performed on over 80K images spanning ~500 objects, and demonstrate the performance in terms of efficiency, scalability and recognition.

1 Introduction

One of the most effective means of knowledge discovery from a large corpus of data is to search for class exemplars. Detecting exemplars goes beyond simple clustering, as the exemplars themselves store compressed information. Frey and Dueck [1] propose an affinity propagation method for locating an optimal set of exemplars. Each data item in the corpus is then associated with the exemplar that best represents it. Affinity propagation uses an index of similarity $s(i,k)$ to indicate how well the data item with index k is suited to be the exemplar for data item i. Affinities are updated in an iterative manner reminiscent of both belief propagation and relaxation labeling. The method can be applied to high dimensional vectors, graphs or any data-structure provided a suitable similarity measure can be defined.

Many knowledge discovery tasks require the identification of exemplars from among a set of sparsely related data, i.e., where most similarities are either unknown or large and negative. To deal with affinity propagation in this case, a sparse similarity matrix

[⋆] Supported by Chinese NSF Foundation project (contract 60972114).

[⋆⋆] Supported by a Royal Society Wolfson Research Merit Awand and EU FP7 FET SIMBAD project (contract 213250).

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 191–201, 2010.

(SSM) with similarity set to $-\infty$ is used. When affinity propagation is applied to this sparse similarity matrix, because messages need not be exchanged between i and k if $s(i,k) = -\infty$, each iteration requires exchanging messages only between a very small subset of the data pairs. Another interesting capability of affinity propagation is that is can be applied to asymmetric or non-metric similarities (i.e. those for which $s(i,k) \neq s(k,i)$) and those for which the similarities do not satisfy the triangle inequality (i.e., $s(i,k) < s(i,j) + s(j,k)$).

The affinity propagation method has been applied to a moderately large image corpus [1]. However, it is still difficult to apply the algorithm to a very large sparse similarity matrix, for instance where the number of data items exceed 10^7. Furthermore, it is difficult to efficiently obtain a sparse similarity matrix for large datasets, and this can cause difficulties in locating exemplars. In this paper we address the question of how to obtain exemplars of a specific object rather than all exemplars of a large image corpus, and how to propagate affinity in this context.

Recently, Xia and Hancock have shown how graph clustering can be effected using similarity propagation and used to discover the set of object classes present in a database of images[7][6][4]. In this work, each image of an object is represented by a graph constructed from a selected group of robust SIFT features. For each pair of graphs, a similarity measure is computed using the cardinality of the maximum common subgraph and the consistency of geometric spatial alignment of the image features. A recursive self organizing map is used to locate a clustering tree (termed RSOM) for the SIFT descriptors, which is then incrementally trained using the method outlined in [2]. For each graph, the K nearest neighbor graphs under the pairwise graph similarity measure can be efficiently located using the RSOM clustering tree.

In this paper, we propose an integrated framework for obtaining object specific image exemplars based on the more principled use of affinity propagation. The paper is organized as follows. In Section 2, we introduce some preliminaries for our work. In Section 3, we present the outline of our method used for exemplar mining. We present experimental results in Section 4 and conclude the paper in Section 5.

2 Ingredients of Our Method

2.1 Image Representation

For each image in the dataset local invariant features are detected. A variety of feature detectors have been developed [8][14][9][15], and these include SIFT [9] and SURF (Speeded Up Robust Features) [8]. We use the method proposed in [5] to extract a selected number $\mathcal{T}$, e.g. $\mathcal{T} = 40$, of salient SIFT features. Each group of selected local features together with their spatial arrangement is regarded as a semantic visual entity. This kind of structured data can be represented by using attributed graphs G [11] (hereafter simply graphs). We can obtain a set of graphs $\mathbb{G} = \{G_l,\ l = 1, 2, ..., N\}$ from a set of images.

2.2 Pairwise Graph Matching

As shown in [10][16][20], the recognition or retrieval results can be significantly improved using the geometry of spatial feature arrangement to verify consistency. In our approach, on the other hand, each image is represented by a graph. As a result the spatial verification problem becomes one of pairwise graph matching (PGM). We perform PGM with the aim of finding a maximum common subgraph (MCS) between two graphs G_l and G_q, and the result is denoted as $MCS(G_l,G_q)$. There are a plethora of available methods for finding matching features consistent with a given set of geometric constraints, and the problem has been proven to be NP-hard. RANSAC provides one popular set of methods, however their implementation is slow [18]. In [3], pairwise graph matching is achieved by combining SIFT feature matching and iterative Procrustes alignment [19]. The method can not only be used to align the feature points, but can also be used to discard those features that do not satisfy the spatial arrangement constraints. Given the $MCS(G_l, G_q)$ obtained by PGM, a similarity measure between the graphs G_l and G_q is defined as follows:

$$R(G_l, G_q) = \|MCS(G_l, G_q)\| \times (\exp(-e(X_l, X_q)))^{\kappa}. \tag{1}$$

Here a) $\|MCS(G_l, G_q)\|$ is the cardinality of the MCS of G_l and G_q, b) κ is the number of roughly mismatched feature pairs by SIFT matching, which is used to amplify the influence of the geometric dissimilarity between X_l and X_q, and c) X_l and X_q are respectively the position coordinates in graphs G_l and G_q corresponding to the vertices of $MCS(G_l, G_q)$.

2.3 Obtaining K-Nearest Neighbors Using RSOM Tree

Consider the graph set $\mathbb{G} = \{G_q,\ q = 1, 2, ..., N\}$. For each graph $G_l \in \mathbb{G}$, and the remaining graphs in the set ($\forall G_q \in \mathbb{G}$), we obtain the pairwise graph similarity measures $R(G_l, G_q)$ using Equation (1). Using the similarity measures we rank the graphs in descending order of similarity and the K top-ranked graphs are defined as the generalized K-nearest neighbor graphs (KNNG) of graph G_l, denoted as $\mathbb{K}\{G_l\}$.

With increasing size of the graph dataset, it becomes time consuming to obtain all $\mathbb{K}\{G_l\}$ if a sequential search strategy is adopted. However, in a large graph set, most of the values of the the similarity measures are very low. For a single graph G_l, if we can efficiently find a subset $\mathbb{G}'$ with significant similarity values from the complete set $\mathbb{G}$ as a filtering stage. Then we only need to perform pairwise graph matching for this subset. To this end, we employ a tree based clustering method.

We use the incremental clustering tree-RSOM reported in [2] for incrementally learning a large corpus of SIFT descriptors. To obtain $\mathbb{K}\{G_l\}$ for each training graph using a trained RSOM tree we proceed as follows. Given a graph G_l, we find the winner of the leaf nodes for each descriptor of this graph and define the union of all graphs for the winners as follows:

$$UG\{G_l\} = \{\ G_q \ \mid\ U_q^j \in G_q, U_q^j \in WL\{U_l^t\}, U_l^t \in G_l\}. \tag{2}$$

where $WL\{U_l^t\}$ is the winner of the leaf nodes for descriptor U_l^t. The frequency of graph G_q, denoted as H_q, represents the number of roughly matched descriptors between two graphs. Since we aim to obtain $\mathbb{K}\{G_l\}$, we need not process all graphs in the subsequent stages. We rank the graphs in $UG\{G_l\}$ according to decreasing frequency H_q of graph

G_q. From the ranked list, we select the first K graphs, denoted by $\mathbb{K}^{'}\{G_l\}$ as follows:

$$\mathbb{K}^{'}\{G_l\} = \{\, G_q \;\mid\; G_q \in UG\{G_l\}, H_q > H_{q+1}, q = 1, 2, ..., K.\}. \tag{3}$$

For each graph G_q in $\mathbb{K}^{'}\{G_l\}$, we will obtain the similarity measure according to Equation (1) and then $\mathbb{K}\{G_l\}$ can be obtained.

2.4 Similarity Propagation Based Graph Clustering(SPGC)

In the text retrieval literature, a standard method for improving performance is query expansion. The query expansion strategy used in [4] is based on the RSOM tree and the set $\mathbb{K}\{G_l\}$ for each graph, obtained in the training stage. Stated simply, the method is as follows. A group of graphs are referred to as siblings of a given graph G_l provided they satisfy the following condition:

$$S\{G_l\} = \{G_q \in \mathbb{K}\{G_l\} \mid R(G_l, G_q) \geq \tau\} \triangleq \; S_\tau\{G_l\}. \tag{4}$$

where τ is a similarity threshold. We use the definition to recursively obtain the family tree for the graph G_l, and this is formally defined as follows.
Family Tree of a Graph **(FTOG):** For any given similarity threshold τ, an FTOG of G_l with k generations and denoted as $M\{G_l, k\}$, is defined as follows:

$$M\{G_l, k\} = M\{G_l, k-1\} \bigcup_{G_q \in L\{G_l, k-1\}} S_\tau\{G_q\}. \tag{5}$$

where, if $k = 1$, $L\{G_l, 1\} = L\{G_l, 0\} \bigcup S\{G_l\}$ and $M\{G_l, 0\} = \{G_l\}$; and the process stops when $M\{G_l, k\} = M\{G_l, k+1\}$. An FTOG, whose graphs satisfy the restriction defined in Equation (4), can be regarded as a cluster of graphs. However, it must be stressed that this is not a clustering method based on a central prototype.

2.5 Affinity Propagation Clustering

Affinity propagation is a clustering method, which commences by considering all the data items as potential exemplars, and then recursively transmits real-valued messages along edges of a network whose nodes are data items. At any item and at any time, the magnitude of each message reflects the current affinity (or support) provided by one node for another as its potential exemplar [1]. After a number of iterations, a good set of exemplars and corresponding clusters emerges. The input of affinity propagation is a collection of real-valued similarities between data items, where the similarity $s(i, k)$ indicates how well data point k is suited as the exemplar for data-point i. In affinity propagation, the number of clusters is not required to be specified. The method can be biassed by adjusting values of $s(i, i)$ referred to as the ”preference”. A data point with a large value of $s(i, i)$ is more likely to be chosen as an exemplar. Two kinds of messages are exchanged between data items, namely ”responsibility” and ”availability”. The ”responsibility” $r(i, k)$, transmitted from item i to item k, reflects how well-suited data item k is to serve as the exemplar for data item i. The ”availability” $a(i, k)$, transmitted from candidate exemplar item k to item i, reflects the accumulated evidence for choosing item

k as the exemplar of item i, given the support from the remaining items in the network. Updating takes place according to the following rules:

$$\begin{cases} r(i,k) := s(x_i, x_k) - \max_{k' \neq k}\{a(i,k') + s(x_i, x_{k'})\} \\ a(i,k) := min\{0, r(k,k) + \sum_{i' \sim \in \{i,k\}} max\{0, r(i',k)\}\} \end{cases} \tag{6}$$

The self-availability is updated in a slightly different way as

$$a(k,k) = \sum_{i' \neq k} max\{0, r(i',k)\}. \tag{7}$$

Upon convergence, the exemplar for the data item indexed i is chosen as $e(x_i) = x_k$ where k satisfies the criterion:

$$k = \arg\max_k\{a(i,k) + r(i,k)\}. \tag{8}$$

The algorithm is halted after a fixed number of iterations or after the exemplars do not change for a given number of iterations. Affinity propagation is not a universally efficient data clustering method. Firstly, if the desirable number of clusters K is small, then the combinatorial problem can be tackled by brute force (considering all N^K possible solutions). Secondly, and most importantly, affinity propagation suffers from quadratic computational complexity in the number of data items N. This hinders its direct use in large-scale applications. To reduce the computational complexity of affinity propagation, in this paper we proposed an algorithm based on similarity propagation based clustering. We split the entire dataset into subsets of graphs, and then perform exemplar extraction on each subset.

3 Mining Exemplars Using Three Stage Clustering

Suppose that we have obtained a large graph set $\mathbb{G} = \{G_l,\ l = 1, 2, ..., N\}$, extracted from an image corpus. In this section, we demonstrate how to efficiently obtain object specific exemplars from such datasets. Our method involves four main steps:

- Step 1 Train the RSOM tree clustering of SIFT descriptors;
- Step 2 Obtain the KNNG for each graph in $\mathbb{G}$;
- Step 3 Obtain the FTOG in a weakly supervised manner using SPGC;
- Step 4 Detect exemplars for each FTOG individually using affinity propagation.

Steps 1-3 follow directly the work reported in [4]. In Step 4, we firstly apply affinity propagation to each FTOG and rank all graphs according to $\arg\max_k\{a(i,k)+r(i,k)\}$. We select J exemplars according to Equation 8 and check whether these J exemplars form an FTOG. If not, we select the $J' > J$ top ranked graphs such that these selected graphs form an FTOG. In this way, we simplify each FTOG M_l to $M_l^{'}$ which is constructed from a group of exemplar graphs G_j^l, $j = 1, 2, ..., ||M_l^{'}||$ and their similarity relationships.

For each feature point as a node k of an exemplar graph G^l_j in the FTOG M'_l, we count the frequency n^l_{jk} of being matched by its nearest neighbors. The simplified FTOG M'_l depends on M_l and R_0 and is a more compact representation.

In this way a compact representation of each FTOG is obtained. The set of compact FTOGs forms the learned model of the object of interest. Once the object model is trained, for a test graph G_l, we can obtain $K\ \{G_l\}$ and use a weighted voting method based on k-nearest neighbor graphs for recognition, where the similarity measure $R(G_l, G_q)$ is used as the weight. We then use the well known F-measure in order to evaluate the recognition performance. A high value of the F-measure f means that both high recall and high precision are achieved. The ideal result is $f = 1$.

4 Experimental Results

4.1 Datasets

We have collected 53,536 images as a training set. This dataset spans more than 500 objects, including some human faces and natural scenes. The image corpus is composed as follows: there are a) 3600 images of 50 objects from COIL 100, labeled A1~A50; b) 161 images of 8 objects used in [18], labeled B1 to B8; c) 20000 images of 10 objects collected in our own lab., labeled as C1 to C10. For each of the objects in C1 to C9, we have collected 1500 images which traverse large variations of imaging conditions, and similarly 6500 images for C10; d) 29875 unlabeled images from many other standard datasets, e.g. Caltech101 [13], PASCAL VOC'07 [12] and Google image, spanning over 450 objects and used as negative samples. For simplicity, the 4 data sets are denoted as A, B, C and D. The objects in Figure 1,2 and 3 are numbered from left to right and then from top to bottom as shown in the corresponding figures. We take 68 images as object examples for recognition, and these are identified as Object 1 to Object 68 in Figure 1, 2 and 3.

For each of these images, we extract ranked SIFT features, using the method presented in [5], of which at most 40 highly ranked features are selected to construct a graph.

4.2 Clustering Results

From the training data we have obtained an RSOM clustering tree with 25334 leaf nodes using the method described in [2]. The method was implemented using Matlab 7.2 and run on a 2.14GHz computer with 2G RAM. In the incrementally training process, we have obtained $K\ \{G_l\}$ for each of the graphs.

Following RSOM tree clustering, we individually obtain FTOG's for the above 68 labeled object classes using the similarity propagation based graph clustering method presented in Section 2. The object clustering results for the 68 object problem are shown in Figure 5. For most of the objects sampled under controlled imaging conditions, ideal performance has been achieved. For 35 objects in COIL 100, 35 models are individually clustered with total unit recall and precision in one FTOG. For 13 objects, 6 FTOGs are obtained. Each group of objects in Figure 6 (A)(B)(C)(D) are actually identical in shape but color. Since it only uses gray scale information in SIFT, our method fails in this case.

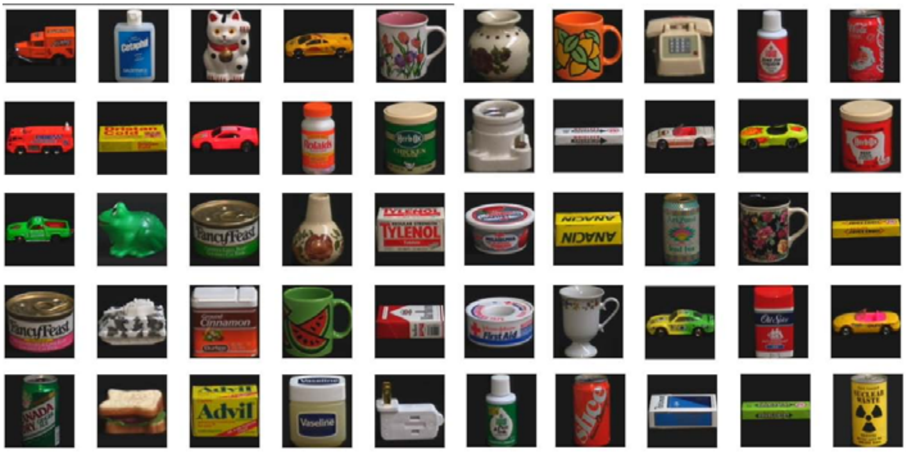


Fig. 1. 50 objects in Coil 100

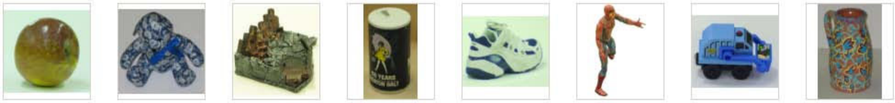


Fig. 2. 8 objects in[18]

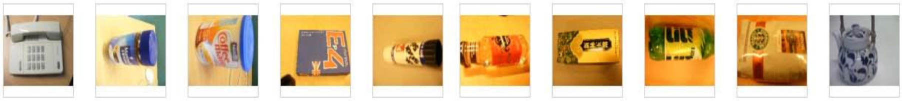


Fig. 3. 10 objects collected by the authors

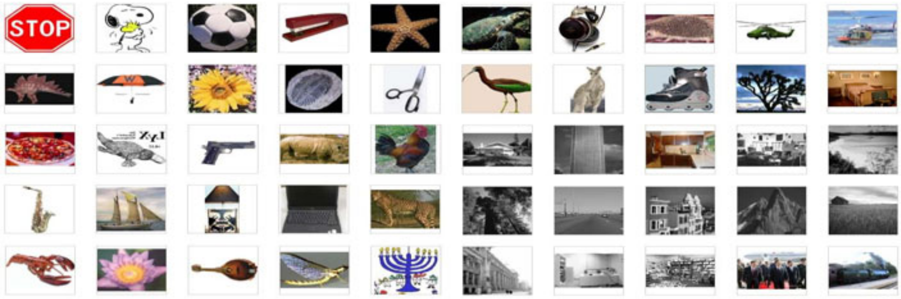
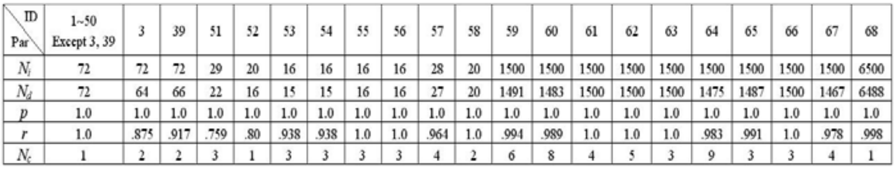


Fig. 4. Unlabeled sample images



ID Par	1~50 Except 3, 39	3	39	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68
N_i	72	72	72	29	20	16	16	16	16	28	20	1500	1500	1500	1500	1500	1500	1500	1500	1500	6500
N_d	72	64	66	22	16	15	15	16	16	27	20	1491	1483	1500	1500	1500	1475	1487	1500	1467	6488
p	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
r	1.0	.875	.917	.759	.80	.938	.938	1.0	1.0	.964	1.0	.994	.989	1.0	1.0	1.0	.983	.991	1.0	.978	.998
N_c	1	2	2	3	1	3	3	3	3	4	2	6	8	4	5	3	9	3	3	4	1

Fig. 5. Results of object clustering using similarity based graph clustering. In the above table, ID is the Object ID; N_i is the number of the initial images of an object; N_d is the number of images clustered by using similarity based graph clustering; N_d^+ is the number of correctly clustered images; p is the precision defined as N_d^+/N_d; r is recall defined as N_d^+/N_i. N_c is the number of clusters for each object.

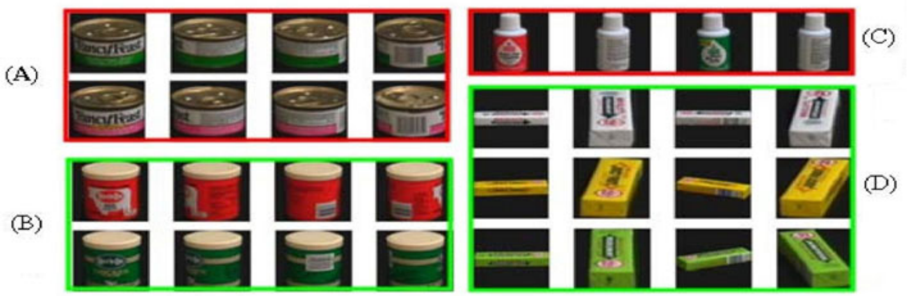


Fig. 6. 6 groups of objects are overlapping-clustered into 6 clusters

We hence regard these objects in the four groups as being correctly clustered according to shape.

Unfortunately, in most practical situations, the images of an object are likely to be obtained with large variations of imaging conditions and are more likely to be clustered into several FTOGs. As a result, each object gives rise to multiple clusters. For objects 51 to 58 there are more than 30 images with large variations in viewing conditions, and the images are not representative enough to perform ideal recognition. However, for objects 59 to 68, the images clustered together are sufficient to form an effective object model which can be used for recognition. For object 68, since there are thousands of images, the different views form a single cluster.

For each FTOG, the exemplar graphs are obtained individually using affinity propagation. The percentage of the exemplar graphs for each object are shown in Figure 7.

4.3 Recognition Test Results

We also collect images of Objects 1 to 68 for recognition experiments. For each of objects A1 to A50 (i.e. those contained in the COIL database), we synthesis 6 images by adding a mixture of salt and pepper noise, speckle noise and Gaussian noise to their original image. The variances of the noise processes are randomly set to 0.03, 0.04 or 0.05. In total, 21600 images of the 50 objects are obtained. For objects B1 to B8, we

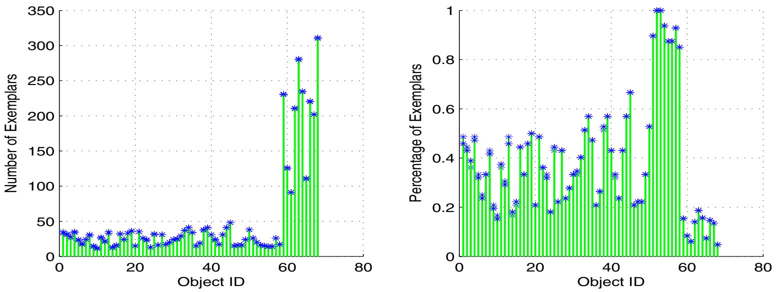


Fig. 7. The numbers and percentages of exemplars obtained by using SPCG and affinity propagation for the objects of interest

Table 1. F-measure f for given test set of Object 1~68

ID	f	ID	f	ID	f	ID	f	ID	f	ID	f	ID	f	ID	f	ID	f	ID	f
1	**.997**	2	**.965**	3	**1.0**	4	**.984**	5	**1.0**	6	**1.0**	7	**1.0**	8	**.990**	9	**1.**	10	**1.0**
11	**.983**	12	**1.0**	13	**.988**	14	**1.0**	15	**1.0**	16	**1.0**	17	**1.0**	18	**.982**	19	**.981**	20	**1.0**
21	**.986**	22	**1.0**	23	**1.**	24	**1.0**	25	**1.0**	26	**1.0**	27	**.993**	28	**1.0**	29	**1.**	30	**1.0**
31	**1.0**	32	**1.0**	33	**.994**	34	**1.0**	35	**1.0**	36	**1.0**	37	**1.0**	38	**.995**	39	**.980**	40	**1.0**
41	**1.0**	42	**0.988**	43	**1.0**	44	**.995**	45	**.941**	46	**1.0**	47	**1.0**	48	**1.0**	49	**1.0**	50	**.989**
51	**.625**	52	**1.0**	53	**1.0**	54	**1.0**	55	**.714**	56	**1.0**	57	**.954**	58	**1.0**	59	**.992**	60	**.998**
51	***.625***	*52*	***1.0***	*53*	***1.0***	*54*	***1.0***	*55*	***.714***	*56*	***1.0***	*57*	***1.0***	*58*	**1.0**				
61	**.992**	62	**.997**	63	**.993**	64	**.983**	65	**.991**	66	**.989**	67	**.994**	68	**1.0**				

have manually obtained all 78 ROI's (ROI: region of interest), each of which includes one object of interest, from 51 test images presented in [18]. For each of the objects C1 to C10, we collect 500 images under similar but not identical imaging conditions. For each of these images, we also extract ranked SIFT features to construct a graph, using the same method presented in [5].

Using the trained model, these object images are recognized according to the following Steps:

1) Obtaining the KNNG for each graph;

2) Recognition by making using a majority voting strategy weighted by the similarities of the corresponding KNNG.

Suppose N_{TP} is the number of correctly recognized instances for an object of interest, N_{FP} is the number of instances incorrectly recognized as the object of interest, N_P is the number of instances belonging to the object of interest. Then the F-measure is defined as follows:

$$f = \frac{2}{1/recall + 1/precision}. \tag{9}$$

where $recall = \frac{N_{TP}}{N_P}$, $precision = \frac{N_{TP}}{N_{TP}+N_{FP}}$. The F-measures for the recognition test for Object 1 to Object 68 are shown in Table 1. It is interesting to note that the test recognition performance of Object B1 to B8 is very close to that obtained in [18] when only SIFT features are used (corresponding to the 8 objects marked in magenta in Table 1. However, we emphasize that our results are obtained with large negative sample sets.

5 Conclusion

This paper has described a framework for learning recognition oriented exemplar models from a large corpus of multi-view images. Our model is a comprehensive integration of the global and local information contained in the local features from different views. The exemplars are extracted in a three-stage clustering process. First, RSOM tree clustering is used to incrementally cluster a large corpus of local invariant feature descriptors. Using RSOM, the K nearest neighbor graphs of each graph can be efficiently obtained without linear search. Second, the similarity propagation based graph clustering method is used to cluster the graph instances of a specific object with high precision. Such a graph cluster is termed an FTOG. Third, each FTOG is then subjected to affinity propagation to obtain the exemplars for a single FTOG. For each additional test graph, the recognition decision is made according to its nearest exemplar. Experiments demonstrate high performance in terms of efficiency, scalability and recognition.

References

1. Frey, B., Dueck, D.: Clustering by Passing Messages Between Data Points. Science 315(5814), 972–976 (2007)
2. Xia, S.P., Liu, J.J., Yuan, Z.T., Yu, H., Zhang, L.F., Yu., W.X.: Cluster-Computer Based Incremental and Distributed RSOM Data-Clustering. ACTA Electronica sinica. 35(3), 385–391 (2007)
3. Xia, S.P., Hancock, E.R.: 3D Object Recognition Using Hyper-Graphs and Ranked Local Invariant Features. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 117–126. Springer, Heidelberg (2008)
4. Xia, S.P., Hancock, E.R.: Pairwise Similarity Propagation Based Graph Clustering for Scalable Object Indexing and Retrieval. In: Torsello, A., Escolano, F., Brun, L. (eds.) GbRPR 2009. LNCS, vol. 5534, pp. 184–194. Springer, Heidelberg (2009)
5. Xia, S.P., Ren, P., Hancock, E.R.: Ranking the Local Invariant Features for the Robust Visual Saliencies. In: ICPR (2008)
6. Xia, S.P., Hancock, E.R.: Learning Class Specific Hypergraphs. In: Foggia, P., Sansone, C., Vento, M. (eds.) Image Analysis and Processing – ICIAP 2009. LNCS, vol. 5716, pp. 269–277. Springer, Heidelberg (2009)
7. Xia, S.P., Hancock, E.R.: Graph-Based Object Class Discovery. In: Jiang, X., Petkov, N. (eds.) Computer Analysis of Images and Patterns. LNCS, vol. 5702, pp. 385–393. Springer, Heidelberg (2009)
8. Bay, H., Tuytelaars, T., Gool, L.V.: SURF: Speeded Up Robust Features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
9. Lowe, D.G.: Distinctive image features from scale-invariant key points. IJCV 60(2), 91–110 (2004)

10. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: ICCV (2007)
11. Chung, F.: Spectral graph theory. American Mathematical Society, Providence (1997)
12. Everingham, M., Gool, L.V., Williams, C., Winn, J., Zisserman, A.: Overview and results of classification challenge. In: The PASCAL VOC 2007 Challenge Workshop, in conj. with ICCV (2007)
13. Li, F.F., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. CVPR 2, 524–531 (2005)
14. Kadir, T., Brady, M., Zisserman, A.: An Invariant Method for Selecting Salient Regions in Images. In: Proc. Eighth ECCV., vol. 1(1), pp. 345–457 (2004)
15. Mikolajczyk, K., Schmid, C.: A Performance Evaluation of Local Descriptors. PAMI 27(10), 1615–1639 (2005)
16. Philbin, J., Chum, O., Isard, M., Sivic, J., Zissermans, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)
17. Philbin, J., Sivic, J., Zisserman, A.: Geometric LDA: A Generative Model for Particular Object Discovery. BMVC (2008)
18. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints. IJCV 66(3), 231–259 (2006)
19. Schonemann, P.H.: A generalized solution of the orthogonal Procrustes problem. Psychometrika 31(3), 1–10 (1966)
20. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: ICCV, pp. 1470–1477 (2003)

Guided Informative Image Partitioning

Nathan Brewer[1,2], Nianjun Liu[2], and Lei Wang[1]

[1] College of Engineering and Computer Science, Australian National University,
Crnr North and Daley Roads, 0200, Canberra, Australia
[2] Canberra Research Laboratory, NICTA
Tower A, 7 London Circuit, 2600, Canberra, Australia
{nathan.brewer,lei.wang}@anu.edu.au, nianjun.liu@nicta.com.au

Abstract. Image partitioning separates an image into multiple visually and semantically homogeneous regions, providing a summary of visual content. Knowing that human observers focus on interesting objects or regions when interpreting a scene, and envisioning the usefulness of this focus in many computer vision tasks, this paper develops a user-attention adaptive image partitioning approach. Given a set of pairs of oversegments labeled by a user as "should be merged" or "should not be merged", the proposed approach produces a fine partitioning in user defined interesting areas, to retain interesting information, and a coarser partitioning in other regions to provide a parsimonious representation. To achieve this, a novel Markov Random Field (MRF) model is used to optimally infer the relationship ("merge" or "not merge") among oversegment pairs, by using the graph nodes to describe the relationship between pairs. By training an SVM classifier to provide the data term, a graph-cut algorithm is employed to infer the best MRF configuration. We discuss the difficulty in translating this configuration back to an image labelling, and develop a non-trivial post-processing to refine the configuration further. Experimental verification on benchmark data sets demonstrates the effectiveness of the proposed approach.

1 Introduction

As human observers, when we are observe the world around us there are things in our field of view that interest us more than others. We subconsciously pay more attention to these areas, taking in more information about them than their surroundings. In computer vision, partitioning an image into multiple visually and semantically homogeneous regions is an important step in scene understanding. It provides a useful summary of visual content and allows a complex visual recognition process to be decomposed to region-level subtasks. While image partitioning has been well studied and widely applied in many vision problems, existing segmentation techniques can not reflect this type of human focussing behavior. In this paper, we present a method for partitioning an image such that some user-defined interesting object(s) are preserved at higher resolution than uninteresting objects.

The idea that the human brain devotes more attention to some areas in the visual field has support from the field of psychology [1]. Such research has likened

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 202–212, 2010.

perception to a spotlight, illuminating different players on a stage as attention is focussed on them. The rest of the stage remains visible, but more details stand out about the interesting actor. The partitioning that we propose is a good analogue of this model of human visual attention. It is important to note that we neither construct or utilize a specific attention model in our approach. Instead, we attempt to produce an informative image partitioning which expands on user input to produce a result which reflects this information by detailing interesting areas with a finer partitioning (placing more, smaller-sized segments) while abstracting uninteresting areas with coarser partitioning (placing fewer, larger-sized segments).

In this paper, we present a new framework for incorporating user-defined interestingness information in order to produce such an informative partitioning. Instead of segmenting an image at the object level, we split an image into a large number of oversegments, small-sized image patches containing similar visual information. A user expresses the relative interestingness of parts of the image by labelling a set of example pairs of oversegments as “should be merged” or “should not be merged”, that is, as side information. Importantly, it is up to the user to define which areas should not be merged, allowing them to dictate objects that may otherwise be seen as background as interesting objects, or to define the focus of an image as uninteresting.

From this side information, an SVM classifier is learned to describe the relationship between pairs of oversegments. To take spatial context in account, an MRF model is employed to optimally infer the relationship between neighboring oversegments as “merge” or “not”. Unlike the standard form, the nodes of the graph in our work denote the relationship of oversegment pairs rather than oversegments themselves, essentially producing an inversion of the edges and nodes in a typical image MRF. The graph-cut algorithm is employed to infer the best MRF configuration. However, we show that the result of a graph-cut with the Potts model, or any other local smoothness criterion cannot be readily used, and a non-trivial post-processing has to be developed to refine the above configuration further, reducing the impact of classification errors.

This distribution of differently sized segments across an image based on the interestingness of their contents provides a novel method for allocating resources to an image for whatever purpose the user has, particularly for classification and compression tasks. Our supervised approach allows us to learn a high quality system for producing this segmentation from a limited amount of side information. The modification we make to the standard MRF image representation to incorporate spatial relationships into our framework has potential application in many areas outside of this paper. We present experimental validation of the ideas presented in this paper, achieving excellent performance across a wide range of image classes.

Clearly, this objective can be seen as an image segmentation problem. However, existing segmentation methods are unable to truly reflect the interest focussing behavior that we describe. Unsupervised segmentation methods such as [2,3] typically generate a segmentation of an image by identifying areas with similar colour or texture, or by attempting to locate the most likely location of the boundary between different objects. These methods do not take the user's interest into account, and are unable to accommodate user input that expresses this interest. Supervised segmentation methods classify areas of an image based on a model learned from a large amount of training data into a set of pre-defined object classes. However, they are focused on how to accurately infer the class label of each pixel to achieve the best segmentation, rather than the resolution of this segmentation. Intuitively, we can choose to consider user attention for each segmented part *after* segmenting the whole image with an existing method. However, this is not efficient because i) Initial segmentation ignores user input when generating the segmentation, and as such much of the effort is wasted elegantly segmenting non-interesting areas; ii) it forces user attention to be expressed at the level of segmented areas; iii) the number of segments generally has to be pre defined. Alternatively, our technique can be seen as a region merging algorithm, such as that in [4]. However, existing region merging methods are typically unsupervised or take user input in the form of segment biases. They also do not allow for regions to be specifically not merged despite similarity, which is required to produce the partitioning we desire.

2 Generating an Informative Image Partitioning

2.1 Use of Oversegments and Oversegment Features

In order to reduce the size of the graph that we process, and to simplify the acquisition of human input, we use an oversegmentation of the input image in our process. The method used to generate this oversegmentation is largely arbitrary, however it is important for the process that the oversegments produced contain only a small-sized image patch with sufficiently homogeneous visual content.

Superpixel methods are a good way of generating the required oversegments, as they attempt to maximally follow the criteria that we have defined. Mori's superpixel generation method [5], based on the Normalized Cut algorithm [3], produces similarly sized typically homogeneous segments across an image. Superpixels are not constrained to any particular distribution across the image, and this lends itself well to ensuring that multiple image objects do not fall into the same oversegment.

The Superpixel Lattice method of Moore et al [6] provides a viable alternative to Mori's superpixels in our work, and the two methods can be used interchangeably. We did not investigate less homogeneous oversegmentation methods, such as the work of Felzenszwalb and Huttenlocher [2] and the classic watershed image transform [7], for generating our input, as it is difficult to control the number of segments obtained from these methods and they tend to produce very differently sized segments within the image.

We have used a simple feature set for classification in this paper. We use colour histograms in the CIE Lab colour space and a texture set. One ten bin, equal bin width histogram is generated for each colour channel. Each histogram is normalised by the size of the superpixel from which it is extracted to remove the effect of oversegment size in comparison. We utilize a set of eight texture features, derived using the method presented by Varma and Zisserman in [8]. The average response of each pixel in an oversegment to each of these eight features is taken as a description of the texture of the oversegment. A feature vector **c** for each oversegment is constructed from this set of visual information.

2.2 Acquiring User Input

User input is taken in the form of pairs of oversegments and an instruction, either "Merge" or "Do Not Merge" these two oversegments. In this way, we do not require any knowledge of the number of classes of object in the image, nor do we require the user to directly assign labels to any part of the image. Input data can represent one of four things: In-class merges (merging similar oversegments), In-class no-merges (marking an object as interesting), interclass no-merges (Identifying that two oversegments belong to different objects) and finally interclass merges (Identifying that two oversegments belong to different objects, but the user would still like them to be merged).

The amount of user input required depends largely on the complexity of the image. Simple problems can be solved with as few as 20 pairs, while more complex scenes can take up to 100 pairs to build a robust model. In class merges, interclass merges and in class no merges can be acquired with a brush input device, but interclass no merges require the user to specify the specific oversegments which should not be merged, as brush input in this case is unreliable.

3 MRF-Based Region Merging

3.1 MRF Model with Edge-to-Vertex Transformation

It is very common in the literature to represent an image as a Markov Random Field [9]. This MRF typically represents each pixel (or oversegment) as a vertex $\mathcal{V}$, with edges $\mathcal{E}$ between adjacent pixels, resulting in an MRF described by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Most applications seek to infer the optimal configuration of this MRF, that is, assigning class labels to each vertex by considering the spatial smoothness. In contrast, *our problem is not going to assign a class label directly to each vertex. Instead, we want to optimally infer the label ("merge" or "not merge") for each edge.* To achieve our goal, we have to conduct an edge-to-vertex transformation, as shown in Figure 1.

Edge-to-vertex transformation has been used in graph and network analysis. For example, the work in [10] applies this method to analyze a city traffic network, in which roads are mapped to vertexes and intersections to edges between vertexes. This transformation is taken by our approach to map the edges in the preceding MRF graph to the vertexes of a new graph. By doing so, each vertex in the new graph denotes the relationship between superpixels, and its label is either "merge" or "not merge". Formally, we define a new graph $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$ with vertices $\mathcal{V}^*$ (corresponding to the edge $\mathcal{E}$ in the preceding graph) and edges

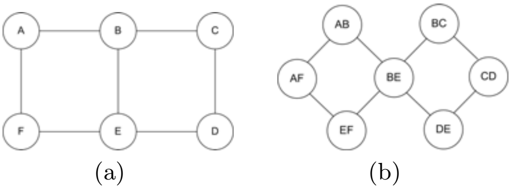


Fig. 1. (a) Initial MRF representation of an image $\mathcal{G}$. (b) The transformed graph $\mathcal{G}^*$ in which we represent edges from $\mathcal{G}$ as nodes.

$\mathcal{E}^*$. α_i^* ($\alpha_i^* \in \{0,1\}$) is the label assigned to $\mathcal{V}_i^*$. Each vertex $\mathcal{V}_i^*$ retains the visual information c_i of the two vertices that it connects as an edge in the old graph. This formulation allows us to simplify the problem into a simple binary labelling, which allows us to employ the standard MRF inference method to find the optimal labels. Because semantic concepts of an image are formed by groups of oversegments and users often pay attention to meaningful objects, adjacent oversegments will often share the same merging (or non-merging) labels. This implies that the distribution of merge and non-merge labels will be distributed in a smooth fashion across the image, except at boundaries among interesting and uninteresting areas.

With the above MRF model, we represent our inference problem as minimisation of the cost function in equation (1), taking the form:

$$U(\alpha^*;\mathbf{c}^*) = \lambda \sum_{i\in\mathcal{V}^*} U_1(\alpha_i^*;c_i^*) + \sum_{ij\in\mathcal{E}^*} U_2(c_i^*;c_j^*)\delta[\alpha_i^* \neq \alpha_j^*] \tag{1}$$

Its data term U_1 represents the cost of assigning vertex i to $\alpha_i = 0$ (do not merge) or $\alpha_i = 1$ (do merge). Its local smoothing term U_2 represents the cost associated with assigning different labels to adjacent nodes. Derivation of the data term used is discussed in section 3.2. Using this form for our cost function allows us to use a graph cut algorithm to find an optimal labelling. Specifically, we use the max flow algorithm described in [11] to efficiently find a minimum cut over a graph where edge weights are defined as:

$$z_{ij} = U_2(c_i^*, c_j^*) = \frac{1}{||c_i^* - c_j^*|| + 1}; z_{i0} = \lambda \cdot U_1(\alpha_i^* = 0; c_i^*); z_{i1} = \lambda \cdot U_1(\alpha_i^* = 1; c_i^*) \tag{2}$$

where z_{i1} is the cost of assigning a node to merge, z_{i0} is the cost of not merging and z_{ij} is calculated based on the similarity between nodes. As shown, we reflect this smoothness by a simple Potts model, assigning the penalty for assigning different labels to adjacent nodes based on the difference between the features in the two nodes. Assigning smoothness in this way allows us to reduce the cost of assigning different labels when there is a significant difference between two nodes, as expected.

3.2 Learning the Data Term with a Support Vector Machine

The data term from Equation (1) is computed as follows. Let c^*_{ia} and c^*_{ib} denote the visual feature vectors of the two vertices that $\mathcal{V}^*_i$ connects when it is an edge in the initial graph. The problem is to estimate the possibility that $\mathcal{V}^*_i$ is labeled as 0 or 1 solely based on c^*_{ia} and c^*_{ib}. Since a user has labeled some example pairs of superpixels as merged or not, calculating the data term can be solved by a learning task.

An SVM classifier is trained as follows. Suppose a user provides a set of example pairs of superpixels. We stack two superpixels $\mathbf{c}^*_{ia}$ and $\mathbf{c}^*_{ib}$ in each pair as a long vector $\mathbf{c}^*_i$. Thus, a training set is obtained as $\{\mathbf{c}^*_1, \alpha_1\}, \{\mathbf{c}^*_2, \alpha_2\}, \cdots, \{\mathbf{c}^*_m, \alpha_m\}$, where $\mathbf{c}^*_i$ is the training vector and α_i is the corresponding label, 0 or 1. Note that whether two superpixels are to be merged or not is independent of its order presented in $\mathbf{c}^*_i$. Hence, for each $\mathbf{c}^*_i$ $(i = 1, \cdots, m)$, we can generate a "shadow" training vector $\overline{\mathbf{c}}^*_i$ in which the order of $\mathbf{c}^*_{ia}$ and $\mathbf{c}^*_{ib}$ is switched and its label remains. Thus, there are $2m$ training samples to be used for SVM training. We use a SVM classifier with a Gaussian RBF kernel of the form: $k(c^*_i, c^*_j) = \exp(-\gamma||c^*_i - c^*_j||^2)$, where γ is a nonnegative parameter which needs to be tuned for our specific application. A misclassification cost parameter C also needs to be tuned to produce optimal classification results. We use a grid search method with ten-fold cross validation on the training set to find optimal values for these parameters.

Feature components in the training set are scaled to the range (0,1) based on the full range of values that are present in the image, not only the training set. Using the popular LibSVM tool [12], we are able to directly determine (with the switch -b in LibSVM) a probability for each vertex $\mathcal{V}^*_i$ being assigned label $\alpha_i = 0$. We define this value as $\Theta[f_{SVM}(c^*_i)]$. From this, we can obtain $U_1(\alpha^*_i = 1, c^*_i) = \Theta[f_{SVM}(c^*_i)]$ and $U_1(\alpha^*_i = 0, c^*_i) = 1 - U_1(\alpha^*_i = 1, c^*_i)$.

3.3 Inverse Edge-to-Node Transformation

After manipulating the graph to incorporate side information in this fashion, we aim to determine whether adjacent nodes in the original graph $\mathcal{G}$ should be merged or not, based on the label assigned in the transformed graph $\mathcal{G}^*$. Clearly, to realise this, we must convert from the node labelling in $\mathcal{G}^*$ back to a labelling on the initial graph $\mathcal{G}$. First, each oversegment is assigned a unique label. Then, we are able to sequentially apply the merges described by $\mathcal{G}^*$, eliminating labels as we merge oversegments. Once all merges have been performed, this labelling corresponds directly to the partitioning of the image.

Unfortunately, the merging behavior described by $\mathcal{G}^*$ cannot always be directly applied, as doing so can produce unresolvable ambiguities in the labelling, in which two nodes should both be merged and not merged.

As such, an additional translation between the edge graph and a partitioning of the image is required. A global translation method is described here with two possible solutions: merge a set of pairs that should otherwise not be merged, which converts some zeros into ones, or split a set of merges by converting ones in the graph into zeros until there are no conflicts.

Applying the first solution is quite simple. One needs only to apply all merges present in $\mathcal{G}^*$. Given the sequential merging of labels that we conduct to generate our partitioning, this will resolve conflicts by allowing merges that our graph cut classifier would otherwise prevent. The second solution prioritises the "do not merge" instruction from $\mathcal{G}^*$ over the merge instruction. This indicates that if $\alpha_i^* = 0$ in the transformed graph, then under no circumstances should the oversegments represented by vertex $\mathcal{V}_i^*$ be present in a single partition, even if there exists a path that would otherwise allow these segments to merge. The location of such splits can be arbitrarily decided, but we are also able to make use of a hierarchical clustering type approach to produce a partitioning in more structured fashion.

3.4 Hierarchical Clustering

Hierarchical clustering is a commonly used clustering method in which the best available merging of two existing clusters is found and applied repetitively, either until a set number of clusters is reached or the best merge passes below some threshold of quality. This lends itself naturally to our application, allowing us to organically grow the partitioning to produce more consistent results.

Cluster Distance and Constrained Hierarchical Clustering. While we do not have a direct measure of the distance between oversegments, the SVM model learned above provides an estimate of the likelihood that any two oversegments should be merged. This can be used to determine the pair of oversegments most likely to be merged, and gives us a non-metric distance measure.

As we aim to preserve the "do not merge" instructions from our edge labelling, we require that any pair of oversegments marked this way remain in separate clusters. This can be taken into account by adding a significant cost to the merging of any two clusters that would break this restriction. Furthermore, we require a spatial restriction such that oversegments and clusters that are not adjacent to one another cannot be merged. This is done by applying a similar cost to non-neighboring clusters as to disjoint clusters.

Constrained Clustering Algorithm. To perform the hierarchical clustering, we first use our SVM to determine the distance between each pair of oversegments in our image. We store these distances as elements in a symmetrical $n \times n$ *distance* matrix, D, where n is the number of oversegments. We then construct from the edge graph a binary *disjoint* matrix X, which is set to one at X_{ij} if oversegments i and j have a "do not merge" instruction between them. Finally, a binary *neighborhood* matrix N is constructed with element $N_{ij} = 1$ if i and j share an edge, and zero otherwise.

We initialise each cluster to contain a single oversegment, then perform hierarchical clustering using a single-link approach, as described in Algorithm 1. The single-link methodology is used here rather than the full-link methodology based on experimental performance. This is likely a result of our non-metric distance function together with the spatial and disjoint constraints we apply.

```
Data: Matrices $D$, $X$ and $N$ of size $n \times n$(see text)
Result: Set of Clusters
set number of clusters $c$ to $n$
set cluster contents $c_i(s)$ to cluster number
while more possible merges exist do
    set high value for $bestCdist$ for $i \leftarrow 1$ to $c$ do
        for $j \leftarrow i$ to $c$ do
            foreach element $a$ in cluster $i$ and element $b$ in cluster $j$ do
                if (there exists $X_{ab} = 1$) or ($\forall N_{ab} = 0$) then
                    Break
                else
                    minCdist = min $D_{ab}$
                end
            end
            if $minCdist < bestCdist$ then $bestCdist = minCdist$, store $ij$
        end
    end
    if $bestCdist$ has changed then
        merge clusters $ij$
    else
        no possible merges
    end
end
```

Algorithm 1. The constrained Hierarchical Clustering Algorithm

4 Experimental Results

4.1 Experimental Setup

For each of the experiments detailed below, we make use of a subset of images from the MSRC data set [13]. We use first 15 entries in the livestock, tree, building, cow and aeroplane subsets of the data set for this test, resulting a total data set of 75 images. All images are segmented using Mori's method [5] to generate approximately 1000 oversegments for each image, and we train a separate model for each image using between 25 and 50 positive and negative training pairs for each image, which corresponds to between 1 and 3% of all neighboring oversegment pairs in the image. The number of training pairs used depended on the complexity of the image. Void areas from the data set are always treated as uninteresting. Some images from the first livestock category were removed from these calculations, as they contained only uninteresting areas. The same set of training pairs is used in each of the experiments presented.

4.2 Focussing Performance

Primarily, we are interested in determining how effective our merging algorithm is at successfully reducing the number of segments in uninteresting regions while leaving a significant number of segments in areas of interest. To do this, we have defined one class of object as the object of interest, as defined in Table 1 and we determine the number of segments present within these objects and within uninteresting objects both before and after applying our method. As we are interested in the amount of merging that takes place in these areas, we have presented the difference between the average number of partitions in these areas as a percentage of the initial number of oversegments in the same area.

Table 1. The partition focussing performance of our approach. We show the average reduction in the number of partitions within interesting and uninteresting objects after applying our method under several resolutions of the loop condition for five different image classes. Method A prioritises merging, Method B prioritises not merging and HC is a Hierarchical approach to Method B.

	Interesting Object (Average% Reduction)			Uninteresting Object (Average% Reduction)		
Interesting Object	Method A	Method B	HC	Method A	Method B	HC
Livestock	33.8%	32.0%	30.5%	96.7%	95.7%	95.4%
Trees	16.0%	15.2%	14.0%	95.5%	94.3%	93.9%
Buildings	18.0%	17.0%	16.2%	94.7%	93.8%	93.4%
Aeroplanes	27.7%	25.6%	23.7%	95.7%	94.7%	93.6%
Cows	28.9%	27.0%	25.3%	96.2%	94.7%	94.3%
Overall	24.0%	22.5%	21.1%	95.7%	94.6%	94.0%

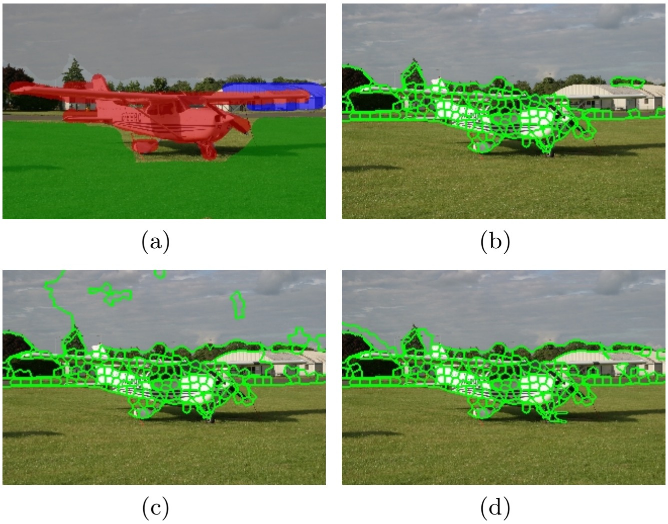


Fig. 2. (a) Input image with ground truth label overlay. (b) Partitioning generated from Method A. (c) Partitioning generated from method B. (d) Partitioning generated from Hierarchical Clustering. The green lines indicate partition boundaries. As can be seen, the partitioning in (b) has allowed excessive merging to the right of screen, combining trees and building with sky. This has been fixed in (c), but with the introduction of unwanted partitions. (d) produces a higher-quality partitioning than (c).

As can be seen from these results, we show a significantly higher reduction in uninteresting areas than in interesting ones under all translation methods. These results demonstrate the effectiveness of our approach at producing a focussed partitioning of an input image given some user information.

In addition to observing the quality of the partition focusing behaviour, we also investigated the amount of segmentation error associated with each of the partitions generated. To calculate this, we first find the partitions that contain more than one class of object from the ground truth, and then we find the number of pixels that would be classified as a different class if these partitions were to be classified according to the predominant label. As expected, we find that there is some increase in the segmentation error after merging, ranging between less than 0.5% and 7% of the image. The loop resolution method has a large impact on this error, which shows that there is a tradeoff between reducing the number of partitions placed over uninteresting areas and reducing the segmentation error. Figure 2 shows a sample initial image, ground truth label and the image partitions produced by allowing merges, disallowing merges, and with various hierarchical clustering schemes.

Our results are slightly skewed due to disparities between the human labeled ground truth information and the true location of the boundary between objects, which typically results in a larger reduction of the number of partitions in interesting areas being reported than is actually observed. This affects both the focussing effect and the segmentation error calculations and is particularly prevalent in the livestock and aeroplanes subsets, as the interesting region in ground truth is noticeably larger than the actual interesting object. Additionally, when more than one uninteresting area was present in the image, our method reported less reduction in these areas as it retains separation between these objects.

The data presented above also allows us to compare the effectiveness of each of the methods that we have used to resolve ambiguities in the instructions between nodes. We are also able to see the effect of the loop resolution method when translating from the edge graph to image labels. Method A, which favors merges, produces slightly fewer partitions in regions of interest, but significantly fewer in uninteresting regions than Method B, which favors keeping objects separate. The hierarchical clustering approach mimics the behaviour of Method B, but provide an obvious improvement in partitioning performance.

5 Conclusions

We have presented in this paper a framework for mimicking the top-down 'spotlight focussing' aspect of the human visual system in digital images, together with a new way of representing an image using a Markov Random Field to incorporate side information. Under this framework, we demonstrate that we are able to produce a high-quality focussed partitioning of an input image given only a relatively small amount of side information.

References

1. Weichselgartner, E., Sperling, G.: Dynamics of automatic and controlled visual attention. Science 238, 778–780 (1987)
2. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. Int. J. Comput. Vision 59(2), 167–181 (2004)
3. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 888–905 (2000)

4. Nock, R., Nielsen, F.: Semi-supervised statistical region refinement for color image segmentation 38(6), 835–846 (2005)
5. Mori, G.: Guiding model search using segmentation. In: ICCV 2005, vol. 2, pp. 1417–1423 (October 2005)
6. Moore, A., Prince, S., Warrell, J., Mohammed, U., Jones, G.: Superpixel lattices. In: CVPR 2008, June 2008, pp. 1–8 (2008)
7. Meyer, F., Beucher, S.: Morphological segmentation 1(1), 21–46 (September 1990)
8. Varma, M., Zisserman, A.: Classifying images of materials: Achieving viewpoint and illumination independence. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 255–271. Springer, Heidelberg (2002)
9. Li, S.Z.: Markov random field modeling in image analysis. Springer, New York (2001)
10. Rosvall, M., Trusina, A., Minnhagen, P., Sneppen, K.: Networks and cities: An information perspective. Physical Review Letters 94 (January 2005)
11. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. Pattern Anal. Mach. Intell. 26(9), 1124–1137 (2004)
12. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001) Software available at, `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
13. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)

Visual Alphabets on Different Levels of Abstraction for the Recognition of Deformable Objects

Martin Stommel[1] and Klaus-Dieter Kuhnert[2]

[1] TZI Center for Computing and Communication Technologies,
University Bremen, Am Fallturm 1, 28359 Bremen, Germany
mstommel@tzi.de

[2] Institute of Real-Time Learning Systems, University of Siegen
Hoelderlinstrasse 3, 57076 Siegen, Germany
kuhnert@fb12.uni-siegen.de

Abstract. Recognition systems for complex and deformable objects must handle a variety of possible object appearances. In this paper, a compositional approach to this problem is studied which splits the set of possible appearances into easier sub-problems. To this end, a grammar is introduced that represents objects by a hierarchy of increasingly abstract visual alphabets. These alphabets store features, complex patterns and different views of objects. The geometrical constraints are optimised to the respective level of abstraction. The performance of the method is demonstrated on a cartoon data base with high intra-class variance.

1 Introduction

Many recent studies are based on compositional approaches where objects are modeled as parts in comparably loose geometric relationships [1,3,4,12,20]. The idea is to tolerate geometric distortions to a certain degree but to model the characteristic features of an object still correctly. The geometrical constraints are formulated statistically [20,4] or structural [1]. Most research on compositional object models covers the training of single object views (e.g. [4]). While this is appropriate for objects that occur preferentially in a certain distinctive pose, it fails for deformable or moving objects. The trade-off between a bag-of-features model and precise geometric constellations [3,1,4] in the image plane is often discussed. Hierarchical models that cover geometric constellations on different levels of abstraction on the other hand are rare, so possible important dependencies remain widely unidentified. To account for noise or the high intra-class variances of deformable objects, the training methods are usually based on flexible matching procedures. Probabilistic approaches are preferred to exact graph matching procedures [7,5,22].

Although the modeling of the local and global geometry has also been addressed by researchers with a stronger computer graphics background (e.g. [16]), the parts of the compositional models are mostly represented by local descriptors in the shape of feature vectors [10,11,17,6]. Heuristical learning or clustering

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 213–222, 2010.

methods (e.g. [9,8,12,11]) are used to condense the feature set, possibly tuning it to a particular application [21]. The resulting feature sets constitute a certain parallel to the alphabet of moderately complex features found by Tanaka [19] in the inferior temporal (IT) cortex. However, the question how these cells work together to build an internal multi-view object representation, has not been answered yet. Nielsen et al. [15] indicate that monkeys represent rotated objects by different sets of features for different angles of the rotation. Humans however seem to use another mechanism. Miyashita et al. [13,14] report that cells in the IT of a monkey could be trained to sequences of arbitrary patterns and conclude that this mechanism could be used to learn different appearances of a single object.

The work presented here takes these considerations into account. In order to recognise deformable objects in multiple views, we propose a hierarchical model that represents objects by visual alphabets on different levels of abstraction. This allows for a more accurate trade-off between geometry and the part characteristics.

2 Noise-Tolerant Syntactical Model

Inspired by Han and Zhu's [5] recognition procedure, we use an attribute grammar for object modelling. Significant differences consist in the visual primitives, the production rules and the noise handling, though. Let

$$\mathcal{G} = (A_N, A_T, R, S) \tag{1}$$

denote our attribute grammar consisting of a visual alphabet A_N of non-terminal elements, a visual alphabet A_T of terminal elements including the empty word ε, a set of rules R and a root element $S \in A_N$.

The terminal elements are the visual primitives that describe our objects. We use edge points, corners and skeleton points as visual primitives, since there are many suitable feature detectors with known performance. Edge points are further parameterised by their orientation. Skeleton points are parameterised by the orientation of the local skeleton line, the local image intensity and the distance to the nearest edge. The different parameterisations are enumerated and each parameterisation is assigned a single terminal element.

Terminal elements are grouped to complex parts of objects which are represented by the non-terminal elements. Non-terminal elements can in turn be grouped to more complex parts or whole objects.

Terminal and non-terminal elements are attributed by a vector

$$g = (x, y, \theta, \sigma, w) \tag{2}$$

which stores the image position x, y of a recognised visual element as well as trained (attributes θ and σ) and temporary (w) information about the recognition process.

Non-terminal elements are expanded to strings of terminal and non-terminal elements using production rules of the form

$$a_0 \rightarrow a_1 a_2 a_3 \ldots a_n | \epsilon, \quad a_0 \in A_N,\ a_1, a_2, a_3 \ldots a_n \in A_T \cup A_N, \tag{3}$$

defining a context free grammar, basically. Each rule is however associated with a number of constraint equations of the general form

$$f_i(x(a_0)) = g_i(x(a_1), x(a_2), x(a_3), \ldots), \quad i = 1, 2, 3, \ldots \tag{4}$$

where f_i and g_i are projection functions on the attributes of the terminal and non-terminal elements [5]. These functions specify the geometrical relationship between visual elements and guide an acyclic, bottom up recognition procedure. Top down information passing could support rule expansions on lower levels of abstraction by providing information about the surrounding. This is an option for future work, however, and the recognition already works without it. Figure 1 illustrates the usage of the production rules.

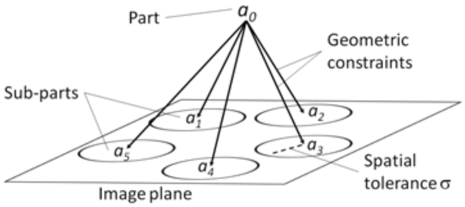


Fig. 1. Illustration of a production rule $a_0 \rightarrow a_1 a_2 a_3 a_4 a_5$. The sub-parts are arranged in certain relative positions, which are modeled by geometric constraints. To account for deformations, small displacements within a tolerance σ are allowed.

The design of the grammar takes into account that due to noise, elements on the right side of a production rule may not be found in an image or that they are spuriously assigned to a wrong element. Noise in the parameterisation of the base features is handled by additional production rules. If an element a_2 is likely to be mistaken for an element a_3, the alternative production rules

$$a_0 \rightarrow a_1 a_2 | a_1 a_3 \tag{5}$$

are created.

If a non-terminal or terminal from the right side of a production rule is not detected in an image, the non-terminal on the left side cannot be expanded. To relax this situation by admitting such missing elements, new rules could be introduced that enumerate all valid subsets of elements on the right side of the original rule. This however would enlarge the rule set unnecessarily.

Instead, special constraint equations are introduced that compensate for missing elements. The aim of these equations is to transfer information about the presence or absence of a certain visual element into the attribute w of a non-terminal element. For the expansion to single terminal elements, this yields the production rules and constraint equations

$$a_0 \rightarrow a_1,\, w(a_0) = 1 \tag{6}$$

$$a_0 \rightarrow\ \varepsilon,\ w(a_0) = 0, \quad a_0 \in A_N,\ a_1 \in A_T. \tag{7}$$

The non-terminal a_0 only expands into a terminal if its weight w is equal to 1. Otherwise it indicates a missing feature.

To handle missing non-terminal elements, the geometry must be taken into account. In our approach, geometric constraints define a relative position d_x, d_y between two elements $a_0, a_i \in A_N$, which could be expressed by a constraint equation like

$$dist(x(a_0), y(a_0), x(a_i) + d_x, y(a_i) + d_y) < \sigma(a_0), \tag{8}$$

where a_0 is on the left side of a production rule, a_i is on the right side, and $dist$ is a suitable distance measure (e.g. the $L_{\max}$-distance). Noise tolerance is introduced by demanding that the number of correctly detected elements on the right side of the production rule exceeds the threshold θ in the attribute vector of the element on the left side of the rule. This results in the production rules and constraint equations

$$a_0 \rightarrow a_1 a_2 a_3 \ldots, \quad a_0, a_1, \dots \in A_N \tag{9}$$

$$\theta(a_0)\ \leq\ \sum_{i\geq 1} \begin{cases} w(a_i) & \text{if eq. 8 holds} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$w(a_0)\ =\ 1 \tag{11}$$

for the correct matching of a part a_0 to a number of sub-parts $a_1, a_2, a_3, \ldots$, and

$$a_0 \rightarrow \epsilon, w(a_0) = 0 \tag{12}$$

for non-terminal a_0 that is not recognised. Equation 10 asserts that the expansion 9 is only valid if a sufficient number of non-terminals on the right side of the production rule is detected. Both the presence of a sub-part (indicated by $w(a_i) = 1$) as well as the geometry are checked. Otherwise, only the expansion into the empty word ϵ is possible. The non-terminal on the left side is then characterised by the attribute $w = 0$ (eq. 12).

Object recognition works by evaluating the constraint equations bottom up from the level of features, over the level of parts, over the level of object poses and appearances, to the level of object classes until the root element S of the grammar is reached. For the root element, a constraint equation demands the attribute value $w(S) = 1$. All visual elements that yield a weight of $w = 1$ are regarded as recognised. Elements with a weight $w = 0$ mark the (near) end of a rule expansion. They occur only if a part of an object can not be matched.

3 Training on Different Levels of Abstraction

The training of the model addresses the visual alphabets A_N and A_T, the noise suppressing parameters σ and θ of the attribute vectors, and the geometrical constraints d_x and d_y of the production rules. The importance of geometrical constraints compared to the feature type is judged differently in the literature. While Weber, Perona et al. [20] promote a highly restrictive constellation model, Crandall et al. [2] and Fergus [4] report a better performance for models with small clique-sizes, i.e. fewer part dependencies. Stommel and Kuhnert [18] find out that the geometrical constraints also depend on the level of abstraction. Therefore we decide to use a differenciated training procedure, that is tuned to the depth of the rule expansion. The training is carried out on the cartoon data base described later. Figure 2 gives an overview.

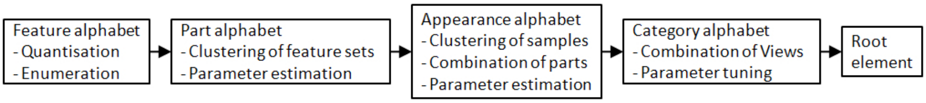


Fig. 2. The Training process comprises the definition of 4 visual alphabets on different levels of abstraction. Taken together they form $A_N \cup A_T$. The production rules assert that non-terminals can only be expanded to less abstract elements.

3.1 Terminal Elements

The deepest level of expansion concerns the terminal elements. They represent edge pieces in different orientations, corners, and center points of homogeneous areas (skeletonisation) with differing intensity, diameter and orientation. The quantisation intervals for the feature parameters are optimised with respect to the information contents measured in a trial classification task. This yields a terminal alphabet of about 1200 elements and the corresponding noise suppressing non-terminals (eq. 6 and 7). The dual integration of edges and the skeletonisation widens the area of application.

3.2 Non-terminals for Object Parts

The non-terminals from the previous step are now grouped to complex object parts. For every part, a non-terminal and the corresponding production rule is created. Since statistical measurements on feature co-occurrences by Stommel and Kuhnert [18] show that spatial proximity is crucial for the frequency of a feature combination in a sample, the creation of more complex visual alphabets is based on the principle of locality.

First, groups of spatially proximate feature vectors are identified in the sample images. To obtain a compact part alphabet with high information capacity, these *candidate patterns* are clustered with regard to their mutual similarity. To this end, every pattern is described by a non-terminal and the corresponding rules that relate it to its terminal features. The relative feature positions are expressed

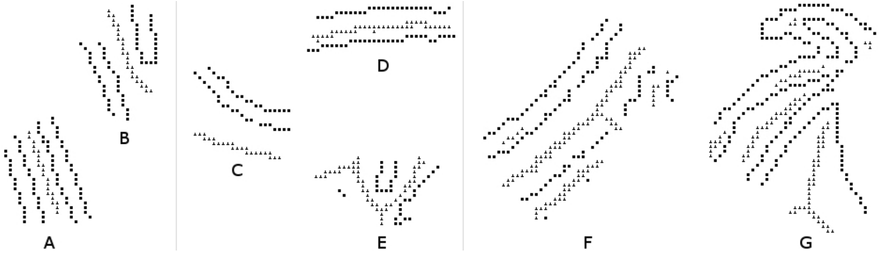


Fig. 3. Patterns from the part alphabet (corresponding to right rule sides): Two patterns from one cluster (patterns A, B). Cluster prototypes of size 20 (C, D, E) and 40 pixels (F, G). Square markings represent edge features, triangles area features.

by constraint equations. Two pattern are regarded similar if the non-terminal of one pattern can be expanded to the terminal elements of the other one. For every resulting cluster, one pattern that is compatible to all other patterns in the cluster is included in the part alphabet. The resulting set consists of about 5000 non-terminals for patterns with a diameter of 10 to 60 pixels.

Initial clusterings showed a high dependency between the threshold θ and the spatial tolerance σ of the attribute vector. Since high thresholds provided the most exact localisations, a fixed threshold $\theta = 90\%$ is chosen for all parts. Further optimisations are done via the remaining free parameter σ, for which an good linear approximation based on the diameter of a pattern is found. Figure 3 shows elements from the resulting part alphabet.

3.3 Non-terminals for Object Appearances

Next, a visual alphabet is trained that abstracts from object pose and appearance. Distinct appearances are found by hierarchically clustering the training images into homogeneous groups. Every group is modeled by a non-terminal together with a production rule that maps it to suitable elements of the part alphabet. The resulting new non-terminals thus constitute a visual alphabet that enumerates the poses and appearances of single objects. The approach is partly neurophysiologically motivated [13,14,15].

Since the right rule sides consist of elements from the part alphabet, the appearance clustering is based on the number of parts that occur jointly in two samples. The occurrence is measured in terms of possible rule expansions.

The result of the clustering is a dendrogram that represents possible groupings of sample images with respect to intersecting part occurrences. Figure 4 illustrates the procedure. For 800 training images and about 5000 patterns in the part alphabet, the dendrogram has a depth of 15.

Next, a set of nodes from the dendrogram must be chosen for modeling and the non-terminals and production rules must be parameterised with respect to their attributes and geometrical constraints.

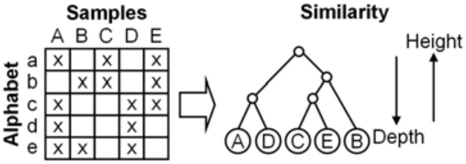
The production rules are constructed by sub-sampling the image positions where parts can be found for all images of a group. For every such position and part, the corresponding non-terminal is added to the right side of the production rule. The geometric constraint is defined accordingly.

The nodes of the dendrogram that are selected for modeling should at least cover all training samples to maximise recall. Overlapping groupings introduce redundancy that increases the robustness to noise at the cost of a bigger model.

Experiments with single appearance models identify the position (depth and height, fig. 4) of a node in the dendrogram as a crucial variable for the parameterisation of the model and the resulting model size (tab. 1): Appearance models for nodes near the root of the dendrogram quickly grow infeasibly large or do not achieve a high precision during classification. It is however possible to restrict the parameter space to a region where a linear model for the noise suppression attribute σ is applicable and the model size stays within treatable limits. The threshold θ is optimised with respect to a minimum number of false matches. Outliers in the histogram of the precision of appearance models are removed, since they increase the false positive rate.

3.4 Non-terminal Elements for Categorisation

A last *category alphabet* is introduced that builds a layer between the root element S and the appearance alphabet. The purpose of this alphabet is to use the redundancy within the appearance alphabet for the optimisation of the recognition system to different configurations in terms of precision and recall. In the



Alphabet	A	B	C	D	E
a	X		X		X
b		X	X		X
c	X			X	X
d	X			X	
e	X	X		X	

Fig. 4. Clustering of object appearances. The table shows which elements of the part alphabet can be expanded to features of a certain sample image by the application of production rules. Samples with similar parts are grouped together. Related groups are represented as nodes in a dendrogram. The height and depth of the nodes are crucial for the parameterisation of the model.

Table 1. Correlation between model parameters and the position of nodes in the appearance dendrogram (fig. 4). The correlation coefficient is normalised to $[-1\ \ 1]$.

1st Variable	2nd Variable	Correlation
Std. deviation size	Size	0.79
Optimal threshold θ	Height	0.73
Optimal threshold θ	Depth	−0.70
Spatial tolerance σ	Height	0.92
Spatial tolerance σ	Depth	−0.69
Number of parts	Spatial tolerance σ	0.81

following, we present two example configurations to achieve either a high suppression of false matches or a high number of detections. Since the appearance models already represent whole objects, the geometrical constraints are set to $d_x = d_y = 0$, i.e. all production rules on this level function as pixelwise bags of features. The elements of the category alphabet represent different classes of objects and therefore the end of the classification process. A sample is assigned the predominant class of all category nodes that have the attribute $w = 1$ after the recognition process.

The first model configuration aims at the suppression of false matches. To this end, for every foreground sample a non-terminal category element is defined that combines all elements of the appearance alphabet that are consistent with this sample. A certain proportion of the resulting category alphabet will also respond to samples from wrong classes. To suppress false matches, the thresholds θ of the category non-terminals are set to an upper bound of the noise level which is estimated practically as the maximum number of non-terminals on the right side of a rule that simultaneously recognise wrong samples.

The second model configuration aims at a higher rate of recognised positive samples. To this end, the threshold θ of every non-terminal from the category alphabet is optimised towards the maximum accuracy measured over all samples. The resulting lower thresholds increase the likelihood of a category element to be expanded to a sample image. The category alphabet is finally pruned until the maximum accuracy is achieved.

4 Data Base and Test Results

The cartoon data base consists of 1600 images showing the head of a cartoon character and 1600 background images of equal size. The background images do not show any cartoon heads. Figure 5 gives an impression of the diversity

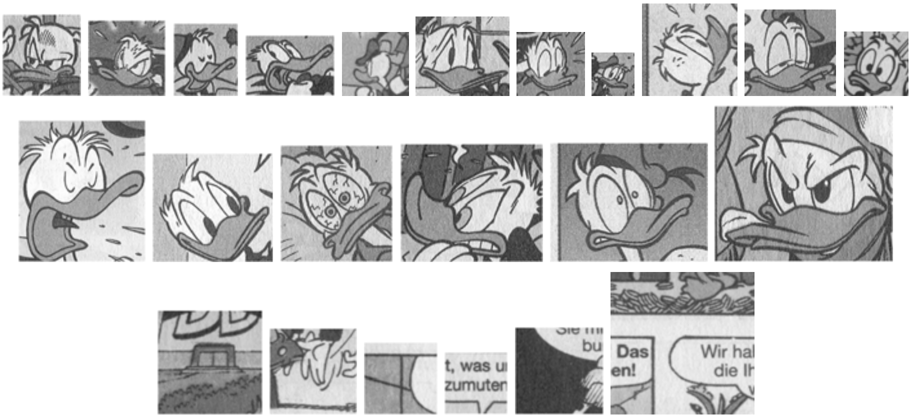


Fig. 5. Samples from the cartoon data base. The upper rows show foreground samples, the bottom row background samples.

of object poses and geometric deformations. The seemingly clear boundaries between adjacent areas are often affected by misalignments between different printing colours. Intensity variations occur across one cartoon page, but also within small areas due to half-tone printing, and between different pages. The data base is thus sufficiently difficult to validate the recognition system. Foreground and background images are randomised and split into equally sized training and test sets. For the first model configuration a precision of 97%, a recall of 57% and an accuracy of 78% is achieved. The recognition is thus very reliable, but also rather selective. The second model configuration yields a recall of 89%, a precision of 72% and an overall accuracy of 77%. The general performance of the model is thus equally high but the recall is significantly higher and the classifier works less conservative.

5 Conclusion

An object recognition system is presented that combines a syntactical model and statistical training method. The structure of the underlying grammar is noise resistent and based on both the geometric properties of feature co-occurrence statistics, as well as on neurophysiological findings about pose representation. Since the training method creates visual alphabets on different levels of abstraction, crucial dependencies on the depth of the rule expansion are taken into account. A data base with images of a strongly deformable cartoon character is used to test the method. Two exemplary model configurations are presented which tune the model either towards a high precision of 97% or a high recall of 89%. A rate of correct classifications of 77%–78% for both configurations demonstrates the performance of the method.

References

1. Behmo, R., Paragios, N., Prinet, V.: Graph Commute Times for Image Representation. In: IEEE Conference in Computer Vision and Pattern Recognition, CVPR 2008 (2008)
2. Crandall, D.J., Felzenszwalb, P.F., Huttenlocher, D.P.: Spatial Priors for Part-Based Recognition Using Statistical Models. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10–17 (2005)
3. Crandall, D.J., Huttenlocher, D.P.: Composite Models of Objects and Scenes for Category Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2007)
4. Fergus, R., Perona, P., Zisserman, A.: A Sparse Object Category Model for Efficient Learning and Complete Recognition. In: Ponce, J., Hebert, M., Schmid, C., Zisserman, A. (eds.) Toward Category-Level Object Recognition. LNCS, vol. 4170, pp. 443–461. Springer, Heidelberg (2006)
5. Han, F., Zhu, S.C.: Bottom-up/Top-Down Image Parsing by Attribute Graph Grammar. In: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV), vol. 2, pp. 1778–1785 (2005)

6. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. Technical Report IRP-TR-03-15, School of Computer Science, Carnegie Mellon University (2004)
7. Lee, W.-J., Duin, R.: An Inexact Graph Comparison Approach in Joint Eigenspace. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 35–44. Springer, Heidelberg (2008)
8. Lin, Z., Hua, G., Davis, L.: Multiple Instance Feature for Robust Part-based Object Detection. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2009)
9. Liu, J., Yang, Y., Shah, M.: Learning Semantic Visual Vocabularies Using Diffusion Distance. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2009)
10. Lowe, D.G.: Object Recognition from Local Scale-Invariant Features. In: Proc. of the International Conference on Computer Vision (ICCV), Kerkyra, Greece, September 1999, vol. 2, pp. 1150–1157 (1999)
11. Mikolajczyk, K., Leibe, B., Schiele, B.: Local Features for Object Class Recognition. In: International Conference on Computer Vision ICCV 2005 (October 2005)
12. Mikolajczyk, K., Leibe, B., Schiele, B.: Multiple Object Class Detection with a Generative Model. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2006, vol. 1, pp. 26–36 (2006)
13. Miyashita, Y., Sakai, K., Higuchi, S.-I., Masui, N.: Localization of Primal Long-Term Memory in the Primate Temporal Cortex. In: Squire, L.R., Weinberger, N.M., Lynch, G., McGaugh, J.L. (eds.) Memory: Organization And Locus of Change (1991)
14. Miyashita, Y.: Inferior Temporal Cortex: Where Visual Perception Meets Memory. Annual Reviews of Neuroscience 16, 245–265 (1993)
15. Nielsen, K.J., Logothetis, N.K., Rainer, G.: Object features used by humans and monkeys to identify rotated shapes. Journal of Vision 8(2), 1–15 (2008)
16. Salzmann, M., Urtasun, R., Fua, P.: Local Deformation Models for Monocular 3D Shape Recovery. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2008)
17. Stark, M., Schiele, B.: How good are local features for classes of geometric objects. In: IEEE 11th International Conference on Computer Vision ICCV, pp. 1–8 (2007)
18. Stommel, M., Kuhnert, K.-D.: A Hierarchical Model for the Recognition of Deformable Objects. In: Int'l Conf. on Computer Vision and Graphics 2008 (ICCVG 2008), Warsaw, Poland, November 10–12 (2008)
19. Tanaka, K.: Inferotemporal cortex and object vision. Annual Reviews of Neuroscience 19, 109–139 (1996)
20. Weber, M., Welling, M., Perona, P.: Unsupervised Learning of Models for Recognition. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 628–641. Springer, Heidelberg (2000)
21. Yang, L., Jin, R., Sukthankar, R., Jurie, F.: Unifying Discriminative Visual Codebook Generation with Classifier Training for Object Category Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2008)
22. Zass, R., Shashua, A.: Probabilistic Graph and Hypergraph Matching. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2008)

Graph Embedding Based on Nodes Attributes Representatives and a Graph of Words Representation

Jaume Gibert and Ernest Valveny

Centre de Visió per Computador, Universitat Autònoma de Barcelona
Edifici O Campus UAB, 08193 Bellaterra, Spain
{jgibert,ernest}@cvc.uab.es

Abstract. Although graph embedding has recently been used to extend statistical pattern recognition techniques to the graph domain, some existing embeddings are usually computationally expensive as they rely on classical graph-based operations. In this paper we present a new way to embed graphs into vector spaces by first encapsulating the information stored in the original graph under another graph representation by clustering the attributes of the graphs to be processed. This new representation makes the association of graphs to vectors an easy step by just arranging both node attributes and the adjacency matrix in the form of vectors. To test our method, we use two different databases of graphs whose nodes attributes are of different nature. A comparison with a reference method permits to show that this new embedding is better in terms of classification rates, while being much more faster.

1 Introduction

Most real-world problems do not fit under the usual data representation by means of feature vectors. Instead, structural representations are more suitable. Graph-based representations offer interesting properties in terms of binary relations between features allowing to adapt the representation to the complexity of data while vectors are constrained to the use of a predefined number of features.

However, while structured representations provide us with a complex and powerful description of the patterns under study, their own complexity makes the processing and analysis of graphs a really hard problem. Graph matching is the process that tries to discover the structural similarity of two graphs. To know more about graph matching we refer the reader to [1], a detailed survey that organizes the whole map of techniques for solving this problem.

On the other hand, many pattern recognition techniques have been developed for data represented in the form of feature vectors. The fact that vector spaces have strong and straightforward mathematical properties, both theoretical and practical, has contributed to the expansion of pattern analysis techniques for the case when patterns are represented by elements in a feature space. In order to make all these techniques available for the case of data structurally represented,

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 223–232, 2010.

for instance, by using graphs, the scientific community has devoted several efforts to find out new ways of adapting them to structured data. Graph embedding is one of them. By graph embedding we understand a function that given a set of graphs G, it maps each graph in the set to an n-dimensional vector,

$$\begin{aligned}\phi : G &\rightarrow \mathbb{R}^n \\ g &\mapsto \phi(g) = (x_1, x_2, \ldots, x_n).\end{aligned} \tag{1}$$

Several graph embeddings have been proposed in the literature so far. Some of them are based on a spectral study of the adjacency matrix or on the Laplacian matrix of the graphs [2]. Other approaches are based on random walks, and particularly on quantum walks, in order to embed nodes into a vector space [3]. Finally, let us consider the embedding proposed in [4] which is based on similarity measures between graphs and a set of prototypes. The measure considered by the authors is the Graph Edit Distance. This embedding will be later explained more in detail as it will be used as a reference system in order to evaluate our approach.

However, some of these embeddings are constrained to specific classes of graphs or still rely on graph matching. The computational complexity of graph matching lies on what is known as the assignment problem. Nodes of one graph have to be identified with nodes of the other one, and such procedure has an exponential computational cost in the number of nodes of the involved graphs. This problem has been addressed, for instance, by means of best-first search techniques like the A^* algorithm or by bipartite graph matching procedures and the Hungarian method. Ideally, however, if nodes of one graph could directly be identified with nodes of the other, we would not have to face this problem and graph matching would be a problem with a straightforward solution.

In this paper we aim to propose a graph embedding avoiding the computational cost of graph matching through an intermediate meta-representation of the graphs in which nodes of a family of graphs become identifiable. We will call this meta-representation graph of words as the underlying idea is based on the well-known *bag of words* technique for document and image classification [5]. Thus, we will cluster node attributes to obtain representatives (words) of the node attributes of a set of graphs. This will lead to represent the whole set by graphs that have exactly the same number of nodes and share the same node

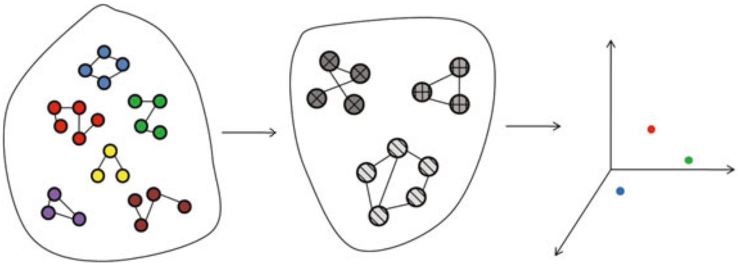


Fig. 1. General scheme of the proposed approach. First step: graph generalization. Second step: graph embedding.

labels. Then, this graph representation can be easily converted into a vector by just taking node attributes and serializing the adjacency matrix. In Fig. 1 the whole procedure is depicted.

In the remainder of this paper we describe, first in Section 2, how the generalization of a set of graphs into a graph of words can be done, and second, in Section 3, how these generalized graphs are converted into vectors. Section 4 describes the databases we have used for the experimentation part and the results and discussions are presented in Section 5. Finally, Section 6 concludes and present the future work to be done.

2 Graph of Words Representation

An attributed graph g, or just a *graph*, is a 4-tuple $g = (V, E, \mu, \nu)$, where V is a finite set of elements, called nodes, $E \subseteq V \times V$ is the set of edges, and $\mu : V \rightarrow L$ and $\nu : E \rightarrow L'$ are the corresponding labelling functions for which each node and edge is correspondingly attributed with a specific label. As there is no restriction on the set of labels for both nodes and edges, this definition allows to describe a large family of graphs. For instance, the set of nodes labels L could be represented either by the vector space $\mathbb{R}^n$ or even by a set of non-numerical attributes $L = \{l_1, l_2, l_3, \dots\}$ with a certain specific semantics. The same happens for the case of edges attributes L'. Edges of the graphs are described by pairs of nodes (u, v), where $u \in E$ is the source node and $v \in E$ is target one. Undirected graphs are all those graphs which contain both $(u, v) \in E$ and $(v, u) \in E$ satisfying $\nu(u, v) = \nu(v, u)$. Unlabelled graphs are graphs where both nodes and edges have the same label, usually called the null label ϵ. In this work we will just consider undirected graphs with unlabelled edges and with all the node labels being of the same nature.

In this section we will introduce a way to generalize almost any set of graphs described as before into a new set of graphs with the same number of nodes and sharing the same node labels. The only constraint about the input graphs is that all node attributes must be of the same type. In addition, if edges of the original graphs are labelled, these labels are ignored in the conversion procedure. The generalization is done in two steps: first, getting the set of nodes of the new graph by clustering and selecting representatives among the original node attributes and second, getting the edges that link the new set of nodes. These two steps are further described in the next subsections.

2.1 Node Representatives

As we already pointed out in the introduction, one of the main problems in graph matching is node assignment. To avoid such problem, we would appreciate an ideal situation where nodes in a family of graphs were directly in correspondence. Not by just *a priori* knowledge about the information stored in the nodes, but by the fact that those nodes were exactly representing the same attributes, in other words, that two graphs in a family of graphs share exactly the same nodes. A possible way to reach this situation is the one described as follows.

Given a set of graphs $G = \{g_1, g_2, \ldots, g_n\}$, each one with its corresponding nodes, edges and labelling functions, $g_i = (V_i, E_i, \mu_i, \nu_i)$, and the corresponding labels sets (L_i, L'_i), we want to represent the whole family of graphs using generalized representations sharing the same nodes. To do so, we first consider all node attributes $\mathcal{L} = \{L_i \,|\, i = 1, \ldots, n\}$. We assume all these sets of labels are of the same nature, this is, we do not consider -if there is such- the case in where for instance L_j would be representing numerical attributes and L_k semantic attributes.

Second, from this set of labels we select a finite number of representatives. Such representatives do not need to be elements from $\mathcal{L}$. We will hereby adopt a quite known notation extracted from the *bag of words* technique for document -originally- and image classification [5]. This technique represents an image by first extracting visual features from interest points and selecting representatives, usually called *visual words*, from the whole set of features of the images under study. The set of representatives is usually built by clustering and is called the visual vocabulary. Every interest point is assigned to its closest word in the vocabulary and finally the image is represented as a histogram of appearing words in the image. We perform the selection of node attribute representatives in a similar way. For this reason we call this new representation the *graph of words*. From the whole set of attributes $\mathcal{L}$, we select representatives, *words*, by means of any clustering technique. How the selection of representatives has to be done for a given family of graphs will depend on the nature of the nodes attributes, and it will be a key issue in the generalization graph. For instance, a k-Means algorithm could be applied if the attributes of the set of graphs belong to $\mathbb{R}^n$, but in other cases, the clustering should be done in a semantic way.

Assume we have already selected a vocabulary $\mathcal{V} = \{w_i \,|\, i = 1, \ldots, N\}$ from the set of attributes $\mathcal{L}$ of the graph family G. We call every w_i a word. The generalization of a graph $g = (V, E, \mu, \nu)$ is done as follows: every node $u \in V$ of the graph is assigned to its closest or most similar word in $\mathcal{V}$. We denote this assignment by

$$\begin{aligned} \lambda : V &\rightarrow \mathcal{V} \\ u &\mapsto \lambda(u) = w. \end{aligned} \tag{2}$$

The concept of similarity here will depend on the nature of the attributes and selected words. The set of words in the vocabulary that has been assigned to at least one point of the original graph will now constitute the set of nodes of the generalized graph. We label each node of the new graph, each word, with the frequency of nodes assigned to it. Fig. 2 shows an example of the procedure. The graph has six nodes. Three of them have been assigned to the red word, two to the green one and one to the blue word. In the generalized graph, these three words will appear as nodes, with their corresponding frequencies. In this example, we assume a vocabulary which also contains a yellow word but no node of the original graph has been assigned to it.

By this procedure we have got a representation of graphs where the set nodes of all the graphs in a given family is the same, this is, the set of representatives or vocabulary. This will allow to treat the assignment problem in a very specific

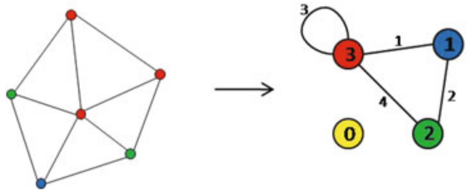


Fig. 2. A graph (left) and its generalization graph (right) or graph-of-words

way so that no computational issues are involved. However, before describing the way to deal with it, we still need to declare which is the set of edges in the graph of words representation. Next section is devoted to it.

2.2 Structural Relations of Representatives

The way of defining edges between words in the generalized graph is the most intuitive one. Given a graph g, let us assume there exist an edge $e = (u, v) \in E$. Each node is assigned to a word in the vocabulary: there are $w, w' \in \mathcal{V}$ such that $\lambda(u) = w$ and $\lambda(v) = w'$. Then we just add the edge (w, w') in the set of edges of the generalized graph. Formally,

$$(w, w') \in E' \iff (u, v) \in E \text{ such that } \lambda(u) = w \text{ and } \lambda(v) = w' \qquad (3)$$

where $E' \subseteq \mathcal{V} \times \mathcal{V}$ is denoting the set of edges of the graph of words. We will label the edge $(w, w') \in E'$ with the frequency this fact occurs, this is, how many times two nodes of the original graph that have been assigned to the words w and w' are linked with an edge of the original set of edges E. Again, Fig. 2 depicts an example of the procedure. For example, a node that has been assigned to the green word is connected to a node that has been assigned to the red word. Thus, the pair $(green, red)$ will constitute an edge of the graph of words. Actually this situation occurs four times, so we label the edge with the number 4.

3 Graph of Words Embedding

Our generalized graphs, the graphs of words, are of a very particular nature as all of them share the same nodes. This property makes the transition between the domain of the graphs and a vector space a really easy problem. Next section describes formally how this can be done.

3.1 Definition

Let $g = (V, E, \mu, \nu)$ be a graph of words with respect to the vocabulary $\mathcal{V} = \{w_i \,|\, i = 1, \ldots, N\}$ of size N. Remember μ was describing the frequency of a word, and ν the frequency of a relation between two words. We can always describe the set of nodes of g as $V \equiv \mathcal{V}$. If a specific word w does not appear as a node of the graph we can consider it as a node with $\mu(w) = 0$. Exactly as the yellow word in the example shown in Fig. 2.

The fact that we control which words are there in the vocabulary, makes the nodes of g identifiable, and therefore, sortable. We can arrange, for all graphs in a set G of graphs of words, the node labels as:

$$\begin{aligned} \phi_w^{\mathcal{V}} &: G \rightarrow \mathbb{R}^N \\ g &\mapsto \phi_w^{\mathcal{V}}(g) = (\mu(w_1), \ldots, \mu(w_N)). \end{aligned} \tag{4}$$

Let us now consider the adjacency matrix A of the graph g. Matrix A is an $N \times N$ matrix and each entry of it a_{ij} is giving the frequency of the relation between the words w_i and w_j. Here, as for the case of the nodes, we can represent a non-existing edge (w_i, w_j), by labelling it with zero value, $\nu(w_i, w_j) = 0$. This matrix is obviously symmetric since the graph-of-words is an undirected graph.

In this case, we can just arrange the adjacency matrix in the form of a vector. Not all entries of the matrix are needed since it is symmetric and that would be redundant. We can just consider, for instance, the upper part of the diagonal and the diagonal.

$$\begin{aligned} \phi_r^{\mathcal{V}} &: G \rightarrow \mathbb{R}^p \\ g &\mapsto \phi_r^{\mathcal{V}}(g) = (\nu(a_{11}), \nu(a_{12}), \ldots, \nu(a_{ij}), \ldots, \nu(a_{NN})), \ \forall i \leq j. \end{aligned} \tag{5}$$

Finally, we define the embedding of a set of graph of words G with respect to the vocabulary $\mathcal{V}$ as the simple concatenation of both the nodes and the edges attributes, this is, of both the words frequencies and the frequencies of the relations between words. Formally,

$$\begin{aligned} \phi_A^{\mathcal{V}} &: G \rightarrow \mathbb{R}^n \\ g &\mapsto \phi_A^{\mathcal{V}}(g) = (\phi_w^{\mathcal{V}}(g), \phi_r^{\mathcal{V}}(g)). \end{aligned} \tag{6}$$

3.2 Computational Issues and Potential Solutions

The main problem we might face after embedding the graphs using $\phi_A^{\mathcal{V}}$ is the dimension of the vectors. Indeed, such dimension increases quadratically with respect to the vocabulary size N. The first part of the vector, the words part $\phi_w^{\mathcal{V}}(g)$ has size N and the second one, $\phi_r^{\mathcal{V}}(g)$, has size $p = (N^2 + N)/2$ resulting in a vector of dimension $n = (N^2 + 3N)/2$. Processing this vector could not be treatable for a large enough vocabulary size.

Dimensionality reduction techniques could be applied in these situations. Semantically, such reduction would discover which are the specific relations of words that are really important in the graph of words representations. Also, in order to avoid such large dimensions, the whole construction of the generalized representation could be performed in a supervised manner. This is, selecting representatives class-wise and executing the embedding part of the adjacency matrix $\phi_r^{\mathcal{V}}$ by just considering relations of words belonging to the same class. We had not faced this problem as the vocabularies we have selected for our databases of graphs were not that large. The construction of the generalized graphs for two different databases is explained in the next section.

4 Experimental Setup

The whole set of experiments is carried out using two databases inspired in the IAM Graph Database Repository [6]. Even though we do not consider semantic attributes of nodes, the two situations under study refer to two different kind of attributes. In the first case, nodes are labelled with (x, y) positions and in the second one with visual descriptors.

In the way the graph of words representation has been defined, the embedded graphs clearly depend on the chosen vocabulary. For instance, the use of a small vocabulary makes easy the introduction of errors in the assignment of points to words and every word would represent too many and possibly wrong points. On the other hand, a large vocabulary would create too much sparsity on the resulting vectors making difficult further analysis of data. Due to these situations, in order to check the sensibility of the representation to the selected vocabulary, different sizes of the nodes representatives sets have been tried for each dataset.

4.1 IAM Letter Database Generalization

The graphs in this database are representing distorted letter drawings. Only the 15 capital letters of the Roman alphabet that consist of straight lines are considered: *A*, *E*, *F*, *H*, *I*, *K*, *L*, *M*, *N*, *T*, *V*, *W*, *X*, *Y*, and *Z*. For details on the construction of the graphs we refer the reader to the database reference. We use exactly the same sets as the authors in the reference: uniformly distributed graphs all over the 15 classes and a training, validation and test sets of 750 graph each. We only work on the *low* level of distortion.

By representing distorted letters using the graph of words, we aim to undo the distortion each letter has suffered. In order to build the generalized graphs for the letters, let us notice that nodes of the graphs are attributed with the (x, y) coordinates with respect to a reference coordinate system. We do not cluster all these coordinates, but instead we just pick a regular grid of $n \times n$ points in the range of the nodes attributes ($n = 3, 7, 11$). This decision was made after viewing how all nodes of all letter graphs were distributed over the plane.

4.2 COIL-100 Database Generalization

The second graph dataset we use in our experiments is representing the COIL-100 object database [7]. It consists of 100 different object, and each image is taken every 5 degrees of rotation. We represent these objects modifying the COIL-DEL representation from the IAM Graph Dataset repository. This representation takes interest points by using the Harris corner detection algorithm, labels these points with their corresponding pixel coordinates and applies a Delaunay triangulation to the nodes for the set of edges. Our modification consists of labelling such points with a visual descriptor that could provide texture information of the objects. In particular we label each node with a SIFT descriptor [8]. Again, the same sets for training, validating and testing the systems are used.

By building the generalized graphs for these objects we intent to describe the structural relations of textural patterns, this is, how visual descriptors are structurally related. In this case, we cluster the set of SIFT descriptors using the k-Means algorithm and finding $k = 50, 100, 150$ centroids as representatives. Thus, the dimensionality of the vector space where graphs are embedded is much larger than in the letter dataset, yet the vectors are still manageable.

5 Results

We want to classify both databases by using the graph of words embedding. Such classification is done by using a *Support Vector Machine* [9,10] on the embedded graphs. Kernels and parameters are properly tuned using the validation sets. We need a reference system to which we can compare how the proposed embedding works, in other words, how good is the idea of embedding the graphs by just arranging their nodes and edges attributes in a row. To do so, we select the proposed embedding in [4], by which each graph is assigned to a vector constructed out of Graph Edit Distances from itself to a set of prototypes. This embedding has been applied to the intermediate graph of words representation. Next section recalls the formal definition of the embedding.

5.1 A Reference System

Let G be a set of graphs and $\mathcal{P} = \{p_1, p_2, \ldots, p_n\} \subseteq G$ a set of prototypes. The embedding is defined as the function $\varphi_n^{\mathcal{P}} : G \to \mathbb{R}^n$ such that

$$\varphi_n^{\mathcal{P}}(g) = (d(g, p_1), d(g, p_2), \ldots, d(g, p_n)) \tag{7}$$

where $d(g, p_i)$ is the Graph Edit Distance between the graph g and the prototype p_i.

As we are using the Graph Edit Distance between instances of graph of words, we need to specify which is the cost function we have used during the matching process. Briefly, the node and edge deletion and insertion costs are the labels (frequencies) of the inserted or deleted nodes and edges. For the substitution of nodes and edges we pick the difference of labels if the substitution is of the same words or edges between the same words, and infinity otherwise.

5.2 Classification Rates

Results for both databases, the Letters and the COIL-100 objects, using the proposed embedding and the reference one are shown in Tab. 1.

We see how the proposed embedding $\phi_A^{\mathcal{V}}$ works better than the reference one no matter whether the dimension of the vectors is larger or smaller than for the graph edit distance embedding. For instance, in the letter database the embedded graphs with respect to the 3×3 grid of points live in a 54-dimensional space when using $\phi_A^{\mathcal{V}}$, and in a 750-dimensional one when the case is $\varphi_n^{\mathcal{P}}$. On the other database, when using a vocabulary size of 150 words and the proposed

Table 1. Classification rates for both databases using the proposed embedding and the reference one. Vocabularies for the Letter Database are $n \times n$ regular grids; k-Means algorithm is used for the COIL Database case. For the case of $\varphi_n^{\mathcal{P}}$ the whole set of training elements is used as the prototypes set.

Database	Vocabulary	Embedding approach	
		Proposed $\phi_A^{\mathcal{V}}$	GED-based $\varphi_n^{\mathcal{P}}$
Letter	3×3	**97.33**%	96.4%
	7×7	**99.2**%	96.13%
	11×11	**96.93**%	94.4%
COIL	50	**82.8**%	80.6%
	100	**88**%	85.6%
	150	**89.4**%	88.9%

embedding, the COIL objects are represented by really high dimensional vectors ($\simeq$ 12000 dimensions) and not that large but still (2400 dimensions) in the case of the graph edit distance based embedding.

Of course, related to what it has been already said in the introduction, the computational cost of the processing and comparison of the graphs is actually null when we arrange them in the form of vectors using $\phi_A^{\mathcal{V}}$. The other situation is not desirable since graph edit distance takes a lot of time. For example, the embedding of COIL objects using $\varphi_n^{\mathcal{P}}$ took a couple of days in a regular personal computer.

6 Conclusions and Future Work

In this work, we have proposed a new way of solving the graph classification problem by means of embedding graphs into vectorial spaces. The embedding procedure consists mainly in two steps: a generalization of the graphs using what we call the graph of words representation by assigning nodes to attribute representatives and then, arranging this new representation in the form of vectors. The experiments have shown the power of the described approach in terms of classification rates and time consuming with respect to a well-known and established embedding of graphs.

The generalization step, the so-called graph of words representation, is dependent on the choice of a set of representatives of the nodes attributes. Different selections of both vocabulary types and sizes will lead to different classifiers, not necessarily one better than others. Small vocabularies will gather more points together into every word, describing global structural relations between nodes of the original graphs, while larger vocabularies will distribute the points into more specific words retaining local information of the graphs under study. Such a situation could be exploited by ensembles of classifiers that would increase the overall accuracy of the individual classifiers.

The whole procedure explained does not take into account edge attributes of the graphs. Several graph databases do have edge attributes in their graphs because such attributes may describe important information about the relation between nodes of the graphs. It is quite clear that we should adapt our graph embedding to those cases where edge attributes are present. One possible way to proceed in this situation would be to also cluster the edge attributes and add a new semantic feature to the relation between words. Then, we should probably face the problem of dimensionality reduction.

This paper is mainly about a really preliminary work in the sense that a lot of experimentation is still needed. We focused on evaluating only the second step in the whole procedure: the representation of the graph of words using vectors. A comparison between the whole proposed embedding, which needs a transition to the generalized graph, and embeddings of the original graphs with other techniques has to be done and constitutes the current work of the authors.

References

1. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. Int. Journal of Pattern Recognition and Artificial Intelligence 18(3), 265–298 (2004)
2. Luo, B., Wilson, R.C., Hancock, E.R.: Spectral embedding of graphs. Pattern Recognition 36(10), 2213–2230 (2003)
3. Emms, D., Wilson, R.C., Hancock, E.R.: Graph Embedding Using Quantum Commute Times. Graph-based Representations in Pattern Recognition, 371–382 (2007)
4. Bunke, H., Riesen, K.: Recent Developments in Graph Classification and Clustering using Graph Embedding Kernels. Pattern Recognition in Information Systems, 3–13 (2008)
5. Dance, C., Willamowski, J., Fan, L., Bray, C., Csurka, G.: Visual categorization with bags of keypoints. In: ECCV International Workshop on Statistical Learning in Computer Vision, pp. 1–22 (2004)
6. Riesen, K., Bunke, H.: IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 287–297. Springer, Heidelberg (2008)
7. Nene, S., Nayar, S., Murase, H.: Columbia Object Image Library: COIL-100. Technical report, Dept. of Computer Science, Columbia University, New York (1996)
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
9. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2001)
10. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001) Software available at, http://www.csie.ntu.edu.tw/~cjlin/libsvm

Extracting Plane Graphs from Images*

Émilie Samuel[1], Colin de la Higuera[2], and Jean-Christophe Janodet[1]

[1] Université de Lyon, F-42023, Saint-Étienne, France
CNRS, UMR5516, Laboratoire Hubert Curien, 42023, Saint-Étienne, France
Université de Saint-Étienne, Jean Monnet, F-42023, Saint-Étienne, France
{emilie.samuel,janodet}@univ-st-etienne.fr
[2] CNRS, UMR6241, LINA, 44322, Nantes, France
Université de Nantes, F-44322, Nantes, France
cdlh@univ-nantes.fr

Abstract. In order to use structural techniques from graph-based pattern recognition, a first necessary step consists in extracting a graph in an automatic way from an image. We propose to extract *plane graphs*, because of algorithmic properties these graphs have for isomorphism related problems. We also consider the problem of extracting semantically well-founded graphs as a compression issue: we get simple graphs from which can be rebuilt images similar to the initial image. The technique we introduce consists in segmenting the original image, extracting interest pixels on the segmented image, then converting these pixels into pointels, which in turn can be related by region-based triangulation. We show the feasibility and interest of this approach in a series of experiments.

Keywords: Plane graphs, images, interest pointels, segmentation, Delaunay triangulation.

1 Introduction

Representing images with graphs is an approach followed in pattern recognition in which it is hoped to benefit from the structural properties of the graphs, but also to be able to use the robustness associated with them [2,10]. A number of ideas have been analysed in order to use these graphs once built, involving edit distances [17] or graph matching [14]. Moreover, databases of synthetically generated graphs [19] or graphs related to pattern recognition benchmarks [16] have been created.

Many different models of graphs extracted from images have already been proposed. A first alternative consists in segmenting the image, in representing each region by a vertex and putting an edge between two vertices whenever the regions have a common border. Further arguments can be added to indicate the size of the region, the length of the border, and so forth. The obtained graphs

* This work was partially supported by an ANR grant BLANC 07-1-184534 (project Sattic), and the IST Programme of the European Community, under the PASCAL 2 Network of Excellence, IST–2006-216886.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 233–243, 2010.

are called Region Adjacency Graphs [18] and have been improved, yielding dual graphs [11], ordered graphs [9], combinatorial maps [12]. A second option consists in extracting interest points (they will now be the vertices) and in relating them according to some neighbourhood relationship (the Delaunay triangulation may be a way to do this). Again, numerical attributes can be added to the representation, either attached to the vertices or to the edges.

Yet neither of the two aforementioned techniques is satisfying: typical drawbacks in the first case are that the structure of the graph actually has little to do with the topological nature of the image (i.e. the spatial disposition of depicted objects, their borders and connexity), in particular when the segmentation is rough. In the second case, not only do many of the so-called interest points not seem to be that interesting (requiring to extract many more points than necessary in order to be sure to have all the important ones), but also, since these points are actually pixels (with non null thickness), the construction is hindered by discrete geometry issues (e.g. what is the frontier between two regions?).

What should be the features of a good graph extracted from an image? On one hand, it should be robust: if the image is slightly distorted, the graph should not be deeply modified. It should also have good algorithmic properties: typically the central isomorphism problem, the related problems of sub-graph isomorphism, maximum common subgraph and graph edit distance should be solvable with polynomial algorithms. Furthermore what we may call the semantic properties of the image should be present in the graph: this might mean that the graph, when drawn, actually is a symbolic representation of this image; for instance, the boundaries between the regions could coincide with the edges of the graph. Finally we would like to be able to rebuild an image from the graph and for this new image to be a good compression of the initial image. In other terms, the *loss* due to the extraction process should be minimal.

In this paper we explore the possibility of combining the advantages of both the techniques presented above: on the one hand, segmentation introduces robustness; on the other hand, extraction of interesting pixels is a good idea and much less pixels are going to be needed from a segmented image. The method we suggest is thus as follows: (1) segment the image into regions, (2) extract s interest pixels, (3) add s' intersection pixels (s' being generally less than s), (4) compute the corresponding interest *pointels*, (5) relate them according to the boundaries of the regions and (6) triangulate the pointels of each region.

2 Definitions

A (digital) *image* I is an array of $n*m$ *pixels*. Each pixel p has a colour described by a 3 dimensional vector $c_I(p)$ in standard RBG colour system; that latter hypothesis corresponds to our experimental settings but any other colour system could be used. The coordinates of the pixels consist in pairs (i, j); their range goes from $(0, 0)$ to $(n-1, m-1)$. Each pixel is delimited from each of his four neighbours by a *linel*, and the four corners of the pixels are called *pointels*. Such pointels and linels are mathematical points and line respectively, with null thickness. This enables us to associate the coordinates of the pointels to those of the pixels; their range goes from $(0, 0)$ to (n, m).

We now introduce the main criterion used in this paper to measure the loss between an original image I_1 and an image I_2 obtained after a series of transformations. We assume that both I_1 and I_2 have the same size $(n * m)$. The *loss* is the L_1 distance between I_1 and I_2 divided by the number $n * m$ of pixels. Formally, we first define the distance d_p between any pixel p_1 in image I_1 and pixel p_2 in image I_2 as the normalised distance between corresponding colour vectors. That is, if $c_{I_1}(p_1) = (r_1, g_1, b_1)$ and $c_{I_2}(p_2) = (r_2, g_2, b_2)$, then $d_p(p_1, p_2) = (|r_2 - r_1| + |g_2 - g_1| + |b_2 - b_1|)/(3 * 255)$. And we state:

$$\text{loss}(I_1, I_2) = \frac{\sum_{i=0}^{i=n-1} \sum_{j=0}^{j=m-1} d_p\big(p(I_1, i, j), p(I_2, i, j)\big)}{n * m}, \tag{1}$$

where $p(I_\ell, i, j)$ denotes pixel with coordinates (i, j) in image I_ℓ.

We now introduce *plane graphs.* We recall that a graph $G = (V, E)$ is *planar* if it can be embedded in the plane, in such a way that no two edges cross. Note that several non homeomorphic embeddings may exist for a given planar graph (i.e. several incomparable drawings of the same graph) [3]. Moreover, it is always possible to move the vertices so that the edges are drawn with straight-line segments [6]. Thus by a *plane graph*, we mean a planar graph for which the embedding is fixed and such that (1) every embedded edge is a straight-line and (2) no two embedded edges intersect, except at their endpoints.

Plane graphs are interesting to represent images for at least two reasons. On one hand, they can be nicely represented as combinatorial maps which also provide us with an elegant data structure [20]. On the other hand, isomorphism and subisomorphism problems were recently studied for combinatorial maps [5], with the following interesting consequence:

Theorem 1. *Let $G = (V, E)$ and $G = (V', E')$ be two plane graphs. Deciding whether G and G' are isomorphic is solvable in time $\mathcal{O}(|V| \times |V'|)$. Moreover, if G is face-connected, deciding if G is a pattern of G' is solvable in time $\mathcal{O}(|V| \times |V'|)$.*

A pattern of a graph can be seen as a subgraph obtained by erasing faces and then edges (called *bridges*) separating 2 erased faces and finally singular vertices (called *hinges*) separating 2 non-contiguous erased faces [4]. Other positive algorithmic results for special classes of graphs are described in [9].

In this paper, we develop techniques to extract a plane graph G from an image. In order to rebuild an image, we also need to have (1) a vector P mapping each vertex with the coordinates of corresponding pointel and (2) a vector C mapping each face of G with the colour in which it should be drawn[1]. Although G, P and C are known, a variety of images can still be drawn. Indeed, embedded edges will cross many pixels and divide them into several parts, so which colour should these pixels be given? We choose the following rule: for each split pixel, we consider the colours of its four corners (pointels) (inherited from the faces in which they

[1] Faces of plane graphs can be denoted and saved using directed edges, and a convention stating that such or such face is on the right of such or such directed edge.

stand - pointels that lie between regions are not taken into account), compute the average colour and assign it to the pixel.

We denote by $\mathrm{Im}(G, C, P)$ the resulting image and can now measure the loss between $\mathrm{Im}(G, C, P)$ and original image I using Eq.(1). Obviously, the loss between an image obtained through some compression or representation by graphs could also be looked into *w.r.t.* the gain in terms of space. We claim that whereas the space needed to encode an image is $n \cdot m \cdot K$, where K is a constant corresponding to the size needed to encode one colour vector (here 24), to encode $\mathrm{Im}(G, C, P)$ the space is about $|V| \cdot (\log n + \log m + 3 + K)$.

3 Construction

In this section, we propose a new way to represent an image as a graph, by taking advantage of both image preprocessing tasks introduced in Section 1: the segmentation task, combined to interest pixels extraction. This allows us to obtain graphs that preserve the semantics and the topology of the original images, while providing us with a valid approximation that minimises the loss. Remember that the goal is to extract, given an image I, a graph G such that, with correct P and C functions, we have $\mathrm{loss}(I, \mathrm{Im}(G, P, C))$ as small as possible.

3.1 Segmentation

Any segmentation process aims at defining disjoint regions made of homogeneous sets of pixels (in some predefined way). Many segmentation algorithms exist, and it is generally possible to target an approximate number of regions by tuning the parameters of any chosen algorithm (see Fig. 1).

Aiming at minimising $\mathrm{loss}(I, \mathrm{Im}(G, P, C))$, it would be suitable to more or less preserve the regions of segmented image while building the graph: the boundaries between them should match, in as close a way as possible, the edges of the graph. We recall that two adjacent pixels are separated by a *linel* (see Section 2), so the regions are delimited by boundaries, sets of consecutive separating linels that form cycles. Of course, one given region may be delimited by more than one cycle, and the borders of the image need to be included.

Now our thesis is that the vertices of the graph should be selected among the endpoints of the separating linels described above, that is, the *pointels*. Indeed,

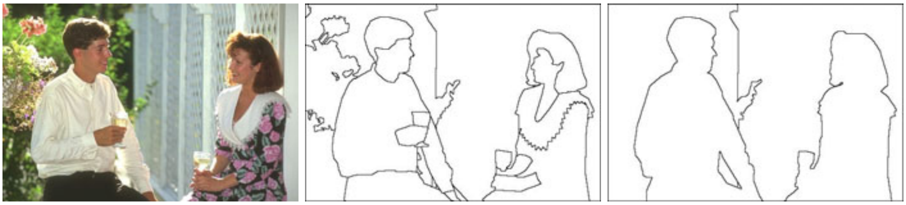


Fig. 1. An image and two segmentations at different levels, coming from the Berkeley Segmentation Dataset [15]

the vertices (and then the edges) must be chosen in such a way as to cover all the important elements structuring the boundaries. Obviously, too many vertices will make us deal with large graphs (which for combinatorial reasons is not recommended). Conversely, too few vertices will result in embedded edges (straight lines) which will be far apart from the boundaries. This means that we have to compress the information displayed in the segmented regions (which was already a compression of the original image): we are not interested in the exact shape of the regions, but in a rough although lifelike sketch (see Fig. 3 w.r.t. Fig. 1).

3.2 Extracting and Cleaning Up Interest Pixels

In order to select the vertices of the graph, we now make use of an interest pixels detector. No consensus exists on what an interest pixel is: each detector (e.g., [8,13]) performs its own measures of local information (based on texture, colour, shape,. . .) on the image and then extracts most stable and rich sets of pixels w.r.t. these features. While usually applied on original images, we run interest pixels detectors on segmented images. Indeed, segmented images are simplified, and detected pixels will thus be situated on, or close to, boundaries. This will prevent us from detecting pixels on textured areas, salient corners pixels situated inside a region, or pixels corresponding to noise. Note that if the number of extracted interest pixels is low, some regions (e.g. regions strictly included in others) may not be detected and thus may not be represented in the final graph.

Despite these precautions, the set of pixels obtained is generally not satisfying for two reasons. On the one hand, due to the principles of detectors, it is still possible for a pixel to be selected while it is far from any region boundary. We aim at eliminating them: for each interest pixel, we consider a mask containing this pixel and its eight neighbours, and then remove the interest pixel from the set if all the pixels in the mask belong to the same region.

On the other hand, the detector is often not able to come up with all the pixels that act as boundaries to three regions or more (i.e. pixels that belong to one region and have at least two other regions among their eight neighbours). Yet these pixels are necessary in order to have embedded edges of the graph close to the boundaries, so we aim at adding them: we consider 3x3 masks again, this time for all the pixels of the image, and add as interest pixels those whose mask intersect three different regions (or at least two, if the pixel is on image's border). We finally add the four corners of the image to the set.

We are clearly now in possession of a “clean” set of interest pixels that could be the vertices of the graph. However, we are still faced with a problem, well-known to researchers in discrete geometry: when tracing the lines corresponding to the edges of such a graph, because of the discrete nature of the pixels, we can end up with zones that are not colourable, or even to edges which intersect. Thus the vertices of the graph should not be pixels but pointels (with null thickness).

3.3 From Pixels to Pointels

In order to define the vertices, we now draw a parallel between linels separating segmented regions and interest pixels. Since linels lie between pixels, we will

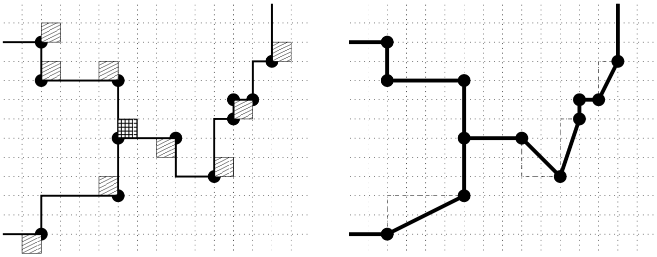


Fig. 2. On the left, the border between 3 regions: 9 (hatched) interest pixels + 1 (striped) intersection pixel yield 12 interest pointels (big dots). On the right, the extracted plane graph: the edges are built following regions boundaries.

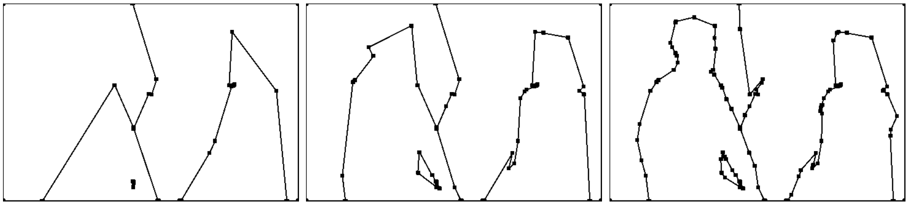


Fig. 3. Several plane graphs obtained from rightmost segmentation in Fig. 1. For the leftmost (resp. middle, rightmost) graph, 25 (resp. 50, 100) interest pixels were detected (and none of them deleted using the uniform mask rule), then 37 (resp. 36, 36) intersection pixels were added, generating 68 (resp. 108, 194) vertices (pointels).

analyse the four corners (pointels) of each interest pixel. Then the goal is to select among these pointels, those which are the most relevant. The rule is as follows: (1) Consider each pointel π of each interest pixel separately; (2) Let p_0, p_1, p_2 and p_3 be the 4 pixels which share pointel π; (3) If $\exists p_i$ such that $c_I(p_{i-1}) \neq c_I(p_i)$ and $c_I(p_i) \neq c_I(p_{i+1})$ (modulo 4) then the pointel is selected.

At this stage, the selected pointels become the vertices of the graph. Note that function P is also defined straightforwardly (we use coordinates of the pointels in the image). They are yet to be related: the edges are built by linking vertices, following the boundaries of each region (see Fig. 2 and 3).

3.4 Triangulation

To complete our method, we finally make use of a kind of Delaunay triangulation. The reason is threefold. (1) Such a technique will improve the robustness of the graph: if the positions of the vertices are slightly modified by transformations applied to the original image, the Delaunay triangulation will remain stable. (2) In a problem such as the plane graph isomorphism discussed in Section 2, triangulation edges will prevent one from matching faces with similar boundaries but different shapes. Note that the benefit of Delaunay triangulation has been investigated for other problems in Pattern Recognition [7]. (3) The first stage of our technique is a segmentation. Yet the resulting regions may be strictly included one into another (see Fig. 1). On the other hand, in the last stage, the edges are built following the boundaries of the regions. So the graph we get is generally not connected (as shown in Fig. 3), which is not combinatorially suitable; the use of the Delaunay triangulation will settle this issue.

Now, in order to preserve existing edges, a Delaunay triangulation takes place in each face of the embedded graph independently of the others. As a region may not be convex and often contains other included regions, the triangulation edges that are not totally included in the region are eliminated. Finally, we merge together the triangulated faces, and obtain the target plane graph G.

4 Experiments

The public benchmark of the Berkeley Segmentation Dataset [15] contains 300 images (481x321 RGB) of natural scenes, coming from the Corel dataset (widely used in computer vision). Each image contains at least one discernible object, and was segmented at varying levels by several individuals without any particular instruction on the type of criterion to use (see Fig. 1 for an example). The number of regions is at least 2, for an average of 20. As for the interest pixels detector, we have used the one from [1], based on multi-resolution contrast information. Below, I denotes an image, I_k its segmentation into k regions, and $\mathrm{Bg}(I_k, s)$ the graph obtained by extracting s interest pixels from the segmented image and following all the steps described in Section 3 (Bg=*Build Graph*).

4.1 Losses Due to the Graph Extraction Process

Firstly, we have evaluated $\mathrm{loss}(I, I_k)$, that is the loss due to the segmentation of I into k regions. For the 1633 available segmentations, the average loss is 10.0%. We also wanted to study its correlation with the number of regions. So, for regions ranging from 3 to 30, we selected 10 images (for a total of 280 segmentations) and computed the corresponding loss, together with the standard deviation. The results are shown on the left of Fig. 4: no correlation can be found. As the individuals who divided the images did not have any particular instructions, the segmentation was mostly semantic. Regions thus depict objects or parts of objects, and they are not necessarily homogeneous in terms of colours, whatever the level of segmentation.

Next, we have studied how the number of interest pixels affected the loss from I_k (the image segmented into k regions) to $\mathrm{Bg}(I_k, s)$ (the graph obtained by extracting s interest pixels from I_k). To this extent, we have extracted 10 to 500 interest pixels and built corresponding graphs. The results are shown on the right of Fig. 4. The loss ranges from 3.1% for 10 interest pixels to 0.2% for 500. Basically, the larger the extracted graph is, the closer the boundaries of the regions are followed by the edges, thus the lower $\mathrm{loss}(I_k, \mathrm{Bg}(I_k, s))$ is.

Finally note that the processing of interest pixels induces widely less loss than the segmentation stage ($< 3\%$ w.r.t. 10%). To confirm this, we have considered

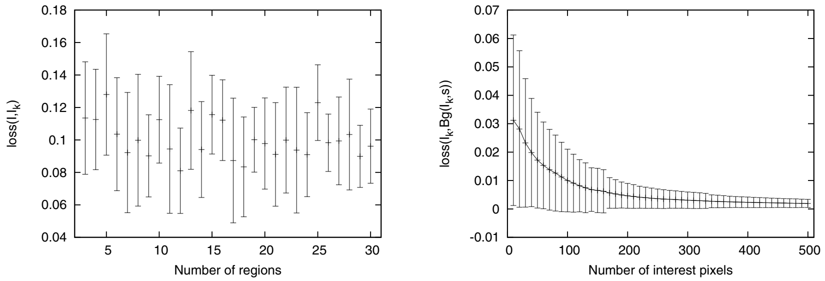


Fig. 4. loss(I, I_k) w.r.t. the number of regions (on the left) and loss$(I_k, \mathrm{Bg}(I_k, s))$ w.r.t. the number of interest pixels (on the right)

Table 1. For one image, losses corresponding to different levels of segmentation

s	k	$\lvert V\rvert$	loss(I, I_k))	loss$(I_k, \mathrm{Bg}(I_k, s))$	loss$(I, \mathrm{Bg}(I_k, s))$
50	3	104	0.084	0.006	0.089
50	7	123	0.084	0.005	0.089
50	9	175	0.084	0.006	0.090
50	13	149	0.084	0.007	0.091
50	20	207	0.083	0.007	0.090

one particular image at different levels of segmentation k and extracted a fixed number of interest pixels. See Table 1.

4.2 Comparison with Another Plane Graph Extraction Method

In order to compare our graph extraction method, we use another more classic one: interest pixels are directly extracted from original images (without pre-segmentation), then converted into interest pointels as described in Sect. 3.3 (with, added, the 4 corners pointels in order to fit whole image) and related by the Delaunay triangulation. We denote $\mathrm{Bg}(I, t_s)$ this graph and compare loss$(I, \mathrm{Bg}(I, t_s))$ with loss$(I, \mathrm{Bg}(I_k, s))$ (see Fig. 5): our method induces slightly less loss, while providing a graph that preserves the semantics of original image.

4.3 Size of the Graph

Given an image, in order to obtain a graph of a given size $|V|$, we can work on the segmentation level and the number of extracted pixels. To assess the trade-off between both, we compare loss$(I, \mathrm{Bg}(I_k, s))$ for different values of k and s. Of particular interest is the case where $\mathrm{Bg}(I_k, s)$ and $\mathrm{Bg}(I_{k'}, s')$ have similar sizes.The situation for 2 images A and B is depicted in Table 2. loss$(I, \mathrm{Bg}(I_k, s))$ is quite similar whether we are considering an important number of interest pixels and a small number of regions, or conversely. Thus any of parameters k and s can be tuned to obtain of graph of a given size, with a slight preference for the first due to the decrease of loss$(I_k, \mathrm{Bg}(I_k, s))$ when s becomes larger.

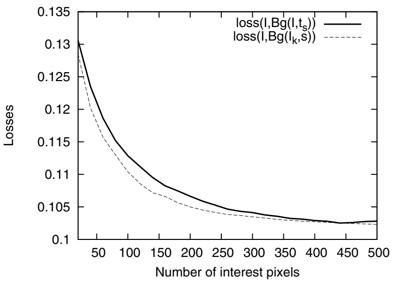


Fig. 5. loss$(I, \mathrm{Bg}(I, t_s))$ and loss$(I, \mathrm{Bg}(I_k, s))$ w.r.t. the number of interest pixels

Table 2. Influence of k and s on the losses for extracting a graph of size $|V|$

Image	s	k	$\|V\|$	loss(I, I_k)	loss$(I_k, \mathrm{Bg}(I_k, s))$	loss$(I, \mathrm{Bg}(I_k, s))$
A	80	7	182	0.084	0.004	0.088
	20	18	188	0.076	0.023	0.098
B	80	11	146	0.166	0.009	0.175
	20	25	145	0.166	0.022	0.188

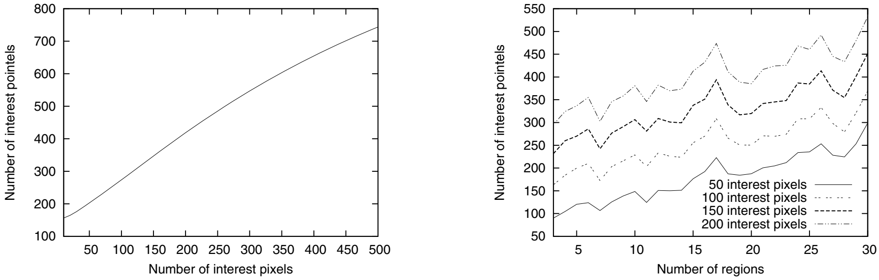


Fig. 6. Number of pointels w.r.t. the number of interest pixels (on the left) and to the number of regions (on the right)

More precisely, to assess the influence of the number s of detected interest pixels on $|V|$, we extracted 10 to 500 interest pixels from 1633 segmented images, added the intersection pixels and computed the corresponding pointels; left-hand curve of Fig. 6 shows that $|V|$ is a quasi-linear function of s.

Finally, concerning the influence of the level of segmentation k on $|V|$, we selected 10 segmentations for a number of regions varying from 3 to 30 (yielding 280 segmentation files); for each of them, we extracted 50, 100, 150 and 200 interest pixels and computed the corresponding pointels (see right of Fig. 6): Even if the curves do not increase steadily, for some fixed s, the more regions there are, the more selected pointels there are; this is due to the intersection pixels, that grows up with the number of regions.

Finally, even though the method is not intended as a compression algorithm, it can be noted that the original images have a theoretical size of 450kB (encoded as JPEG, they are about 70kB). The segmented files have an average size of 28kB, and a graph with 150 vertices requires less than 1kB.

5 Discussion, Open Questions and Conclusion

The bottleneck problem to using graphs in pattern recognition for image recognition tasks is to extract good graphs, where *good* means both reasonably small and semantically stable. We have introduced a new method for extracting graphs from images, which compresses the original information while preserving the semantics. Experiments show that the loss due to this process is limited. Moreover, these graphs have nice properties that are interesting for isomorphism-related problems: they are planar and connected, and their size is low. Moreover, this method is general enough to be applied to any kind of images (while usual methods are often adapted to particular classes of images (e.g see [16]).

It would be interesting to study alternative loss functions, to experiment with different segmentation algorithms and other ways of creating connexity and structural rigidity. Finally, we aim at proving the validity of the extracted graphs in pattern recognition tasks, and contributing to the field with a free distribution of the described software.

References

1. Bres, S., Jolion, J.-M.: Detection of interest points for image indexation. In: Huijsmans, D.P., Smeulders, A.W.M. (eds.) VISUAL 1999. LNCS, vol. 1614, pp. 427–434. Springer, Heidelberg (1999)
2. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. Pattern Recogn. and Artificial Intell. 18(3), 265–298 (2004)
3. Cori, R.: Un code pour les graphes planaires et ses applications. In: Astérisque, vol. 27, Soc. Math. de France, Paris, France (1975)
4. Damiand, G., de la Higuera, C., Janodet, J.-C., Samuel, E., Solnon, C.: A polynomial algorithm for subisomorphism of open plane graphs. In: MLG 2009 electronic proceedings (2009)
5. Damiand, G., de la Higuera, C., Janodet, J.-C., Samuel, E., Solnon, C.: A polynomial algorithm for submap isomorphism: Application to searching patterns in images. In: Torsello, A., Escolano, F., Brun, L. (eds.) GbRPR 2009. LNCS, vol. 5534, pp. 102–112. Springer, Heidelberg (2009)
6. Fàry, I.: On straight-line representation of planar graphs. Acta Scientiarum Mathematicarum 11, 229–233 (1948)
7. Finch, A.M., Wilson, R.C., Hancock, E.R.: Matching delaunay graphs. Pattern Recognition 30(1), 123–140 (1997)
8. Harris, C., Stephens, M.: A combined corner and edge detection. In: Proceedings of the Fourth Alvey Vision Conference, pp. 147–151 (1988)
9. Jiang, X., Bunke, H.: Optimal quadratic-time isomorphism of ordered graphs. Pattern Recognition 32(7), 1273–1283 (1999)
10. Kandel, A., Bunke, H., Last, M. (eds.): Applied Graph Theory in Computer Vision and Pattern Recognition. Studies in Computational Intelligence, vol. 52. Springer, Heidelberg (2007)

11. Kropatsch, W., Macho, H.: Finding the structure of connected components using dual irregular pyramids. In: Proc. DGCI 1995, pp. 147–158 (1995)
12. Lienhardt, P.: Topological models for boundary representation: a comparison with n-dimensional generalized maps. Computer-Aided Design 23(1), 59–82 (1991)
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
14. Lozano, M.A., Escolano, F.: Protein classification by matching and clustering surface graphs. Pattern Recognition 39(4), 539–551 (2006)
15. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. 8th Int. Conf. Computer Vision, July 2001, vol. 2, pp. 416–423 (2001) `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/`
16. Riesen, K., Bunke, H.: IAM graph database repository for graph based pattern recognition and machine learning. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 287–297. Springer, Heidelberg (2008)
17. Riesen, K., Bunke, H.: Approximate graph edit distance computation by means of bipartite graph matching. Image Vision Comput. 27(7), 950–959 (2009)
18. Rosenfeld, A.: Adjacency in digital pictures. Infor. and Control 26(1), 24–33 (1974)
19. Santo, M.D., Foggia, P., Sansone, C., Vento, M.: A large database of graphs and its use for benchmarking graph isomorphism algorithms. Pattern Recognition Letters 24(8), 1067–1079 (2003)
20. Tutte, W.: A census of planar maps. Canad. J. Math. 15, 249–271 (1963)

Indexing Tree and Subtree by Using a Structure Network

Mingming Zhang and Shinichiro Omachi

Graduate School of Engineering, Tohoku University
Aoba 6-6-05, Aramaki, Aoba-ku, Sendai-shi, 980-8579 Japan
johnson@aso.ecei.tohoku.ac.jp, machi@ecei.tohoku.ac.jp

Abstract. In pattern recognition, graphs become alluring more and more as structural pattern representations due to their richer representability than feature vectors. However, there are many challenging problems using graphs for pattern recognition. One is that it is difficult to investigate the relationships of graphs effectively, even of trees. In this paper, we focus on the structure relationship analysis of trees, such as tree and subtree isomorphism, maximum common subtree, minimum common supertree, etc., which is almost suffered from all kinds of tree recognition problems. For investigating the relationships of structures of trees, we propose a structure network to represent the evolutional relationships of structures of trees. Moreover, for a lot of tree isomorphism problems appearing in the application of structure network, we propose a method that encodes the structure of tree as a numerical sequence, and illustrate its efficiency by comparing it with traditional matching method for tree isomorphism problem.

Keywords: tree isomorphism, subtree isomorphism, tree indexing, structure analysis, tree measure.

1 Introduction

In pattern recognition and computer vision, the graph representation of objects has become alluring more and more along with the increase of the needs to consider the context of features. In a graph, not only the traditional features can be represented as vertices, but also the relationships defined on them can be represented as edges. Especially the tree as the simplest graph is used widely in many studies, such as data mining, computational biology, image analysis, document analysis, even in automatic theorem proving and compiler optimization [1].

However, when we are enjoying the advantages brought from the graph representation, we are suffering from some difficult problems at the same time. One is the high computational complexity; there exist many NP-complete or NP-hard problems in graph and tree problems [2]. Moreover, many efficient traditional methods working on the feature vectors can not be used on the graph and tree patterns straightforwardly [3]. Since matching graphs or trees directly is impractical when considering their computational complexity, many approximation methods are discussed, such as tree edit distance [1,4], spectrum-based

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 244–253, 2010.

methods [5,6], common supergraph methods [7], and kernel methods [8]. In all of these methods, the tree edit distance methods are studied very widely in practices [9,10], and are extended to kernel methods [11], pair HMM methods [12], etc. On the other hand, in order to convert the tree classification problems to traditional classification problems, some approaches that embed trees into vector space, based on the dissimilarity computed by edit distance [13,14] or based on the spectrum of tree [15] had been proposed. Furthermore, by selecting a number of candidates from database with indexing methods [16,17] before matching them one by one, the times of matching can be reduced greatly.

In all of these problems of trees, a basic problem is the relationship analysis of tree structures, such as isomorphism, subtree or supertree, minimum common supertree, maximum common subtree, etc. Even more, in edit distance algorithm, as a sub-problem in the dynamic program, it is necessary to investigate all the matching possibility of common subtrees. Although tree isomorphism can be determined by matching method, no method can measure the structure of tree by numerical value clearly. In this paper we focus on these tree problems, and try to address the relationships analysis of tree structures by defining a structure map. Furthermore we also propose a method for encoding the structure of a tree.

The rest of this paper is structured as follows. In section 2, we discuss how to construct the structure network for addressing the relationship analysis of tree structures. In section 3, by clustering the vertices in a tree, we give a method to encode the structure of tree that can be used to query a tree by its structure in database or in the structure network. In section 4, we state experiments for identifying the effectiveness of the proposed encoding method. Finally we will discuss some potential applications, and give our conclusion in section 5.

2 Structure Network

In this section, we discuss how to construct the structure map. First, we give some definitions and algorithms that are used throughout this paper. The trees considering in this paper are unlabeled and undirected trees. The graph isomorphism is defined as follow.

Definition 1. *Two graphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ are isomorphic if and only if there is a bijection φ between V_1 and V_2 such that for every pair of vertices $i, j \in V_1$, $(i, j) \in E_1$ if, and only if, $(\varphi(i), \varphi(j)) \in E_2$.*

In this paper, when two trees are isomorphic we will say that their structures are equal or they have the same structure. To determine whether two unlabeled trees T_1 and T_2 are isomorphic, we use the algorithm introduced by Kucera in [18] as shown bellow:

1. In each tree, label all the leaves with 1. If the numbers of leaves in T_1 and T_2 do not coincide, stop with “non-isomorphic”.
2. In each tree, determine the sets of unlabeled vertices S_1 and S_2 such that every neighbor of $v \in S_i$, except at most one, has a label. Tentatively, label

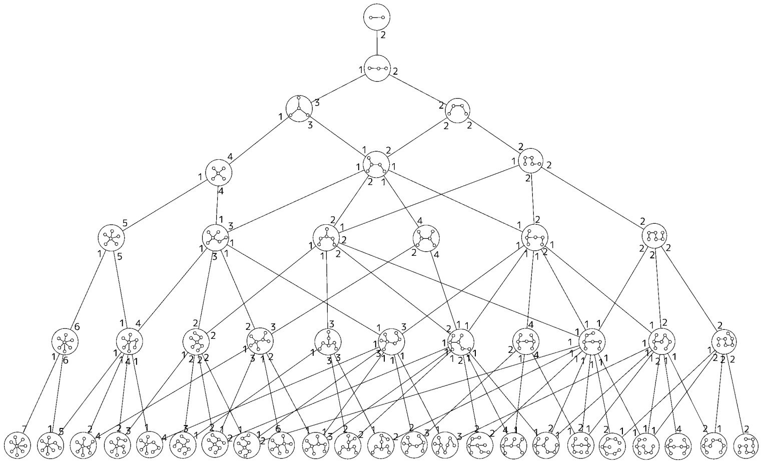


Fig. 1. Structure map under 8 vertices

v with the list $(l_1, \cdots, l_k)$ of labels of its labeled neighbors, sorted in non-decreasing order. Compare the respective tentative labels for the vertices in S_1 and S_2. If these labels do not agree (as multi-sets), stop with "non-isomorphic".

3. Substitute the tentative labels (which are lists of numbers) by new labels which are just numbers $k, k+1, k+2, \cdots$ in an arbitrary way. The only restriction is that vertices with the same tentative labels should get the same new labels, and that the labels $k, k+1, k+2, \cdots$ have not been used before.
4. If not all vertices have labels, go back to step 2.
5. Stop with "isomorphic".

As stated in section 1, usually, except for determining the isomorphism problem between two trees, we also need to deal with other relationships between trees. Although for subtree isomorphism problem Shamir gives an efficient matching method in [19], matching every pair of trees in a database is a very hard work when only using matching methods. So in this section we propose a method to construct a structure network, and discuss how to joint a structure of tree into this network. An example of this structure network under 8 vertices is shown in Fig. 1. In the structure network, each structure is drawn as a node that can indicate a cluster of trees with the same structure. All nodes of the network are organized into layers by their size. The edge indicates a sub-super relationship and only the adjacent layers can be linked. The numbers around each node will be discussed later.

There are two ways to construct the structure network, one is a top-down way that evolve the network from a two vertices structure. Another is a bottom-up way. When a tree is given, if its structure is outside of this network, then decompose it bottom-up until it can be completely connected into the network.

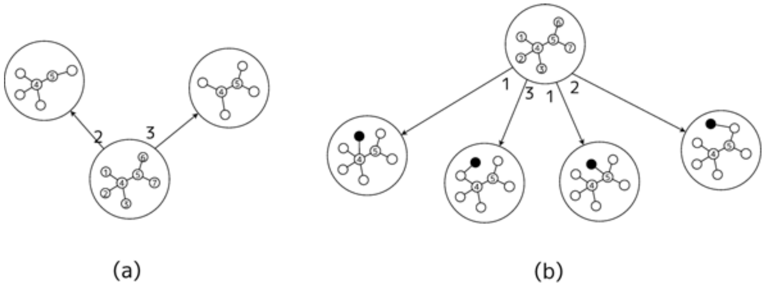


Fig. 2. The probability of reduction(a) and growth(b) of a topolgy

For a node in the network, there are two related processes; one is reduction: by removing a leaf in it, it can reduce to a sub-structure, and the other one is growth: by inserting an edge into a structure, it can grow up to a super-structure. First, let us consider the number that how many sub-structures can reduce to from a structure and the number that how many super-structures can grow up to. In general, the number of sub-structure that can reduce to is equal to the number of categories of its leaves. Inversely, the number of super-structures that can grow up to is equal to the number of categories of all vertices. For example, considering the structure shown in Fig. 2, (a) shows the reduction, and (b) shows the growth. In this case all of the vertices can be clustered into 4 categories, and two of them are leaves.

$$\{1, 2, 3\}, \{4\}, \{5\}, \{6, 7\}$$

For the clusters of leaves, their sizes are 3 and 2, which indicate the probabilities reducing to these two kinds of sub-structure when removing a leaf randomly. Similarly, the size of each cluster indicates the probability growing to each super-topology when inserting an edge randomly. Here we can cluster vertices by using their histogram of graph distance. Firstly for a vertex compute the graph distances from it to all the other vertices, and use a histogram diagram to count the distribution of distances. Such histograms of vertices in Fig. 2 are shown in Fig. 3. The histograms of vertices 1,2,3 are (a), 4 is (b), 5 is (c), 6,7 are (d). Then by using these histograms we can cluster vertices and find out the super-structures and sub-structures in up and down layers efficiently.

The top-down algorithm to construct the structure network until size N is shown as bellow. $T^{(N)}$ denotes a structure with N vertices.

```
initialize the structure network $S = \phi$;
initialize the two-vertices structure $t^{(2)}$ and add it into $S$;
for each $i$ from 2 to $N - 1$; do
    for each structure $t_{sub} \in T^{(i)}$, where $T^{(i)} \subseteq S$; do
        cluster all vertices in $t_{sub}$;
        for each cluster $C$ do
            select an arbitrary vertex $v \in C$;
```

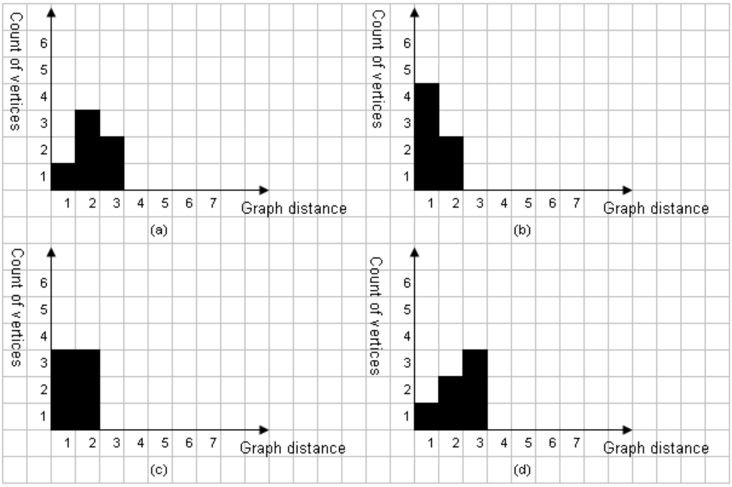


Fig. 3. Four kinds of histogram of vertex graph distance

```
          add an edge connecting to $v$ to construct a new structure $t_{new}$;
          look for the isomorphic structure $t_{sup}$ of $t_{new}$, where $t_{sup} \in T^{(i+1)}$
            and $T^{(i+1)} \subseteq S$;
          if found then
              link the structure $t_{sub}$ to $t_{sup}$;
          else
              add $t_{new}$ into $S$;
              link the structure $t_{sub}$ to $t_{new}$;
          end if
        done;
    done;
done;
```

When a given structure indicated by $t^{(N)}$ is not in the network S, a bottom-up algorithm used to joint it into the structure network is:

```
look for the isomorphic structure $t_{sup}$ of $t^{(N)}$, where $t_{sup} \in T^{(N)}$ and $T^{(N)} \subseteq S$;
if found then
    return $t_{sup}$;
else
    add a new cluster into $S$ and use $t^{(N)}$ to indicate it;
    for each $t_{sup} \in S$ where $t_{sup}$ have no link to the structure in $T^{(Size(t_{sup})-1)}$;
      do
        cluster all leaves in $t_{sup}$;
        for each cluster $C$; do
            select an arbitrary leaves $v \in C$;
            remove $v$ to construct a new structure $t_{new}$;
            look for the isomorphic structure $t_{sub}$ of $t_{new}$, where $t_{sub} \in T^{(size(t_{new}))}$
              and $T^{(size(t_{new}))} \subseteq S$;
            if found then
                link the structure $t_{sup}$ to $t_{sub}$;
            else
                add $t_{new}$ into $S$;
                link the structure $t_{sup}$ to $t_{new}$;
            end if
        done;
    done;
end if;
```

3 Encoding a Structure

In this section, we focus on how to describe a structure in a numerical way. As described in section 2, generally, to determine the isomorphism of two trees the matching method given by Kucera in [18] is used. Although its time complexity is $O(n)$, since it is a matching method, in some applications such as querying a structure from a database, a large amount of matching is neccesary. For dealing with the structures of trees more efficiently, in this section we discuss how to encode the structure informations of trees. We noticed that the essence of the isomorphism algorithm in fact is a procedure of clustering vertices step by step. In another way, as shown in section 2, using the histogram of graph distances the vertices can be clustered well too. For example, let us consider two isomorphic trees shown in Fig. 4. In the isomorphism algorithm a label is computed and assigned to each node, and their correspondences are shown in Table 1. In the matching method, to decide these correspondent relationship the adjacent labels should also be considered.

In contrast, in a column in Table 1, we show their histograms of graph distances. In this way, the correspondent relationships between vertices are decided uniquely by their histograms. Furthermore, the difference from matching method is that we can use histograms to encode the topology of a tree. Due to the connectivity of the tree, the histogram is also connected. That is for an arbitrary

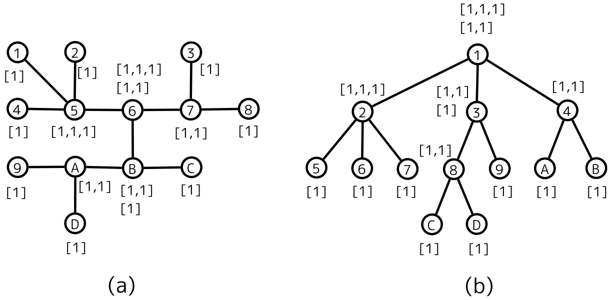


Fig. 4. Example of the isomorphic matching algorithm for Tree (a) and (b)

Table 1. The correspondence between vertices and their histograms

Vertices in Tree (a)	Vertices in Tree (b)	Label	Histogram
1,2,4	5,6,7	[1]	[1,3,2,4,2]
9,D	C,D	[1]	[1,2,2,2,5]
3,8	A,B	[1]	[1,2,2,5,2]
C	9	[1]	[1,2,4,5]
5	2	[1,1,1]	[4,2,4,2]
A	8	[1,1]	[3,2,2,5]
7	4	[1,1]	[3,2,5,2]
B	3	[[1,1],[1]]	[3,4,5]
6	1	[[1,1,1],[1,1]]	[3,7,2]

distance i, if ith element of the histogram is not zero, then the $(i-1)$th element is not zero too. So we can use a sequence to denote the histogram only including the nonzero element namely histogram sequence. Using this histogram sequence we can encode each cluster to a field defined as:

$$\begin{aligned} h &:= \text{histogram sequence;} \\ l &:= \text{lengh of } h; \\ s &:= \text{size of cluster;} \\ field &:= (l+2), s, h; \end{aligned}$$

Finally, sort all cluster fields alphabetically, and link them to compose a long sequence. This long sequence is the final result of encoding the structure of a tree, and it includes all the clustering informations of vertices obviously. For example, the structure of the tree in Fig. 4 can be encoded as

$$\begin{aligned} (\ &\mathbf{5}, 1, 3, 4, 5, \mathbf{5}, 1, 3, 7, 2, \mathbf{6}, 1, 1, 2, 4, 5, \mathbf{6}, 1, 3, 2, 2, 5, \mathbf{6}, 1, 3, 2, 5, 2, \\ &\mathbf{6}, 1, 4, 2, 4, 2, \mathbf{7}, 2, 1, 2, 2, 2, 5, \mathbf{7}, 2, 1, 2, 2, 5, 2, \mathbf{7}, 3, 1, 3, 2, 4, 2). \end{aligned}$$

In the algorithm proposed in section 2, by replacing the matching method to the encoding method where the step is looking for isomorphic structure, the algorithm could be more efficient.

4 Experiments

In this section, we will verify the effectiveness of the encoding method and discuss the indexing tree and subtree by structure network. We first create a database that contains about 20,000 unlabeled and undirected trees with up to 20 vertices.

4.1 Experimental Settings

In this section, we address the algorithm used to create the database of trees. Here we use $T^{(j)}$ to denote the set of j vertices trees, and $T_i^{(j)}$ denotes its element. The algorithm is:

```
prepare $T^{(2)}$;
foreach $j$ from 2 to 20 do
    if $|T^{(j)}| * (j+1) < 1500$ then
        foreach $T_i^j$ do
            foreach vertex $v \in V(T_i^j)$ do
                add an edge adjacent to $v$ to construct a new tree and save it;
            done
        done
    else do
        while $|T^{(j+1)}| < 1500$ do
            select a $T_i^{(j)}$ randomly;
            select a vertex $v \in V(T_i^{(j)})$ randomly;
            add an edge adjacent to $v$ to construct a new tree and save it;
        done
    done
done
```

With this algorithm, we made a tree database including 19,410 trees. These trees are clustered into 8,251 categories by using the isomorphism algorithm. In these clusters, 2,898 clusters which include more than one tree.

4.2 Results

Firstly, we cluster all the trees in database by using the proposed encoding method. Then the result is compared with the one of the isomorphism algorithm. As shown in Table. 2, the clustering result of the proposed encoding method is the same as the isomorphism algorithm. In concern with the computing complexity, in an LAM/MPI environment with 10 nodes(CPUs), clustering this database by using isomorphism algorithm spent about 9.5 days. In contract, computing all the encoding sequence only needs 45 minutes.

For the structure network, because constructing it is based on the encoding method or isomorphism algorithm, the correctness of clustering is clear. With it, we discuss the time and spatial complexity of construction. In Fig. 5 we show three kinds of rough statistic data of constructing the structure network of under 15 vertices by using top-down algorithm. In Fig. 5 (a) we show the raise of the number of nodes and edges along with the size of tree. In Fig. 5 (b) the computing time is shown. Although it is expensive to construct the structure network, once the network is be constructed offline, to investigate the relationships between trees can be achieved by only using shortest path algorithm. Moreover, by using the bottom-up algorithm, it is possible to construct the structure network partially.

Table 2. The results of experiments

proposed method	Encoding Index
correct rate	100%

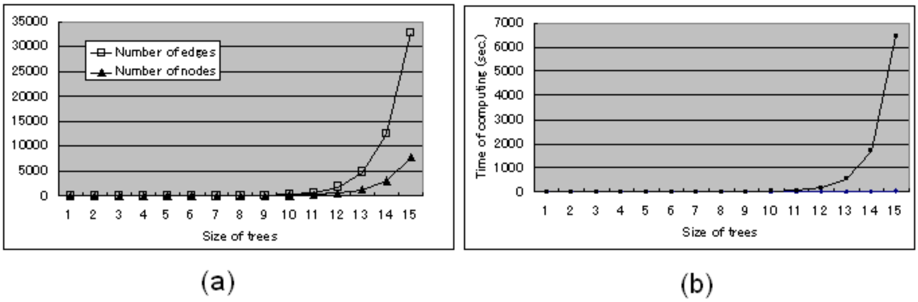


Fig. 5. Complexity of constructing the structure network:(a) the number of nodes and edges (b) the computing times

5 Conclusion

In this paper, we proposed a method to construct a structure network for indexing tree and subtree, and based on the clustering vertices of a tree, we proposed an encoding method to indicate the structure of a tree as a numerical sequence for indexing. As shown in the results of experiment, we can conclude the effectiveness of the proposed methods. Furthermore, tree edit distance algorithm is exploited widely in the applications as stated in section 1. Since each edit operator in tree edit distance algorithm is related to a link in the structure network, by defining proper cost functions on the edges and nodes of structure network it is possible to find out the optimal edit path efficiently on the structure network for two trees. On the other hand, since each edge of structure network is related to two probabilities according to two different directions, it is possible to compute a conditional probability between two structure and the joint probability of them by proability propagation. For future work, first we are going to investigate the encoding method in detail for a more compact index. In application side, we will try to combine the structure network with the tree edit distance to improve the efficiency of tree edit distance algorithm or to develop the structure network to a Bayesian network for recognition of tree patterns.

References

1. Bille, P.: A survey on tree edit distance and related problems. Theoretical Computer Science 337(1-3), 217–239 (2005)
2. Köbler, J., Schöning, U., Torán, J.: The graph isomorphism problem: its structural complexity. Springer, Heidelberg (1993)
3. Bunke, H., Irniger, C., Neuhaus, M.: Graph matching - challenges and potential solutions. In: Roli, F., Vitulano, S. (eds.) ICIAP 2005. LNCS, vol. 3617, pp. 1–10. Springer, Heidelberg (2005)
4. Klein, P.N., Sebastian, T.B., Kimia, B.B.: Shape matching using edit-distance: an implementation. In: SODA, pp. 781–790 (2001)

5. Wilson, R.C., Zhu, P.: A study of graph spectra for comparing graphs and trees. Pattern Recognition 41(9), 2833–2841 (2008)
6. Wilson, R., Hancock, E., Luo, B.: Pattern vectors from algebraic graph theory. IEEE Ttansactions on Pattern Analysis and Machine Intelligence 27(7), 1112–1124 (2005)
7. Bunke, H., Foggia, P., Guidobaldi, C., Vento, M.: Graph clustering using the weighted minimum common supergraph. In: Hancock, E.R., Vento, M. (eds.) GbRPR 2003. LNCS, vol. 2726, pp. 235–246. Springer, Heidelberg (2003)
8. Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized kernels between labeled graphs. In: Fawcett, T., Mishra, N. (eds.) ICML, pp. 321–328. AAAI Press, Menlo Park (2003)
9. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing their shock graphs. IEEE Trans. Pattern Anal. Mach. Intell. 26(5), 550–571 (2004)
10. Neuhaus, M., Bunke, H.: Self-organizing maps for learning the edit costs in graph matching. IEEE Transactions on Systems, Man, and Cybernetics, Part B 35(3), 503–514 (2005)
11. Neuhaus, M., Bunke, H.: Edit distance-based kernel functions for structural pattern classification. Pattern Recognition 39(10), 1852–1863 (2006)
12. Sakakibara, Y.: Pair hidden markov models on tree structures. In: ISMB (Supplement of Bioinformatics), pp. 232–240 (2003)
13. Torsello, A., Hancock, E.R.: Graph embedding using tree edit-union. Pattern Recognition 40(5), 1393–1405 (2007)
14. Riesen, K., Neuhaus, M., Bunke, H.: Graph embedding in vector spaces by means of prototype selection. In: Escolano, F., Vento, M. (eds.) GbRPR. LNCS, vol. 4538, pp. 383–393. Springer, Heidelberg (2007)
15. Bai, X., Hancock, E.R.: Heat kernels, manifolds and graph embedding. In: [20], pp. 198–206
16. Shokoufandeh, A., Dickinson, S.J., Siddiqi, K., Zucker, S.W.: Indexing using a spectral encoding of topological structure. In: CVPR, pp. 2491–2497. IEEE Computer Society, Los Alamitos (1999)
17. Irniger, C., Bunke, H.: Decision tree structures for graph database filtering. In: [20], pp. 66–75
18. Kučera, L.: Combinatorial algorithms. Taylor & Francis, Abington (1990)
19. Shamir, R., Tsur, D.: Faster subtree isomorphism. J. Algorithms 33(2), 267–280 (1999)
20. Fred, A., Caelli, T.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.): SSPR&SPR 2004. LNCS, vol. 3138. Springer, Heidelberg (2004)

Attributed Graph Matching for Image-Features Association Using SIFT Descriptors

Gerard Sanromà[1], René Alquézar[2], and Francesc Serratosa[1]

[1] Departament d'Enginyeria Informàtica i Matemàtiques,
Universitat Rovira i Virgili, Tarragona, Spain
gerard.sanroma,francesc.serratosa@urv.cat

[2] Institut de Robtica i Informtica Industrial, CSIC-UPC, Barcelona, Spain
ralquezar@iri.upc.edu

Abstract. Image-features matching based on SIFT descriptors is subject to the misplacement of certain matches due to the local nature of the SIFT representations. Some well-known outlier rejectors aim to remove those misplaced matches by imposing geometrical consistency. We present two graph matching approaches (one continuous and one discrete) aimed at the matching of SIFT features in a geometrically consistent way. The two main novelties are that, both local and contextual coherence are imposed during the optimization process and, a model of structural consistency is presented that accounts for the quality rather than the quantity of the surrounding matches. Experimental results show that our methods achieve good results under various types of noise.

Keywords: attributed graph matching, SIFT, image registration, discrete labeling, softassign.

1 Introduction

Image-features matching based on Local Invariant Features Extraction (LIFE) methods has become a topic of increasing interest over the last decade. LIFE methods extract stable representations from a selected set of characteristic regions (features) of the image. These local representations are aimed to be invariant at a certain extent to image deformations such as changes in illumination, position of the camera, ... Mikolajczyk and Schmid [1] identified Lowe's SIFT descriptors [2] as the most stable representations among a number of approaches.

SIFT features are located at the salient points of the scale-space. Each SIFT feature retains the magnitudes and orientations of the image gradient at its neighboring pixels. This information is represented in a 128-length vector.

Despite its efficiency, image-features matching based on local information is still subject to the misplacement of certain matches. A number of approaches have been presented aimed at fixing these misplacements by removing incorrect associations with the use of higher-level information. To cite some examples, RANSAC [3] has been successfully applied to outlier rejection by fitting a geometrical model. More in the topic of the present paper, Aguilar et al. [4] have

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 254–263, 2010.

recently presented an approach that use a graph transformation to select a subset of geometrically consistent associations.

The main aim of our work is to fix the misplaced matches by relocating them when possible, so as to obtain a higher amount of useful matches than the outlier rejectors. We face this relocation as an attributed graph matching problem where we seek for the set of assignments that best fit the constraints imposed by both the local descriptors (attributes of the nodes) and the geometrical arrangement of the features (structural relations). One of the main contributions of this work is the development of a structural model of consistency that accounts, not only for edge-consistency, but also for the matching quality of the surrounding assignments.

We present two graph matching approaches. The first, presented in section 3, is cast in the continuous assignment framework provided by Softassign [5]. The second, presented in section 4, is a probabilistic model cast in a discrete labeling scheme [6]. We have evaluated the matching precision and recall of both methods under different sources of noise.

In section 2 some preliminary concepts are given. We present in section 5 comparative results with Aguilar et al.'s outlier rejector [4], RANSAC used to fit a fundamental matrix [7], and Luo and Hancock's structural graph matching approach [8]. Finally, in section 6 some conclusions are drawn.

2 Preliminaries

Consider an image I_M showing a certain scene. Consider another image I_D showing the same scene as I_M but with some random variations such as viewpoint change, illumination variation, nonrigid deformations in the objects of the scene, etc ... Consider two sets of SIFT features (keypoints) X, Y from the images I_D and I_M.

Definition 1. *According to SIFT matching, a keypoint i from I_D with descriptor (column) vector x_i and position inside the image $\left(p_1^{(i)}, p_2^{(i)}\right)$ is matched to a keypoint j from I_M iff:*

$$\frac{||x_i - y_j||}{||x_i - y_{i,2min}||} < \rho \tag{1}$$

where $||x|| = \sqrt{x^\top x}$ is the Euclidean length (L^2 norm), $y_{i,2min} \in Y$ is the descriptor with the second smallest distance from $x_i \in X$, and $0 < \rho \leq 1$ is a ratio defining the tolerance to false positives.

This is, a keypoint i from I_D is matched to the closest (in the descriptor-vector space) keypoint j from I_M if the ratio of their distance to the second smallest distance from i is below a certain value $0 < \rho \leq 1$. If this condition is not met, then keypoint i is leaved unmatched.

Definition 2. *We define a graph G_M representing a set of SIFT keypoints from the image I_M as a three tuple $G_M = (V_M, M, Y)$ where $v_\alpha \in V_M$ is a node* *associated to a SIFT keypoint with position* $\left(p_1^{(\alpha)}, p_2^{(\alpha)}\right)$, $y_\alpha \in Y$ *is the SIFT descriptor associated to node* v_α *and,* M *is the adjacency matrix (thus,* $M_{\alpha\beta} = 1$ *indicates that nodes* v_α *and* v_β *are adjacent and* $M_{\alpha\beta} = 0$ *otherwise).*

Consider also the graph $G_D = (V_D, D, X)$ that represent a set of keypoints from I_D.

Definition 3. *We define the probability of matching nodes* $v_a \in V_D$ *with* $v_\alpha \in V_M$ *with regards to the nodes' attributes with the following quantity*

$$P_{a\alpha} = \frac{\frac{1}{||x_a - y_\alpha||}}{\sum_{\alpha'} \frac{1}{||x_a - y_{\alpha'}||}} = \frac{1}{||x_a - y_\alpha|| \sum_{\alpha'} \frac{1}{||x_a - y_{\alpha'}||}} \tag{2}$$

which is a quantity proportional to the inverse of the distance between their descriptors (normalized to sum up to one).

Definition 4. *We define the threshold probability for sending a node* $v_a \in V_D$ *to* null *(i.e., leaving it unmatched) as*

$$P_{a\emptyset} = \frac{\frac{1}{\rho||x_a - y_{a,2min}||}}{\sum_{\alpha'} \frac{1}{||x_a - y_{\alpha'}||}} = \frac{1}{\rho||x_a - y_{a,2min}|| \sum_{\alpha'} \frac{1}{||x_a - y_{\alpha'}||}} \tag{3}$$

which places the threshold probability for $v_a \to \emptyset$ *at the distance* $\rho||x_a - y_{a,2min}||$ *which is the maximum distance permitted for an* v_α *to satisfy* $\frac{||x_a - y_\alpha||}{||x_a - y_{a,2min}||} < \rho$.

Note that the matching probabilities of equations (2) and (3) define the same matching criterion as definition 1.

It is a well-known strategy to state that a match from a node $v_a \in V_D$ to a node $v_\alpha \in V_M$ is more likely to occur as more nodes adjacent to v_a are assigned to nodes adjacent to v_α [5][8].

Definition 5. *We define a* hit *as a node* $v_b \in V_D$ *adjacent to* v_a *that is matched to a node* $v_\beta \in V_M$ *adjacent to* v_α.

In sections 3 and 4 we develop measures for gauging the structural consistency of a given match $v_a \to v_\alpha$. The novelty of the proposed measures lies on the fact that they do not rely on the quantity but on the quality of those *hits*.

Our approaches to attributed graph matching aim to estimate an assignment function $f : V_D \to V_M$ that best fits the criteria imposed by both the SIFT attributes (local constraints) and the structural relations of the graphs (contextual constraints). Accordingly, $f(a) = \alpha$ means that node $v_a \in V_D$ is matched to node $v_\alpha \in V_M$, and $f(a) = \emptyset$ means that it is not matched to any node.

Definition 6. *We define the assignment variable* S *such that* $s_{a\alpha} \in S$ *and* $s_{a\alpha} = 1$ *if* $f(a) = \alpha$ *and* $s_{a\alpha} = 0$ *otherwise. The assignment variable is subject to the constraints* $\forall a, \sum_\alpha s_{a\alpha} = \{0, 1\}$ *and* $\forall \alpha, \sum_a s_{a\alpha} = \{0, 1\}$.

This is, each node $v_a \in V_D$ can be assigned only to one node $v_\alpha \in V_M$.

In the following two sections we present a continuous and a discrete labeling approach to attributed graph matching.

3 A Continuous Labeling Approach

Graduated Assignment (Softassign) [5] is a well-known optimization algorithm that has been widely used to find suboptimal solutions to the graph matching problem. It estimates the assignment variable S that minimizes the following objective function:

$$\mathcal{F}(S) = -\frac{1}{2}\sum_{a}^{|V_D|}\sum_{\alpha}^{|V_M|}\sum_{b}^{|V_D|}\sum_{\beta}^{|V_M|} s_{a\alpha} s_{b\beta} C_{a\alpha b\beta} \tag{4}$$

where s_{ij} are the components of the assignment variable S (definition 6) and $C_{a\alpha b\beta}$ are the compatibility coefficients for the simultaneous associations $v_a \rightarrow v_\alpha$ and $v_b \rightarrow v_\beta$.

This measure originates from the relaxation labeling processes [9]. Gold and Rangarajan [5] have turned this minimization into an iterative assignment problem where a double stochastic matrix of continuous assignments $\tilde{\mathcal{S}}$ is updated at iteration $(n+1)$ according to the following expression

$$\tilde{\mathcal{S}}^{(n+1)} = \arg\max_{\tilde{S}} \sum_{a}^{|V_D|}\sum_{\alpha}^{|V_M|} Q_{a\alpha}\tilde{S} \tag{5}$$

where $Q_{a\alpha}$ is a quantity depending on the continuous assignment matrix of the current iteration $\tilde{S}^{(n)}$, and corresponds to the derivative of the objective function

$$Q_{a\alpha} = -\frac{\delta\mathcal{F}}{\delta\tilde{s}_{a\alpha}} = +\sum_{b}^{|V_D|}\sum_{\beta}^{|V_M|} \tilde{s}_{b\beta}^{(n)} C_{a\alpha b\beta} \tag{6}$$

where $0 \leq \tilde{s}_{ij} \leq 1$ is the (i,j) component of the continuous assignment matrix $\tilde{S}$. The assignment problem of equation (5) is solved in a continuous (soft) way using a continuation method controlled by a parameter to gradually push from continuous to discrete solutions (see reference [5] for more details about the algorithm).

We consider that a candidate association $v_a \rightarrow v_\alpha$ with a high probability regarding the local information but with low support from its surrounding matches is likely to be an outlier (i.e., a geometrically inconsistent association). In this case v_a should not be matched to v_α. On the other hand, a candidate association with a not-enough-high local probability (i.e., $P_{a\alpha} < P_{a\emptyset}$) but with high support from the surrounding matches, is likely to be an inlier. In that case v_a should be matched to v_α. We propose the following expression as it reflects this desired behaviour

$$T_{a\alpha} = \frac{P_{a\alpha}}{P_{a\emptyset}} + \frac{P_{a\alpha}}{P_{a\emptyset}}\left[\sum_{b}^{|V_D|}\sum_{\beta}^{|V_M|}\left(\frac{P_{b\beta}}{P_{b\emptyset}} D_{ab} M_{\alpha\beta}\tilde{s}_{b\beta}\right) - K_{\emptyset_1}\right] \tag{7}$$

where P_{ij}, $P_{i\emptyset}$ are the probabilities for matching $v_i \rightarrow v_j$, $v_i \rightarrow \emptyset$ regarding the nodes' attributes (equations (2), (3)) and; D and M are the adjacency matrices of G_D and G_M.

This measure is composed by a sum of two parts. The first part contributes with the matching quality regarding the nodes' local information, $\frac{P_{a\alpha}}{P_{a\emptyset}}$. This quotient is > 1 if $v_a \rightarrow v_\alpha$ is more likely than $v_a \rightarrow \emptyset$ in terms of local consistency and, ≤ 1 otherwise. The second part contributes with a quantity proportional to the sum of the quality of the *hits*. The *hits* are by definition the only terms of the double summatory different to zero (i.e., $\{(b, \beta) \mid D_{ab} = 1, M_{\alpha\beta} = 1, \tilde{s}_{b\beta} \neq 0\}$). The constant $K_{\emptyset_1}$ represents the threshold contribution required from the *hits* in order to boost the overall measure $T_{a\alpha}$. This second part can be interpreted in the following way: it is > 0 if $v_a \rightarrow v_\alpha$ is more likely than $v_a \rightarrow \emptyset$ in terms of contextual consistency and, it is ≤ 0 otherwise.

Consider the case of a candidate association with a high local and a low contextual consistency. Despite of the high quantity of the first part of $T_{a\alpha}$, the negative contribution of the second part would smooth the overall measure. In the case of a candidate association with a not-enough-high local and a high contextual consistency, the positive contribution of the second part would boost the overall measure.

We have to express $T_{a\alpha}$ in the same terms of $Q_{a\alpha}$ (equation (6)) so that we can use it under the framework of Softassign. In the following expression we rearrange the terms in order to express our measure in terms of the compatibility coefficients $C_{a\alpha b\beta}$

$$Q_{a\alpha} = \sum_{b}^{|V_D|} \sum_{\beta}^{|V_M|} \tilde{s}_{b\beta}^{(n)} \left[\frac{P_{a\alpha}}{P_{a\emptyset}} \left(\frac{P_{b\beta}}{P_{b\emptyset}} D_{ab} M_{\alpha\beta} + \frac{1 - K_{\emptyset_1}}{N} \right) \right] \tag{8}$$

where the expression between brackets corresponds to $C_{a\alpha b\beta}$ of equation (6) and $N = \sum_b \sum_\beta \tilde{s}_{b\beta}$ that is a number aproximately equal to the number of nodes of the graphs (due to the double stochastic nature of the assignment variable used in Softassign).

At the end of the algorithm the continuous assignment matrix $\tilde{S}$ is turned into a (discrete) assignment variable S such that all the nodes $v_a \in V_D$ are assigned to some $v_\alpha \in V_M$. Finally, we remove the assignments $s_{a\alpha} = 1$ with coefficients $Q_{a\alpha} < 1$ since they do not satisfy the combined constraints.

4 A Discrete Labeling Approach

The idea of discrete labelling [6] is to visit each node and update f in order to gain the maximum improvement in our matching criterion. The difference with other approaches such as softassign [5] or probabilistic relaxation [9] is that the assignment variable is discretely updated, not allowing for soft assignments.

We want to maximise the joint probability of a graph given the assignment function f. To do so, our iterative algorithm visits all the nodes of the graph at each iteration, and updates f in order to increase this joint probability

$$P\left(G_D | G_M, f\right) = \prod_{v_a \in V_D} P\left(v_a \rightarrow v_{f(a)} | f\right) \tag{9}$$

The update equation of the assignment function is

$$f(u) = \underset{\{\alpha=1\ldots|V_M|\}\bigcup\{\emptyset\}}{\arg\max} P\left(v_a \rightarrow v_\alpha|f\right) \tag{10}$$

(we use a cleaning heuristic in order to guarantee that f is an injective function).

We have designed our matching criterion as a product of the following two quantities:

$$P\left(v_a \rightarrow v_\alpha|f\right) = P_{a\alpha}R_{a\alpha} \tag{11}$$

where $P_{a\alpha}$ and $R_{a\alpha}$ stand for the matching probability according to the current node attributes and the structural relations, respectively. We use the multiplication to combine both quantities as we find it a natural way to combine probability measures as well as it doesn't need further parameters (as opposed to other operators such as linear weighting).

We use the expresion presented in equation (2) (and equation (3)) to gauge the likelihood, regarding the node's attributes, of the putative match $v_a \rightarrow v_\alpha$ (and $v_a \rightarrow \emptyset$, when appropriate).

In the remainder of this section we develop the matching likelihood for the association $v_a \rightarrow v_\alpha$ regarding the structural relations ($R_{a\alpha}$). Our aim is to define a model that takes into account the quality of the surrounding matches.

Luo and Hancock [8] showed how to factorise, using the Bayesian theory, the hard-to-model matching probability given the entire state of the assignments S into easy-to-model unary assignment probability terms:

$$P\left(v_a \rightarrow v_\alpha|S\right) = g_a \prod_{v_b \in V_D} \prod_{v_\beta \in V_M} p\left(v_a \rightarrow v_\alpha|s_{b\beta}\right) \tag{12}$$

where $g_a = \left[1/p(v_a)\right]^{|V_D||V_M|-1}$ is a constant only depending on node v_a.

The model for the unary assignment probabilities presented in [8] used the Bernoulli distribution in order to accomodate *hits* and no *hits* (definition 5) with fixed probabilities $(1 - P_e)$ and P_e (being P_e the probability of error). We present a new model aimed at giving a more fine-grained measure by assessing the *hits* according with their quality, while giving room for possible structural errors in the case of no *hit*. The proposed expression is

$$p\left(v_a \rightarrow v_\alpha|s_{b\beta}\right) = P_{b\beta}^{D_{ab}M_{\alpha\beta}s_{b\beta}} \left[\xi P_{b\emptyset}\right]^{\left(1-D_{ab}M_{\alpha\beta}s_{b\beta}\right)} \tag{13}$$

where D and M are the adjacency matrices of G_D and G_M, respectively; $P_{b\beta}$ is the quality term of the association $v_b \rightarrow v_\beta$ (eq. (2)) and, $[\xi P_{b\emptyset}]$ is the gound-level contribution in the case of *no hit* expressed in reference to $P_{b\emptyset}$ (eq. (3)). The parameter $0 < \xi \leq 1$ regulates the ground-level contribution. When $\xi \rightarrow 0$, there is small room for structural errors and then, the update equation (11) relies mostly on the structural model. On the other hand, when $\xi \rightarrow 1$, the ground-level approaches the quality term and the structural model becomes ambiguous.

In a similar manner that is done in [8], we state equations (12) and (13) in the exponential form, obtaining the following expression

$$R_{a\alpha} = h_a \exp\left[\sum_b \sum_\beta \log\left(\frac{P_{b\beta}}{\xi P_{b\emptyset}}\right) D_{ab} M_{\alpha\beta} s_{b\beta}\right] \tag{14}$$

where $h_a = \exp[\sum_b \sum_\beta \log(\xi P_{b\emptyset})] g_a$ is a constant that does not depend on either the graph structure or the state of the correspondences.

Finally, we define the threshold probability for sending a node $v_a \in V_D$ to *null* according to the structural relations as

$$R_{a\emptyset} = h_a \exp\left[K_{\emptyset_2} \log\left(\frac{1}{\xi}\right)\right] = h_a \exp\left[-K_{\emptyset_2} \log(\xi)\right] \tag{15}$$

where $K_{\emptyset_2} \geq 0, K_{\emptyset_2} \in \Re$ is a parameter defining the minimum number of *hits* with quality term $P_{b\beta} \geq P_{b\emptyset}$ required for the match $v_a \to v_\alpha$ to be more structurally likely than $v_a \to \emptyset$.

The algorithm operates updating the assignment function f as stated in equation (10) with the new combined constraints provided in equation (11). This is, at each iteration, each node $v_a \in V_D$ is assigned to the node $v_\alpha \in V_M$ with the highest probability. If the target with the highest probability is $\emptyset$, then v_a is leaved unmatched.

5 Experiments

We have compared both the continuous and the discrete attributed graph matching approaches of the present work (C-AGM and D-AGM) to the following approaches: Graph Transformation Matching (GTM) [4], RANSAC used to fit a fundamental matrix [7] and, Structural Graph Matching with the EM Algorithm (SGM-EM) [8]. We have evaluated the matching *Precision* and *Recall* scores of each method under the following types of perturbations: *image distortions*, *geometrical noise* and *clutter* (point contamination). We have used the *F-measure* to plot the results. F-measure is defined as the weighted harmonic mean of Precision (P) and Recall (R) and its expression is $F = (2 \times \mathrm{P} \times \mathrm{R}) / (\mathrm{P} + \mathrm{R})$.

The graphs used in our methods (C-AGM and D-AGM) have been generated as described in definition 2. Graph structures for all the methods using graphs (i.e., C-AGM, D-AGM, GTM and SGM-EM) have been generated using a K-nearest-neighbours approach with $K = 4$ (i.e., edges are placed joining a keypoint with its K nearest neighbours in space). All the methods have been initialized with the configuration of matches returned by a classical SIFT matching using a ratio $\rho = 1$ (the best value for the outlier rejectors). As the C-AGM method permits the use of continuous assignments we have initialized them with the probabilities due to local information (i.e., $s_{a\alpha} = P_{a\alpha}$). The keypoint-sets size used in the experiments has been $N = 20$. Our methods (C-AGM and D-AGM) have done 20 iterations, and we have used $\xi = 0.5$ (D-AGM) and $\rho = 1$ (C-AGM and D-AGM). We have empirically set $K_{\emptyset_1} = 0.6$ and $K_{\emptyset_2} = 2.3$ in the clutter experiments, and $K_{\emptyset_1} = K_{\emptyset_2} = 0$ in the others. The tolerance threshold for RANSAC has been set to 0.01, and the number of iterations to 1000 (as suggested in [7]). The probability of error P_e for the SGM-EM method has been set to 0.0003, and the number of iterations to 100.

For each experiment we have arbitrarly chosen a grayscale image I_0 from the Camera Movements and Deformable Objects' databases used in [4].

In the image distortion experiments, we generate I_1 by simultaneously applying the following types of perturbations to I_0: image resizing, to simulate changes in the distance from the objects in the image; image rotation, to simulate changes in viewpoint; image intensity adjustement, to simulate illumination changes and; gaussian white noise addition to pixel intensity values, to simulate deterioration in the viewing conditions.

We extract the SIFT keypoints from images I_1 and I_0, obtaining coordinate vector-sets $\mathbf{P}$ and $\mathbf{Q}$, and SIFT descriptor-sets X and Y, respectively. We define $\widetilde{\mathbf{P}}$ as the result of the mapping from points in $\mathbf{P}$ back to the reference of I_0. We compute $\widetilde{\mathbf{P}}$ by applying to $\mathbf{P}$ the inverse resizing and rotation from the perturbation. We set the ground truth assignments on the basis of the proximity between the points in $\mathbf{Q}$ and $\widetilde{\mathbf{P}}$. Then, for a given $\mathbf{q}_i \in \mathbf{Q}$, we select as its ground truth assignment the most salient $\widetilde{\mathbf{p}}_j \in \widetilde{\mathbf{P}}$ among the ones falling inside a certain radius r from $\mathbf{q}_i$. Saliency is decided according to the gradient magnitude of the SIFT features [2]. The proximity radius has been set to $r = 0.03 \times l$, where l is the diagonal-length of the image. The keypoints that are not involved in any ground truth assignment are discarded. So, at the end of this step we end up with keypoint-sets $\mathbf{Q}' = (\mathbf{q}'_1, \ldots, \mathbf{q}'_N)$ and $\mathbf{P}' = (\mathbf{p}'_1, \ldots, \mathbf{p}'_N)$, and a bijective mapping $f_{gtr} : \mathbf{P}' \rightarrow \mathbf{Q}'$ of ground truth assignments.

Once the N ground truth assignments have been established, we implement the clutter by adding a certain amount of the remaining points in both $\mathbf{P}$ and $\mathbf{Q}$ to $\mathbf{P}'$ and $\mathbf{Q}'$. Clutter points are carefully selected not to fall inside the radius of proximity r of any pre-existent point. Thus, we can safely assume that they have no correspondence in the other point-set.

Finally, geometrical noise consists on adding random gaussian noise with zero mean and a certain standard deviation σ_g to the point positions $\mathbf{p}_i = (p_x, p_y)$. This type of noise simulates nonrigid deformations in the position of the features.

Each plot is the average of the experiments on 10 images. Due to the random nature of the noise, we have run 10 experiments for each image.

Figure 1 shows the F-measure plots for an increasing amount of image distortions. Both geometrical noise and clutter have been set to zero.

Figure 2 shows the results for an increasing number of clutter points. The amount of point contamination has ranged from 0% to 80% of the total N points. Neither background geometrical noise nor image distortions have been introduced.

Figure 3 shows the results for geometrical noise with σ_g ranging from 0% to 50% of μ_d (where μ_d is the mean of the pairwise distances between the points). Neither image distortions nor clutter have been introduced.

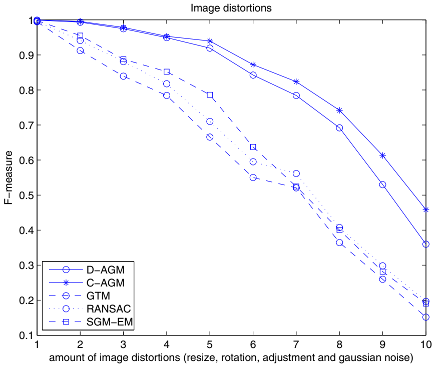


Fig. 1. Image distortions

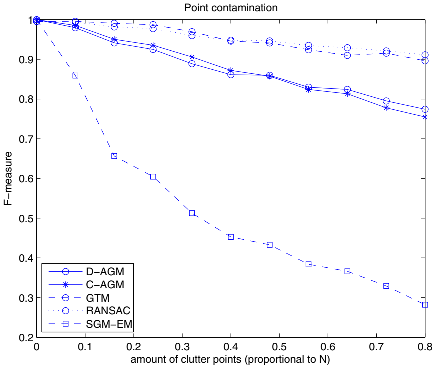


Fig. 2. Point contamination

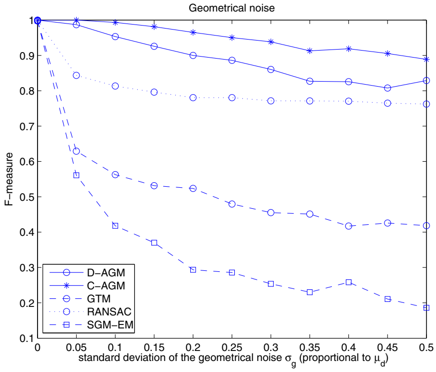


Fig. 3. Geometrical noise

6 Conclusions

We have presented a continuous and a discrete graph matching approach aimed at the matching of SIFT features in a structurally consistent way. They present two main novelties. On one hand, they force local and contextual consistency during the optimization process. On the other hand, they present a model of structural consistency based on the quality, rather than the quantity, of the surrounding matches. These features make them flexible and robust in front of various types of noise as seen in the experiments.

In the image distortion experiments, the methods that are not based on outlier rejection (C-AGM, D-AGM, SGM-EM) recover better than the others from matching misplacements. Specifically our attributed approaches (C-AGM, D-AGM) perform better than a purely structural one (SGM-EM). In the experiments with geometrical noise, the methods that only use structural information (GTM, SGM-EM) experience a considerable decreasing in performance. Our approaches (C-AGM, D-AGM) remain the most stable even under severe noise conditions. In the point contamination experiments, outlier rejectors (GTM, RANSAC) show the best performance. The continuous approach (C-AGM) performs better than the discrete one (D-AGM) in the image distortions and geometrical noise experiments. Results suggest us to work towards the achievement of a better stability in front of point contamination.

References

1. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(10), 1615–1630 (2005)
2. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2) (January 2004)
3. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Comunications of the ACM 24(6), 381–395 (1981)
4. Aguilar, W., Frauel, Y., Escolano, F., Martinez-Perez, M.E.: A robust graph transformation matching for non-rigid registration. Image and Vision Computing 27, 897–910 (2009)
5. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(4) (April 1996)
6. Waltz, D.: Understanding line drawings of scenes with shadows. In: The Psychology of Computer Vision, McGraw-Hill, New York (1975)
7. http://www.csse.uwa.edu.au/~pk/research/matlabfns/
8. Luo, B., Hancock, E.R.: Structural graph matching using the em algorithm and singular value decomposition. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(10) (October 2001)
9. Rosenfeld, A., Hummel, R.A., Zucker, S.W.: Scene labelling by relaxation operations. IEEE Transactions on Systems, Man and Cybernetics (6), 420–433 (1976)

A Causal Extraction Scheme in Top-Down Pyramids for Large Images Segmentation

Romain Goffe[1], Guillaume Damiand[2], and Luc Brun[3]

[1] SIC-XLIM, Université de Poitiers, CNRS, UMR6172, F-86962,
Futuroscope Chasseneuil, France
goffe@sic.univ-poitiers.fr

[2] LIRIS, Université de Lyon, CNRS, UMR5205, F-69622, Villeurbanne, France
guillaume.damiand@liris.cnrs.fr

[3] GREYC, ENSICAEN, CNRS, UMR6072, 6 Boulevard du Maréchal Juin,
F-14050, Caen, France
luc.brun@greyc.ensicaen.fr

Abstract. Applicative fields based on the analysis of large images must deal with two important problems. First, the size in memory of such images usually forbids a global image analysis hereby inducing numerous problems for the design of a global image partition. Second, due to the high resolution of such images, global features only appear at low resolutions and a single resolution analysis may loose important information. The tiled top-down pyramidal model has been designed to solve this two major challenges. This model provides a hierarchical encoding of the image at single or multiple resolutions using a top-down construction scheme. Moreover, the use of tiles bounds the amount of memory required by the model while allowing global image analysis. The main limitation of this model is the splitting step used to build one additional partition from the above level. Indeed, this step requires to temporary refine the split region up to the pixel level which entails high memory requirements and processing time. In this paper, we propose a new splitting step within the tiled top-down pyramidal framework which overcomes the previously mentioned limitations.

Keywords: Irregular pyramid; Topological model; Tiled data structure; Combinatorial map.

1 Introduction

High resolution image analysis usually entails memory issues that prevent them from being processed by common models. Moreover, in multi-resolution images, the amount of details at full resolution is likely to mask global features which only appear at lower resolutions. For instance, applicative fields such as whole slide microscopic imaging produce large multi-resolution images with resolutions up to 40 000 × 40 000: low resolutions let appear global features such as tissues delimitations while high resolutions allow to discern the different phases

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 264–274, 2010.

of mitosis within cells. As a result, analyzing such images implies a hierarchical representation with memory constraint.

The segmentation of an image defines an image partition into connected regions. Models for such partitions usually encode either geometrical or topological features of the partition. Operations involving both types of information are thus hard or costly to implement. For example, RAG-based data structures lack efficient access to regions' geometry. This drawback has entailed the design of topological maps [3,5] for an efficient representation of both geometrical and topological information while allowing modifications of a partition through split and merge operations. Yet, they cannot apply to multi-resolution images since they do not encode a hierarchy of partitions.

Quadtrees and regular pyramids' frameworks provide a multi-resolution description of the image and a hierarchical segmentation scheme [1] inducing a hierarchy of regions that may be defined onto such models. However, both models present several drawbacks: a given regular pyramid may fail to encode connected regions of any size and shape nor provide an efficient access to the neighborhood of a region. Moreover, both quadtrees and regular pyramids do not ensure that connected regions defined at a given level remain connected at the level below. The irregular pyramid framework has been introduced by [16,17] to overcome these limitations with different segmentation schemes such as [10,14]. Finally, in order to access both geometrical and topological information, [4,8] proposed a model of irregular pyramids composed of combinatorial maps. When applied to high resolution images, the bottom-up construction scheme of combinatorial pyramids raises at least two issues: memory usage and relevance of extracted information. Indeed, a bottom-up scheme starts from an explicit encoding of the whole initial partition: for large images this requires a large amount of memory especially if additional levels must also be computed. Moreover, in hierarchical data analysis, extracted information is usually more relevant if the construction scheme allows to use a region to influence the way its children (defined at a higher resolution) are processed. As a result, [7] have introduced the tiled top-down framework for combinatorial pyramids.

A tiled top-down pyramid is a hierarchical model based on topological maps [3,5] and thus, provides an efficient access to both geometrical and topological information. A top-down pyramidal model allows to reduce memory usage by encoding upper levels in the pyramid and refining only areas of interest. Moreover, the subdivision in tiles allows to bound the required amount of memory. Yet, its main drawback comes from its construction process [6]: in order to refine a region, a first step splits it into basic regions enclosing single pixels before the application of a merging step. This step may thus require a large amount of available memory to make temporary regions. Since the main operation in a top-down scheme consists in regions' splitting, we have explored alternative splitting techniques for combinatorial models. Different approaches have been proposed such as insertion operations [2] or incremental extractions [5,3] but those methods are not designed for a causal hierarchical model [9]. This causal property is fundamental within the tiled top-down pyramidal framework since it ensures the existence of a hierarchy.

In this paper, we propose a new method for the construction of tiled top-down pyramids which avoids the temporary split of a region into basic regions enclosing a single pixel while preserving the causality of the pyramid. In section 2, we present the different topological models used to define a tiled top-down pyramid. In section 3, we detail our causal extraction for such pyramids. Finally, experiments and segmentation results are proposed in section 4 in order to emphasize the advantage of our method.

2 Recalls

2.1 Combinatorial Maps

In two dimensions, a combinatorial map (noted 2-map) is a set of vertices, edges and faces that encodes the subdivision and incidence relationships of a topological space [15]. A complete decomposition of an image results in a set of abstract basic elements called *darts*. We introduce two operators noted β_i, $i \in \{1, 2\}$ that apply on darts in order to represent adjacency relationships (Fig. 1).

Definition 1 (2-dimensional combinatorial map). *A two-dimensional combinatorial map M (or 2-map) is a triplet $M = (D, \beta_1, \beta_2)$ where:*

(1) D is a finite set of darts;
(2) β_1 is a permutation[1] *on D;*
(3) β_2 is an involution[2] *on D.*

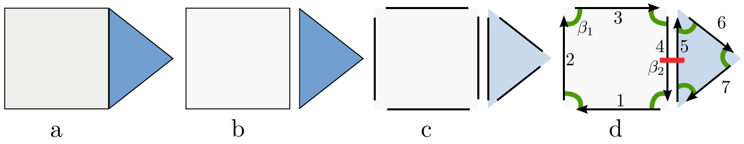


Fig. 1. Combinatorial maps: construction by successive decompositions. (a) Original image. (b) Decomposed faces. (c) Decomposed edges. (d) 2-Map: arrows represent darts, β_1 and β_2 operators are respectively represented by arcs and segments.

Intuitively, we can consider a map as a planar graph where β_i operators explicitly define the relationships between edges and where darts allow to differentiate the two extremities of an edge (a dart may be assimilated to a half-edge). In practice, the β_1 permutation allows to turn around a face: it links a dart of a face to the next one encountered while turning clockwise around it. The β_2 involution separates two adjacent faces: it links a dart to the other dart that belongs to the

[1] A *permutation* is a one to one mapping from S onto S.
[2] An *involution* f is a one to one mapping from S onto S such that $f = f^{-1}$.

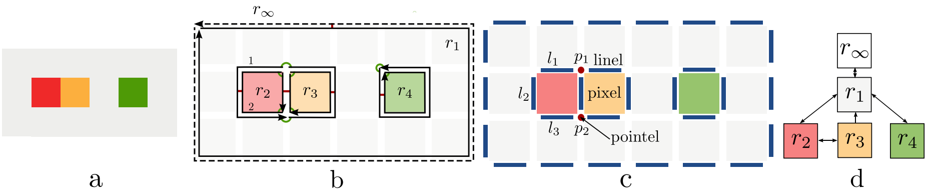


Fig. 2. Topological map: three complementary models for image representation. (a) Original image. (b) Combinatorial map for topological relationships. Dotted arrow denotes the dart of the infinite region. (c) Interpixel matrix for geometrical encoding: pointels and linels are represented by bold circles and segments. (d) Tree of regions.

same edge but has an opposite orientation. For instance, in figure1.d, $\beta_1(3) = 4$ and $\beta_2(4) = 5$. As a result, a 2-map is a connected set of cells of 0, 1, and 2 dimensions. For $i \in \{0, 1, 2\}$, an i-cell respectively denotes a vertex, an edge and a face. The degree of an i-cell is its number of distinct incident $(i + 1)$-cells.

2.2 Topological Maps

Since combinatorial maps only describe topological relationships, an extension of the model which also encodes geometrical information has been introduced for a full representation of a partition: the model of topological map [3,5]. A topological map combines three distinct models: a 2-map that encodes topological relationships, a matrix of *interpixel elements* [13,12] that encodes the geometry of the partition elements, and a tree of regions for inclusion relationships. These three models are illustrated in figure2 and described below.

Minimal combinatorial map. As illustrated in figure2.b, a 2-map encodes topological relationships through β_1 and β_2 operators (section2.1). The combinatorial map is *minimal* in number of cells: there is not any vertex with a degree equal to 2 and therefore, the removal of any element would change the topology. For implementation purposes, darts and regions are linked together: a dart knows the region it belongs to and a region knows a representative dart (arbitrary chosen on the external border of the region). Note that the infinite region may be omitted in some figures for visibility reasons.

Matrix of interpixel elements. Pointels, linels and pixels [12] represent the geometry of a partition. Associating geometrical information to a topological element is an operation called *embedding*. Similarly to vertices, edges and faces, we respectively refer to pointels, linels and pixels as i-cells, $i \in \{0, 1, 2\}$. We respectively denote by $pointel(d)$ and $linel(d)$ the first pointel and linel of the embedding of a dart d. For example, in figure2.c, the embedding of the edge (1,2) is the sequence of linels (l_1, l_2, l_3); $linel(1) = l_1$, $pointel(2) = p_2$; $degree(p_1) = degree(p_2) = 3$.

Tree of regions. The tree of regions describes inclusion relationships: a region is the father of the regions it contains. In figure2.d, r_1 contains r_2, r_3 and r_4, r_2

and r_3 are adjacent. The root of the tree encodes the background of the image and is called the infinite region (noted r_∞).

2.3 Tiled Top-Down Framework for Combinatorial Pyramids

A hierarchical extension of the topological map model is proposed by [6]. Contrary to bottom-up methods, this framework uses a top-down approach which induces a segmentation process based on a rough partition refined at further levels: it results in a major memory reduction since regions may only be encoded at the top level of the pyramid. Moreover, it allows to take advantage of the *focus of attention* over interesting regions: the segmentation of a region can be adapted according to the features of its parent. Despite this memory reduction, [7] proposes a subdivision of the levels into topological tiles in order to bound the amount of required memory.

A topological tile is a topological map with an additional involution β_2' which applies on darts belonging to a border shared by two adjacent tiles [7] to ensure their topological connection. The juxtaposition of topological tiles composes a tiled topological map. Such a map may contain fictive elements along the tiles' borders when, according to a given merging criterion, pixels on both sides of a tile's border belong to a same region. Linels encoding these fictive borders are marked by a flag indicating their fictive state. Tiled combinatorial maps (definition 2) redefine the operators β_1 and β_2 [7] to abstract those fictive elements (figure 3.c).

Definition 2 (Tiled combinatorial map). *Let T be a set of connected topological tiles $T = \{t(i,j)\}_{(i,j)\in\{0,\dots,W\}\times\{0,\dots,H\}}$. Let D be the set of darts of T with a real embedding. A tiled combinatorial map M is a triplet $M = (D, \delta_1, \delta_2)$ where, $\forall d \in D$:*

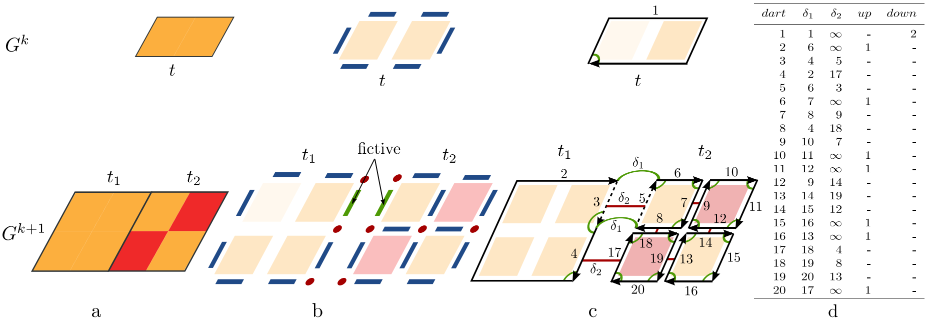
(1) δ_1 is a permutation on D such as:

$$\delta_1(d) = \beta_1((\beta_2'\beta_1)^n(d)) \text{ with } n = min\{p \in \mathbb{N} \mid linel(\beta_1((\beta_2'\beta_1)^p(d)) \text{ is real}\}$$

(2) δ_2 is an involution on D such as:

$$\delta_2(d) = \begin{cases} \beta_2'(d) \text{ if } \beta_2'(d) \text{ exists} \\ \beta_2(d) \text{ otherwise} \end{cases}$$

A tiled top-down topological pyramid is a stack of finer and finer partitions denoted by $P = \{G^k\}_{k\in\{0,\dots,n\}}$ where G^{k+1} is a tiled combinatorial map (definition 2) deduced from G^k by splitting operations. Within a pyramid, a tiles is denoted by $t(i,j,k)$ where (i,j,k) are the coordinates (i,j) of t at level G^k. Besides, the pyramid may swap or load tiles between memory and disk and spread global modifications: if an operation modifies a tile's border, adjacent tiles that are either on disk or in memory should be updated. Note that a stack of successively refined partitions differs from the notion of resolution used within the regular pyramid framework: top-down pyramids can be constructed either from single or multi-resolution images. Although, the pyramid mixes both regular



dart	δ_1	δ_2	*up*	*down*
1	1	∞	-	2
2	6	∞	1	-
3	4	5	-	-
4	2	17	-	-
5	6	3	-	-
6	7	∞	1	-
7	8	9	-	-
8	4	18	-	-
9	10	7	-	-
10	11	∞	1	-
11	12	∞	1	-
12	9	14	-	-
13	14	19	-	-
14	15	12	-	-
15	16	∞	1	-
16	13	∞	1	-
17	18	4	-	-
18	19	8	-	-
19	20	13	-	-
20	17	∞	1	-

Fig. 3. Representation of a top-down pyramid composed of two levels G^k and G^{k+1}. (a) Original image: a tile t is decomposed in two tiles t_1 and t_2 ($ratio = 2 \times 2$). (b) Interpixel matrix: fictive borders appear between t_1 and t_2. (c) Tiled combinatorial maps: δ_1 and δ_2 are represented by arcs and segments. Dotted arrows denote darts with a fictive embedding. (d) Relationships between darts.

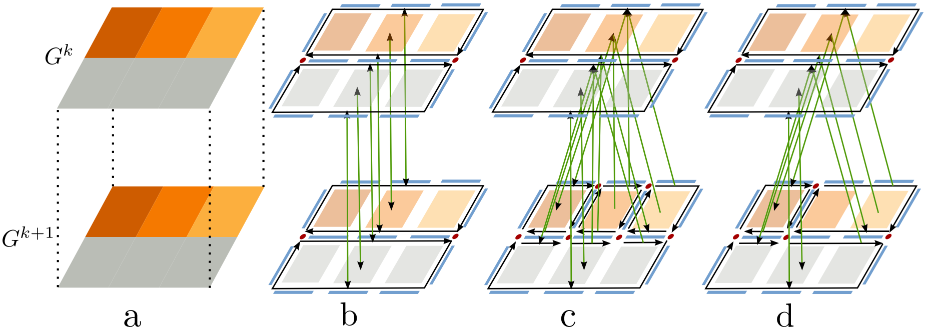


Fig. 4. Refinement of the regions that compose a level of a top-down pyramid. (a) Original image. (b) Level duplication and *up/down* relationships between darts and regions. (c) Burst of selected regions into basic regions enclosing a single pixel. (d) Regions merging according to segmentation criterion.

and irregular notions, the top-down model remains an irregular pyramid since it handles fictive borders between the tiles. The resulting sequence of partitions is a *causal structure* [9] where hierarchical relationships are encoded through *up/down* relationships between tiles, darts and regions.

A first strategy to build the pyramid is to start from a single region and refines it according to segmentation criteria. The operation is performed in three steps. First, the last level is duplicated and *up/down* relationships are set (figure 4.b). Second, a *splitting criterion* indicates the regions to refine. Those regions are split into a set of basic regions enclosing a single pixel (figure 4.c). Third, the basic regions are merged according to a *merging criterion* (figure 4.d). Since any couple of adjacent regions may be merged, this refinement step may encode any subdivision of the parent region. Yet, this solution presents a major drawback:

the splitting step in one region per pixel is a bottom-up refinement that encodes every pixel of the split region. Such an operation implies useless calculus and important memory requirements.

3 A Causal Extraction Scheme for Tiled Top-Down Pyramids

Our main objective consists in avoiding the bottom-up refinement step temporary creating one region per pixel. We propose with algorithm 1 a hierarchical extension of the extraction scheme introduced by [5]. In order to adapt it to the tiled top-down pyramidal framework, we must fulfill two main constraints:

- the *causal constraint* entails that existing borders are preserved from one level to an other: any border defined in G^k must exist in G^{k+1} and darts and regions of G^k must be connected to their children in G^{k+1}.
- the *top-down constraint* (focus of attention): since we use a splitting criterion which determines whether a region should be refined in the next level, no border should be created within regions whose splitting criterion is *false*.

Those two constraints are illustrated in figure 5. In figure 5.a, the causality is ensured as each border defined in G^k exists in G^{k+1} with *up/down* relationships set accordingly. For example, when the darts 3 and 4 are created, they must be linked with their respective parents 1 and 2. In figure 5.b, the focus of attention for the top-down construction is respected as regions whose splitting criterion is set to *false* are not refined: no border is inserted in $down(r_1)$ while r_2 is refined in G^{k+1}.

The global scheme of the extraction algorithm is the following. All the pixels of a tile t in G^{k+1} are traversed with a scanline traversal from top-left to bottom-right corner. For each pixel p, we create the region enclosing it (line 1 of algorithm 1) which results in the insertion of two new borders between the p and its *top* and *left* neighbors. Then, we determine whether those borders should

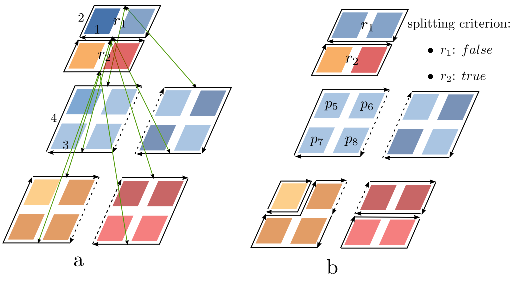


Fig. 5. Extraction constraints for a tiled top-down pyramid. G^k is a single tile t decomposed in G^{k+1} into 4 subtiles ($ratio = 2 * 2$). (a) Causal constraint: preserves existing borders. (b) Top-down constraint: the focus of attention only refines regions whose *splitting criterion* is *true*. Dotted arrows denote fictive borders.

Algorithm 1. Causal extraction algorithm.

```
  Data: A tile $t$ in $G^k$.
  Result: Extraction of the sons of $t$ in $G^{k+1}$.
  $S_1 \leftarrow$ embedding structure of $t$;
  $S_2 \leftarrow$ regions structure of $t$;
  foreach son $t'$ of $t$ in $G^{k+1}$ do
      foreach pixel $p(x, y)$ in $t'$ do
          $p \leftarrow p(x, y)$, $p' \leftarrow p(x-1, y)$, $p'' \leftarrow p(x, y-1)$;
1         Create region $r$ enclosing $p$;
2         if $SameRegion(p, p', S_1, S_2)$ then
              Remove border between $region(p)$ and $region(p')$;
3         if $SameRegion(p, p'', S_1, S_2)$ then
              Remove border between $region(p)$ and $region(p'')$;
4         Set up/down relationships for $r$ and its darts;
5     Simplify and compute the tree of regions for $t'$;
```

be kept or removed (line 2-3) by calling algorithm 2. First, our two constraints must be respected: a border is kept if it corresponds to an existing border in G^k (line 1 of algorithm 2) and is removed if it corresponds to a forbidden refinement (line 2). Once those two conditions are verified, a merging criterion determines if the border is kept (line 3). Since a topological map aims at representing regions, the merging criterion has to define a partition: we may use a quantization of the tile *up* but conversely, a criterion based on a minimum gradient would lead to the creation of an inconsistent map with dangling edges. The last step finalizes the extraction by removing degree two vertices and computing the tree of regions (line 5 of algorithm 1) as described in [5]. In the following, we implicitly use projections between levels: if p is a pixel of G^{k+1}, $region(p)$ in G^k refers to the region of the projection of pixel p in G^k. In order to answer the first two conditions of algorithm 2, two external structures related to $up(t)$ called *embedding structure* and *regions structure* are required and detailed below.

Embedding structure. This structure establishes the correspondence between darts and their embedding in a tile by mapping each linel to its dart. Thus, we can determine from the embedding of a dart d in G^{k+1} if there exists a dart d' in G^k whose embedding is down-projected onto the one of d. In this case, the border is kept and we establish *up/down* relationships between $d = down(d')$ and $d' = up(d)$ (line 4 of algorithm 1).

Regions structure. Similarly to an image of labels, this structure maps each tile's pixel to its region. It allows to know the splitting criterion of a pixel p in the above level. For instance in figure 5.b, we can determine that $region(p_7)$ in G^k is r_1 whose splitting criterion is *false* (r_1 must not be refined) so we do not keep the border between p_7 and p_5.

Algorithm 2. SameRegion.

```
Data: Two adjacent pixels $p$ and $p'$ of a tile $t$ at $G^{k+1}$.
      embedding structure and regions structure of $up(t)$ at $G^k$.
Result: true if $p$ and $p'$ belong to the same region.
$r \leftarrow region(p)$ in $G^k$, $r' \leftarrow region(p')$ in $G^k$;
if $r \neq r'$ then
    return false;
if $splitting\_criterion(r) = false$ then
    return true;
return $merging\_criterion(p, p')$;
```

4 Experiments

This section has two main objectives: show the advantage of our causal extraction algorithm compared to the initial burst-merge refinement and present first segmentation results obtained on histological images.

Table 1. Memory usage and runtime comparisons between the burst-merge refinement construction and the causal extraction process for a multi-resolution histological image

level size	number	number	burst-merge		causal extract.	
(pixels)	of darts	of regions	runtime (s)	ram (MB)	runtime (s)	ram (MB)
4 000×3 036	483 662	187 033	181	104	105	90
8 000×6 072	962 368	392 395	354	106	272	90
32 000×24 291	4 670 978	1 767 890	4 611	100	3 869	92

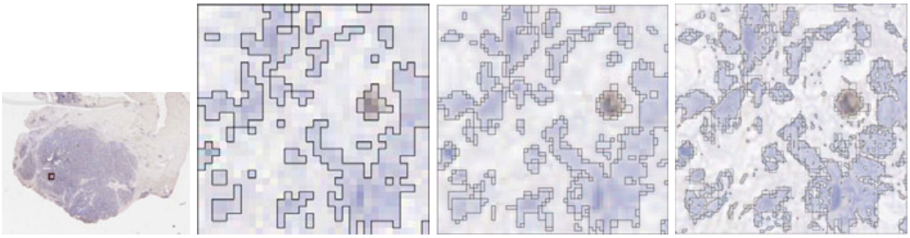


Fig. 6. Causal extraction of a tiled top-down pyramid from a multi-resolution histological image. From left to right: low resolution of the original image, three different levels of the pyramid with increasing resolutions encoding a same small area.

In table 1, we provide runtime[3] and memory usage for our causal extraction algorithm (column 6-7) compared to a burst-merge refinement (column 4-5) with

[3] The model is implemented in C++ and computations are carried out on an Intel E5300@2GHz with 2GB RAM.

a fixed tile size of 512×512. We can notice that our extraction method globally improves the computational time from 15 to 40% with a reduction of memory usage around 10%. The computational time is mostly due to pixel traversal (to get colorimetric information for the regions) and segmentation criteria computation.

We illustrate the results of our extraction in figure 6. This example shows the tiled map of the same small area of an histological image at different resolutions $(4\,000 \times 3\,036, 8\,000 \times 6\,072, 32\,000 \times 24\,291)$. The segmentation is based on an image quantization algorithm in two classes [11].

5 Conclusion

In this paper, we have presented an alternative construction scheme for the extraction of tiled top-down pyramids that overcomes the main drawbacks of the previous method. First, our method requires less computational time and memory usage. Second, it still preserves the top-down hierarchical relationships of the model. Therefore, our method can favorably act as a replacement for the burst and merge refinement step during the construction of a tiled top-down pyramid. Finally, our causal extraction algorithm presents interesting perspectives for parallel computation that we plan to implement in our future work.

Acknowledgements

This research is part of the FoGrImMi project, supported by the ANR foundation under grant ANR-06-MDCA-008-01/FOGRIMMI.

References

1. Bister, M., Cornelis, J., Rosenfeld, A.: A critical view of pyramid segmentation algorithms. Pattern Recognition Letters 11(9), 605–617 (1990)
2. Braquelaire, J.P., Domenger, J.P.: Representation of segmented images with discrete geometric maps. Image and Vision Computing 17(10), 715–735 (1999)
3. Brun, L., Domenger, J.-P., Mokhtari, M.: Incremental modifications of segmented image defined by discrete maps. J. Visual Communication and Image Representation 14(3), 251–290 (2003)
4. Brun, L., Kropatsch, W.G.: Combinatorial pyramids. In: ICIP (2), pp. 33–36 (2003)
5. Damiand, G., Bertrand, Y., Fiorio, C.: Topological model for two-dimensional image representation: definition and optimal extraction algorithm. Computer Vision and Image Understanding 93(2), 111–154 (2004)
6. Goffe, R., Brun, L., Damiand, G.: A top down construction scheme for irregular pyramids. In: VISSAPP (1), pp. 163–170 (2009)
7. Goffe, R., Damiand, G., Brun, L.: Extraction of tiled top-down irregular pyramids from large images. In: 13th International Workshop on Combinatorial Image Analysis (IWCIA 2009), pp. 123–137. RPS, Singapore (2009)
8. Grasset-Simon, C., Damiand, G., Lienhardt, P.: n-d generalized map pyramids: Definition, representations and basic operations. Pattern Recognition 39(4), 527–538 (2006)

9. Guigues, L., Cocquerez, J.P., Le Men, H.: Scale-sets image analysis. International Journal of Computer Vision 68(3), 289–317 (2006)
10. Jolion, J.-M., Montanvert, A.: The adaptative pyramid: A framework for 2d image analysis. CVGIP 55(3), 339–348 (1992)
11. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An efficient k-means clustering algorithm: Analysis and implementation. IEEE Trans. on PAMI 24(7), 881–892 (2002)
12. Khalimsky, E., Kopperman, R., Meyer, P.R.: Boundaries in digital planes. Journal of Applied Mathematics and Stochastic Analysis 3(1), 27–55 (1990)
13. Kovalevsky, V.A.: Finite topology as applied to image analysis. Computer Vision, Graphics, and Image Processing 46(2), 141–161 (1989)
14. Kropatsch, W.: Building irregular pyramids by dual graph contraction. IEE Proceedings - Vision, Image, and Signal Processing 142, 366–374 (1995)
15. Lienhardt, P.: Subdivisions of *n*-dimensional spaces and *n*-dimensional generalized maps. In: Symposium on Computational Geometry, pp. 228–236 (1989)
16. Meer, P.: Stochastic image pyramids. Computer Vision, Graphics, and Image Processing 45(3), 269–294 (1989)
17. Montanvert, A., Meer, P., Rosenfeld, A.: Hierarchical image analysis using irregular tessellations. IEEE Trans. Pattern Anal. Mach. Intell. 13(4), 307–316 (1991)

Fast Population Game Dynamics for Dominant Sets and Other Quadratic Optimization Problems

Samuel Rota Bulò[1], Immanuel M. Bomze[2], and Marcello Pelillo[1]

[1] Dipartimento di Informatica - Univ. of Venice - Italy
{srotabul,pelillo}@dsi.unive.it
[2] Department of Statistics and Decision Support Systems - Univ. of Vienna - Austria
immanuel.bomze@univie.ac.at

Abstract. We propose a fast population game dynamics, motivated by the analogy with infection and immunization processes within a population of "players," for finding dominant sets, a powerful graph-theoretical notion of a cluster. Each step of the proposed dynamics is shown to have a linear time/space complexity and we show that, under the assumption of symmetric affinities, the average population payoff is strictly increasing along any non-constant trajectory, thereby allowing us to prove that dominant sets are asymptotically stable (i.e., attractive) points for the proposed dynamics. The approach is general and can be applied to a large class of quadratic optimization problems arising in computer vision. Experimentally, the proposed dynamics is found to be orders of magnitude faster than and as accurate as standard algorithms.

1 Introduction

Dominant sets are a graph-theoretical notion of a cluster [1], which have found application in problems as diverse as the analysis of fMRI data [2], content-based image retrieval [3], detection of anomalous activities in video streams [4], bioinformatics [5], human action recognition [6] and matching problems [7,8].

Computationally, the standard approach to finding dominant sets in an edge-weighted graph is to use *replicator dynamics*, a class of evolutionary game-theoretic algorithms inspired by Darwinian selection processes. However, a typical problem associated with these algorithms is the scaling behavior with the number of data. On a dataset containing N examples, the computationally complexity of each replicator dynamics step is $\mathcal{O}(N^2)$, thereby hindering their applicability to problems involving very large data sets, such as high-resolution imagery and spatio-temporal data.

In order to avoid this drawback, in this paper we propose a new population game dynamics for finding dominant sets which turns out to be dramatically faster and even more accurate than standard approaches from evolutionary game theory. Our approach is motivated by the analogy with infection and immunization processes within a population of "players." The selection mechanism

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 275–285, 2010.

governing our dynamics iteratively performs an infection step, which consists of spreading (or suppressing) the most successful (unsuccessful) strategies in the population. The infection phase is then protracted as long as the selected "infective" strategy performs better (or worse, if not extinct) than the average population's payoff. As opposed to standard techniques, such as the replicator dynamics or best-response dynamics, which can be considered interior-point methods, our algorithm resembles a vertex-pivoting method. Each step of the proposed dynamics is shown to have a linear time/space complexity and we show that, under the assumption of symmetric affinities, the average population payoff is strictly increasing along any non-constant trajectory, thereby allowing us to prove that dominant sets (i.e., ESS equilibria of the underlying "grouping game" [9]) are asymptotically stable points for the proposed dynamics.

We provide experimental evidence that the proposed algorithm is orders of magnitude faster than standard dynamics on two computer vision applications, namely image segmentation and region-based hierarchical image matching, while preserving the quality of the solutions found.

Although the main focus in this paper is dominant sets, we note that the proposed approach is general and can be applied to a large class of optimization problem, instances of which abound in computer vision and pattern recognition (e.g., graph matching, stereo matching, image labeling, etc.).

2 Basics of Evolutionary Game Theory

Evolutionary game theory considers an idealized scenario whereby pairs of individuals are repeatedly drawn at random from a large, ideally infinite, population to play a symmetric two-player game. Let $O = \{1, \dots, n\}$ be the set of *pure strategies* available to the players and let A be the $n \times n$ payoff or utility matrix [10], where a_{ij} is the payoff that a player gains when playing the strategy i against an opponent playing strategy j. A *mixed strategy* is a probability distribution $\mathbf{x} = (x_1, x_2, \dots, x_n)^\top$ over the available strategies in O. Mixed strategies lie in the standard simplex Δ of the n-dimensional Euclidean space, which is defined as

$$\Delta = \left\{ \mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^{n} x_i = 1 \text{ and } x_i \geq 0,\ i = 1, \dots, n \right\} .$$

We denote by $\mathbf{e}^i$ the ith column of the identity matrix. The *support* of a mixed strategy $\mathbf{x} \in \Delta$, denoted by $\sigma(\mathbf{x})$, defines the set of elements with non-zero probability: $\sigma(\mathbf{x}) = \{i \in O : x_i > 0\}$. The expected payoff that a player obtains by playing the pure strategy i against an opponent playing a mixed strategy $\mathbf{x}$ is $\pi(\mathbf{e}^i|\mathbf{x}) = (A\mathbf{x})_i = \sum_j a_{ij}x_j$ .Hence, the expected payoff received by adopting a mixed strategy $\mathbf{y}$ is given by $\pi(\mathbf{y}|\mathbf{x}) = \mathbf{y}^\top A\mathbf{x}$ while the population expected payoff is $\pi(\mathbf{x}) = \pi(\mathbf{x}|\mathbf{x}) = \mathbf{x}^\top A\mathbf{x}$. For notational compactness, in the sequel we will write $\pi(\mathbf{y} - \mathbf{x}|\mathbf{z})$ for the payoff difference $\pi(\mathbf{y}|\mathbf{z}) - \pi(\mathbf{x}|\mathbf{z})$, and $\pi(\mathbf{y} - \mathbf{x})$ for $\pi(\mathbf{y} - \mathbf{x}|\mathbf{y}) - \pi(\mathbf{y} - \mathbf{x}|\mathbf{x})$.

A mixed strategy $\mathbf{x}$ is a *(symmetric) Nash (equilibrium) strategy* if for all $\mathbf{y} \in \Delta$, we have $\pi(\mathbf{y}-\mathbf{x}|\mathbf{x}) \leq 0$. This implies that $\pi(\mathbf{e}^i-\mathbf{x}|\mathbf{x}) \leq 0$ for all $i \in O$, which in turn implies that $\pi(\mathbf{e}^i-\mathbf{x}|\mathbf{x}) = 0$ for all $i \in \sigma(\mathbf{x})$. Hence, the payoff is constant across all (pure) strategies in the support of $\mathbf{x}$, while all strategies outside the support of $\mathbf{x}$ earn a payoff that is less than or equal $\pi(\mathbf{x})$.

A strategy $\mathbf{x}$ is said to be an *Evolutionary Stable Strategy* (ESS) if it is a Nash strategy (*equilibrium condition*) and for all $\mathbf{y} \in \Delta\setminus\{\mathbf{x}\}$ satisfying $\pi(\mathbf{y}-\mathbf{x}|\mathbf{x}) = 0$ we have $\pi(\mathbf{y}-\mathbf{x}|\mathbf{y}) < 0$ (*stability condition*). Intuitively, ESS's are strategies such that any small deviation from them will lead to an inferior payoff. ESS's can be found by *replicator dynamics* (RD), a classic formalization of a natural selection process [10].

3 Dominant Sets and Their Characterizations

The dominant set framework is a pairwise clustering approach [1] that is based on the notion of a dominant set, which can be seen as an edge-weighted generalization of a clique. The framework is based on a recursive characterization of the weight $W_S(i)$ of element i with respect to a set S of elements, and characterizes a group as a *dominant set*, i.e., a set that satisfies:

1. $W_S(i) > 0$, for all $i \in S$,
2. $W_{S\cup\{i\}}(i) < 0$, for all $i \notin S$.

These conditions correspond to the two main properties of a cluster: the first regards internal homogeneity, whereas the second regards external heterogeneity.

The characteristic vector $\mathbf{x}^S$ of a set $S \subseteq V$ is defined as

$$x_i^S = \begin{cases} \frac{W_S(i)}{W(S)} & \text{if } i \in S\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

The following result establishes a one-to-one correspondence between ESS's and dominant sets [9].

Theorem 1. *If $S \subseteq V$ is a dominant set with respect to affinity matrix A, then $\mathbf{x}^S$ is an ESS for a two-player game with payoff matrix A.*

Conversely, if $\mathbf{x}$ is an ESS for a two-person game with payoff matrix A, then $S = \sigma(\mathbf{x})$ is a dominant set with respect to A, provided that $W_{S\cup\{i\}}(i) \neq 0$ for all $i \notin S$.

Under the assumption of a symmetric affinity matrix A there exists a one-to-one correspondence between dominant sets and the (strict) local solutions of the following so-called standard quadratic program (StQP) [1]:

$$\max\left\{\mathbf{x}^\top A\mathbf{x} \,:\, \mathbf{x} \in \Delta\right\}\,. \tag{1}$$

4 A New Class of Evolutionary Dynamics

Let $\mathbf{x} \in \Delta$ be the *incumbent* population state, $\mathbf{y}$ be the *mutant* population invading $\mathbf{x}$ and let $\mathbf{z} = (1-\varepsilon)\mathbf{x} + \varepsilon\mathbf{y}$ be the population state obtained by injecting into $\mathbf{x}$ a small share of $\mathbf{y}$-strategists. The *score function* of $\mathbf{y}$ versus $\mathbf{x}$ [11] is given by:

$$h_{\mathbf{x}}(\mathbf{y},\varepsilon) = \pi(\mathbf{y}-\mathbf{x}|\mathbf{z}) = \varepsilon\pi(\mathbf{y}-\mathbf{x}) + \pi(\mathbf{y}-\mathbf{x}|\mathbf{x})\,.$$

Following [12], we define the *(neutral) invasion barrier* $b_{\mathbf{x}}(\mathbf{y})$ of $\mathbf{x} \in \Delta$ against any mutant strategy $\mathbf{y}$ as the largest population share $\varepsilon_{\mathbf{y}}$ of $\mathbf{y}$-strategists such that for all smaller positive population shares ε, $\mathbf{x}$ earns a higher or equal payoff than $\mathbf{y}$ in the post-entry population $\mathbf{z}$. Formally:

$$b_{\mathbf{x}}(\mathbf{y}) = \inf(\{\varepsilon \in (0,1) : h_{\mathbf{x}}(\mathbf{y},\varepsilon) > 0\} \cup \{1\})\,.$$

Given populations $\mathbf{x}, \mathbf{y} \in \Delta$, we say that $\mathbf{x}$ is *immune* against $\mathbf{y}$ if $b_{\mathbf{x}}(\mathbf{y}) > 0$. Trivially, a population is always immune against itself. Note that, $\mathbf{x}$ is immune against $\mathbf{y}$ if and only if either $\pi(\mathbf{y}-\mathbf{x}|\mathbf{x}) < 0$ or $\pi(\mathbf{y}-\mathbf{x}|\mathbf{x}) = 0$ and $\pi(\mathbf{y}-\mathbf{x}) \leq 0$. If $\pi(\mathbf{y}-\mathbf{x}|\mathbf{x}) > 0$ we say that $\mathbf{y}$ is *infective* for $\mathbf{x}$. We denote the set of infective strategies for $\mathbf{x}$ as

$$\Upsilon(\mathbf{x}) = \{\mathbf{y} \in \Delta :\ \pi(\mathbf{y}-\mathbf{x}|\mathbf{x}) > 0\}\,.$$

Consider $\mathbf{y} \in \Upsilon(\mathbf{x})$; clearly, this implies $b_{\mathbf{x}}(\mathbf{y}) = 0$. If we allow for invasion of a share ε of $\mathbf{y}$-strategists as long as the score function of $\mathbf{y}$ versus $\mathbf{x}$ is positive, at the end we will have a share of $\delta_{\mathbf{y}}(\mathbf{x})$ mutants in the post-entry population, where

$$\delta_{\mathbf{y}}(\mathbf{x}) = \inf\left(\{\varepsilon \in (0,1) :\ h_{\mathbf{x}}(\mathbf{y},\varepsilon) \leq 0\} \cup \{1\}\right)\,.$$

Note that if $\mathbf{y}$ is infective for $\mathbf{x}$, then $\delta_{\mathbf{y}}(\mathbf{x}) > 0$, whereas if $\mathbf{x}$ is immune against $\mathbf{y}$, then $\delta_{\mathbf{y}}(\mathbf{x}) = 0$. Since score functions are (affine-)linear, there is a simpler expression $\delta_{\mathbf{y}}(\mathbf{x}) = \min\left[\frac{\pi(\mathbf{x}-\mathbf{y}|\mathbf{x})}{\pi(\mathbf{y}-\mathbf{x})}, 1\right]$, if $\pi(\mathbf{y}-\mathbf{x}) < 0$, and $\delta_{\mathbf{y}}(\mathbf{x}) = 1$, otherwise.

Proposition 1. *Let $\mathbf{y} \in \Upsilon(\mathbf{x})$ and $\mathbf{z} = (1-\delta)\mathbf{x} + \delta\mathbf{y}$, where $\delta = \delta_{\mathbf{y}}(\mathbf{x})$. Then $\mathbf{y} \notin \Upsilon(\mathbf{z})$.*

The proof of this result is straightforward by linearity and can be found, e.g., in [13].

The core idea of our method is based on the fact that $\mathbf{x} \in \Delta$ is a Nash equilibrium if and only if $\Upsilon(\mathbf{x}) = \emptyset$ (we prove this in Theorem 2). Therefore, as long as we find a strategy $\mathbf{y} \in \Upsilon(\mathbf{x})$, we update the population state according to Proposition 1 in order obtain a new population $\mathbf{z}$ such that $\mathbf{y} \notin \Upsilon(\mathbf{z})$ and we reiterate this process until no infective strategy can be found, or in other words, a Nash equilibrium is reached.

The formalization of this process provides us with a class of new dynamics which, for evident reasons, is called *Infection and Immunization Dynamics* (InImDyn):

$$\mathbf{x}^{(t+1)} = \delta_{\mathcal{S}(\mathbf{x}^{(t)})}(\mathbf{x}^{(t)})[\mathcal{S}(\mathbf{x}^{(t)}) - \mathbf{x}^{(t)}] + \mathbf{x}^{(t)}\,. \qquad (2)$$

Here, $\mathcal{S} : \Delta \to \Delta$ is a generic *strategy selection* function which returns an infective strategy for $\mathbf{x}$ if it exists, or $\mathbf{x}$ otherwise:

$$\mathcal{S}(\mathbf{x}) = \begin{cases} \mathbf{y} & \text{for some } \mathbf{y} \in \Upsilon(\mathbf{x}) \;\; \text{if } \Upsilon(\mathbf{x}) \neq \emptyset\,, \\ \mathbf{x} & \text{otherwise}\,. \end{cases} \tag{3}$$

By running these dynamics we aim at reaching a population state that can not be infected by any other strategy. In fact, if this is the case, then $\mathbf{x}$ is a Nash strategy, which happens if and only if it is fixed (i.e., stationary) under dynamics (2):

Theorem 2. *Let $\mathbf{x} \in \Delta$ be a strategy. Then the following statements are equivalent:*
(a) $\Upsilon(\mathbf{x}) = \emptyset$: there is no infective strategy for $\mathbf{x}$;
(b) $\mathbf{x}$ is a Nash strategy;
(c) $\mathbf{x}$ is a fixed point under dynamics (2).

Proof. A strategy $\mathbf{x}$ is a Nash strategy if and only if $\pi(\mathbf{y} - \mathbf{x}|\mathbf{x}) \leq 0$ for all $\mathbf{y} \in \Delta$. This is true if and only if $\Upsilon(\mathbf{x}) = \emptyset$. Further, $\delta = 0$ implies $\mathcal{S}(\mathbf{x}) = \mathbf{x}$. Conversely, if $\mathcal{S}(\mathbf{x})$ returns $\mathbf{x}$, then we are in a fixed point. By construction of $\mathcal{S}(\mathbf{x})$ this happens only if there is no infective strategy for $\mathbf{x}$.

The following result shows that average payoff is strictly increasing along any non-constant trajectory of the dynamics (2), provided that the payoff matrix is symmetric.

Theorem 3. *Let $\{\mathbf{x}^{(t)}\}_{t\geq 0}$ be a trajectory of (2). Then for all $t \geq 0$,*

$$\pi(\mathbf{x}^{(t+1)}) \geq \pi(\mathbf{x}^{(t)})\,,$$

with equality if and only if $\mathbf{x}^{(t)} = \mathbf{x}^{(t+1)}$, provided that the payoff matrix is symmetric.

Proof. Again, let us write $\mathbf{x}$ for $\mathbf{x}^{(t)}$ and δ for $\delta_{\mathcal{S}(\mathbf{x})}(\mathbf{x})$. As shown in [13], we have

$$\pi(\mathbf{x}^{(t+1)}) - \pi(\mathbf{x}^{(t)}) = \delta\left[h_{\mathbf{y}}(\mathbf{x}, \delta) + \pi(\mathbf{y} - \mathbf{x}|\mathbf{x})\right]\,.$$

If $\mathbf{x}^{(t+1)} \neq \mathbf{x}^{(t)}$, then $\mathbf{x}$ is no Nash strategy, and $\mathbf{y} = \mathcal{S}(\mathbf{x})$ returns an infective strategy. Hence $\delta > 0$ and

$$h_{\mathbf{y}}(\mathbf{x}, \delta) + \pi(\mathbf{y} - \mathbf{x}|\mathbf{x}) \geq \pi(\mathbf{y} - \mathbf{x}|\mathbf{x}) > 0$$

(in fact, if $\delta < 1$, then even $h_{\mathbf{y}}(\mathbf{x}, \delta) = 0$), so that we obtain a strict increase of the population payoff. On the other hand, if $\pi(\mathbf{x}^{(t+1)}) = \pi(\mathbf{x}^{(t)})$, then the above equation implies $\delta = 0$ or $h_{\mathbf{x}}(\mathbf{x}, \delta) = \pi(\mathbf{y} - \mathbf{x}|\mathbf{x}) = 0$, due to nonnegativity of both quantities above. In particular, we have $\delta = 0$ or $\pi(\mathbf{y} - \mathbf{x}|\mathbf{x}) = 0$. In both cases, $\mathbf{y} = \mathcal{S}(\mathbf{x})$ cannot be infective for $\mathbf{x}$. Thus $\Upsilon(\mathbf{x}) = \emptyset$ and $\mathbf{x}$ must be a fixed point, according to Theorem 2. This establishes the last assertion of the theorem.

Theorem 3 shows that by running InImDyn , under symmetric payoff function, we strictly increase the population payoff until we reach a Nash equilibrium at a fixed point. This of course holds for any selection function $\mathcal{S}(\mathbf{x})$ satisfying (3). However, the way we choose $\mathcal{S}(\mathbf{x})$ may affect the efficiency of the dynamics. The next section introduces a particular selection function that leads to a well-performing dynamics for our purposes.

5 A Pure Strategy Selection Function

Depending on how we choose the function $\mathcal{S}(\mathbf{x})$ in (2), we may obtain different dynamics. One in particular, which is simple and leads to nice properties, consists in allowing only infective pure strategies.

Given a population $\mathbf{x}$, we define the co-strategy of $\mathbf{e}^i$ with respect to $\mathbf{x}$ as

$$\overline{\mathbf{e}^i}_{\mathbf{x}} = \frac{x_i}{x_i - 1}(\mathbf{e}^i - \mathbf{x}) + \mathbf{x}\,.$$

Note that if $\pi(\mathbf{e}^i - \mathbf{x}|\mathbf{x}) \neq 0$ then either $\mathbf{e}^i \in \Upsilon(\mathbf{x})$ or $\overline{\mathbf{e}^i}_{\mathbf{x}} \in \Upsilon(\mathbf{x})$.

Consider the strategy selection function $\mathcal{S}_{Pure}(\mathbf{x})$, which finds a pure strategy i maximizing $|\pi(\mathbf{e}^i - \mathbf{x}|\mathbf{x})|$, and returns $\mathbf{e}^i$, $\overline{\mathbf{e}^i}_{\mathbf{x}}$ or $\mathbf{x}$ according to whether $\pi(\mathbf{e}^i - \mathbf{x}|\mathbf{x})$ is positive, negative or zero. Let $\mathcal{M}(\mathbf{x})$ be a pure strategy such that

$$\mathcal{M}(\mathbf{x}) \in \underset{i=1,\ldots,n}{\arg\max}\, |\pi(\mathbf{e}^i - \mathbf{x}|\mathbf{x})|\,.$$

Then $\mathcal{S}_{Pure}(\mathbf{x})$ can be written as

$$\mathcal{S}_{Pure}(\mathbf{x}) = \begin{cases} \mathbf{e}^i & \text{if } \pi(\mathbf{e}^i - \mathbf{x}|\mathbf{x}) > 0 \text{ and } i = \mathcal{M}(\mathbf{x}) \\ \overline{\mathbf{e}^i}_{\mathbf{x}} & \text{if } \pi(\mathbf{e}^i - \mathbf{x}|\mathbf{x}) < 0 \text{ and } i = \mathcal{M}(\mathbf{x}) \\ \mathbf{x} & \text{otherwise}\,. \end{cases}$$

Note that the search space for an infective strategy is reduced from Δ to a finite set. Therefore, it is not obvious that $\mathcal{S}_{Pure}(\mathbf{x})$ is a well-defined selection function, i.e., it satisfies (3). The next theorem shows that indeed it is.

Proposition 2. *Let $\mathbf{x} \in \Delta$ be a population. There exists an infective strategy for $\mathbf{x}$, i.e., $\Upsilon(\mathbf{x}) \neq \emptyset$, if and only if $\mathcal{S}_{Pure}(\mathbf{x}) \in \Upsilon(\mathbf{x})$.*

Proof. Let $\mathbf{y} \in \Upsilon(\mathbf{x})$. Then $0 < \pi(\mathbf{y} - \mathbf{x}|\mathbf{x}) = \sum_{i=1}^{n} y_i \pi(\mathbf{e}^i - \mathbf{x}|\mathbf{x})$. But this implies that there exists at least one infective pure strategy for $\mathbf{x}$, i.e., $\mathbf{e}^i \in \Upsilon(\mathbf{x})$ for some $i = 1, \ldots, n$. The converse trivially holds.

A fixed point of InImDyn is *asymptotically stable* if any trajectory starting sufficiently close to $\mathbf{x}$ converges to $\mathbf{x}$.

Theorem 4. *A state $\mathbf{x}$ is asymptotically stable for* InImDyn *with $\mathcal{S}_{Pure}$ as strategy selection function if and only if $\mathbf{x}$ is an ESS, provided that the payoff matrix is symmetric.*

Proof. If the payoff matrix is symmetric, every accumulation point of InImDyn with $\mathcal{S}_{Pure}$ is a Nash equilibrium [13]. Moreover *ESS*s are strict local maximizers of $\pi(\mathbf{x})$ over Δ and vice versa [10].

If $\mathbf{x}$ is asymptotically stable, then there exists a neighborhood U of $\mathbf{x}$ in Δ such that any trajectory starting in U converges to $\mathbf{x}$. By Theorem 3 this implies that $\pi(\mathbf{x}) > \pi(\mathbf{y})$ for all $\mathbf{y} \in U$, $\mathbf{y} \neq \mathbf{x}$. Hence, $\mathbf{x}$ is a strict local maximizer of $\pi(\mathbf{x})$ and therefore $\mathbf{x}$ is an ESS.

Conversely, if $\mathbf{x}$ is an ESS then $\mathbf{x}$ is a strict local maximizer of $\pi(\mathbf{x})$ and an isolated Nash equilibrium. Hence, there exists a neighborhood U of $\mathbf{x}$ in Δ where $\pi(\mathbf{x})$ is strictly concave and $\mathbf{x}$ is the only accumulation point. This together with Theorem 3 implies that any trajectory starting in U will converge to $\mathbf{x}$. Hence, $\mathbf{x}$ is asymptotically stable.

This selection function exhibits the nice property of rendering the complexity per iteration of our new dynamics linear in both space and time, as opposed to the replicator dynamics, which have quadratic space/time complexity per iteration.

Theorem 5. *Given the iterate $\mathbf{x}^{(t)}$ and its linear transformation $A\mathbf{x}^{(t)}$, both space and time requirement of one iteration step is linear in n, the number of objects.*

Proof. Again abbreviate $\mathbf{x} = \mathbf{x}^{(t)}$. Now, given $A\mathbf{x}$ we can straightforwardly compute in linear time and space $\pi(\mathbf{x})$ and $\mathcal{S}_{Pure}(\mathbf{x})$. Assume that $\mathcal{S}_{Pure}(\mathbf{x}) = \mathbf{e}^i$, then the computation of $\delta_{\mathbf{e}^i}(\mathbf{x})$ has a linear complexity, since $\pi(\mathbf{x} - \mathbf{e}^i|\mathbf{x}) = (A\mathbf{x})_i - \pi(\mathbf{x})$ and $\pi(\mathbf{e}^i - \mathbf{x}) = a_{ii} - 2A\mathbf{x} + \pi(\mathbf{x})$. Moreover, $A\mathbf{x}^{(t+1)}$ can be also computed in linear time and space since

$$A\mathbf{x}^{(t+1)} = \delta_{\mathbf{e}^i}(\mathbf{x})\,[A_i - A\mathbf{x}] + A\mathbf{x}\,,$$

where A_i is the ith column of A. Similar arguments hold if $\mathcal{S}_{Pure}(\mathbf{x}) = \overline{\mathbf{e}^i}_{\mathbf{x}}$. Indeed,

$$\pi(\overline{\mathbf{e}^i}_{\mathbf{x}} - \mathbf{x}|\mathbf{x}) = \frac{x_i}{x_i - 1}\pi(\mathbf{e}^i - \mathbf{x}|\mathbf{x})\,,$$

$$\pi(\overline{\mathbf{e}^i}_{\mathbf{x}} - \mathbf{x}) = \left(\frac{x_i}{x_i - 1}\right)^2 \pi(\mathbf{e}^i - \mathbf{x})\,,$$

and finally,

$$A\mathbf{x}^{(t+1)} = \left(\frac{x_i}{x_i - 1}\right) \delta_{\overline{\mathbf{e}^i}_{\mathbf{x}}}(\mathbf{x})\,[A_i - A\mathbf{x}] + A\mathbf{x}\,.$$

Hence the result.

The only step of quadratic complexity is the first one, where we need to compute $A\mathbf{x}^{(0)}$. Even this can be reduced to linear complexity, if we start from a pure strategy $\mathbf{e}^i$, in which case we have $A\mathbf{x}^{(0)} = A_i$. Note that the latter is impossible, e.g., for the replicator dynamics.

6 Experimental Results

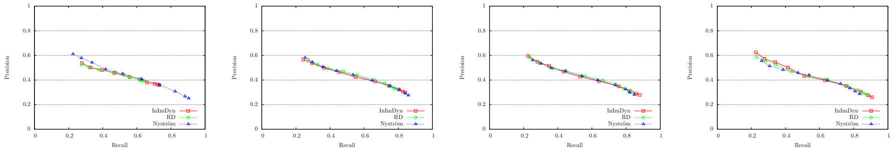
In order to test the effectiveness of our algorithm, we present experiments on some computer vision problems, which have been attacked using the dominant-set framework or related quadratic optimization problems. Our aim is to show the computational gain over the standard algorithm used in the literature, namely the replicator dynamics (RD). Specifically, we present comparisons on image segmentation [1] and region-based hierarchical image matching [8].

The stopping criterion adopted with our dynamics is a measure of the accuracy of the Nash equilibrium, which is given by $\epsilon(\mathbf{x}) = \sum_i \min\left\{x_i, \pi(\mathbf{x} - \mathbf{e}^i|\mathbf{x})\right\}^2$. Indeed, $\epsilon(\mathbf{x})$ is 0 if and only if $\mathbf{x}$ is a Nash equilibrium. In the experiments, we stopped the dynamics at accurate solutions, namely when $\epsilon(\mathbf{x}) < 10^{-10}$. As for RD, we stopped the dynamics either when $\epsilon(\mathbf{x}) < 10^{-10}$ or when a maximum number of iterations was exceeded.

6.1 Image Segmentation

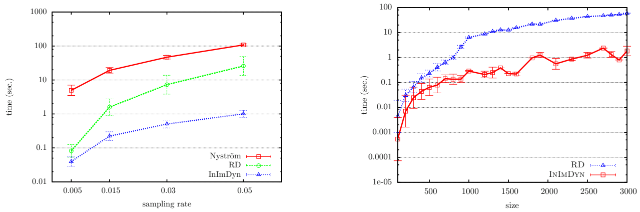
We performed image segmentation experiments over the whole Berkeley dataset [14] using the dominant-set framework as published in [1]. The affinity between two pixels i and j was computed based on color and using the standard Gaussian kernel. Our InImDyn algorithm was compared against standard replicator dynamics (RD) [1] (using the out-of-sample extension described in [15]) as well as the Nyström method [16]. The algorithms were coded in C and run on a AMD Sempron 3 GHz computer with 1GB RAM. To test the behavior of the algorithms under different input sizes we performed experiments at different pixel sampling rates, namely 0.005, 0.015, 0.03 and 0.05, which roughly correspond to affinity matrices of size 200, 600, 1200 and 2000, respectively. Since the Nyström method, as opposed to the dominant set approach, needs as input the desired number of clusters, we selected an optimal one after a careful tuning phase.

In Figure 2(a) we report (in logarithmic scale) the average computational times (in seconds) per image obtained with the three approaches. The computational gain of InImDyn over the replicator dynamics is remarkable and it clearly increases at larger sampling rates. It is worth mentioning that InImDyn other than being faster, achieved also better approximations of Nash equilibriums as



(a) s.r. 0.005 (b) s.r. 0.015 (c) s.r. 0.03 (d) s.r. 0.05

Fig. 1. Precision/Recall plots obtained on the Berkeley Image Database (s.r.=sampling rate)



(a) Image segmentation

(b) Image matching

Fig. 2. Average execution times (in logarithmic scale) for the image segmentation and region-based hierarchical image matching applications

opposed to RD. As for the quality of the segmentation results, we report in Figure 1 the average precision/recall obtained in the experiment with the different sampling rates. As can be seen, all the approaches perform equivalently, in particular RD and InImDyn achieved precisely the same results as expected.

6.2 Region-Based Hierarchical Image Matching

In [8] the authors present an approach to region-based hierarchical image matching, aimed at identifying the most similar regions in two images, according to a similarity measure defined in terms of geometric and photometric properties. To this end, each image is mapped into a tree of recursively embedded regions, obtained by a multiscale segmentation algorithm. In this way the image matching problem is cast into a tree matching problem, that is solved recursively through a set of sub-matching problems, each of which is then attacked using replicator dynamics (see [8] for details). Given that typically hundreds of sub-matching problems are generated by a single image matching instance, it is of primary importance to have at one's disposal a fast matching algorithm. This makes our solution particularly appealing for this application.

We compared the running time of InImDyn and RD over a set of images taken from the original paper [8]. We run the experiments on a machine equipped with 8 Intel Xeon 2.33 GHz CPUs and 8 GB RAM. Figure 2(b) shows the average computation times (in seconds) needed by RD and InImDyn to solve the set of sub-matching problems generated from 10 image matching instances. Since each image matching problem generated sub-matching problems of different sizes, we grouped the instances having approximately the same size together. We plotted the average running time within each group (in logarithmic scale) as a function of the instance sizes and reported the standard deviations as error bars. Again, as can be seen, InImDyn turned out to be orders of magnitude faster than RD.

7 From QPs to StQPs

Although in this paper we focused mainly on dominant sets, which lead to quadratic optimization problems over the standard simplex (StQPs), the proposed approach is indeed more general and can be applied to a large class of quadratic programming problems (QPs), instances of which frequently arise in computer vision and pattern recognition.

In fact, consider a general QP over a bounded polyhedron

$$\max\left\{\frac{1}{2}\mathbf{x}^\top Q\mathbf{x} + \mathbf{c}^\top\mathbf{x} \,:\, \mathbf{x} \in M\right\} , \tag{4}$$

where $M = \operatorname{conv}\{\mathbf{v}_1, \ldots, \mathbf{v}_k\} \subseteq \mathbb{R}^n$ is the convex hull of the points $\mathbf{v}_i$, which form the columns of a $n \times k$-matrix V. Then we can write the QP in (4) as the following StQP:

$$\max\left\{\mathbf{y}^\top \hat{Q}\mathbf{y} \,:\, \mathbf{y} \in \Delta\right\} ,$$

where $\hat{Q} = \frac{1}{2}\left(V^\top QV + \mathbf{e}^\top V^\top \mathbf{c} + \mathbf{c}^\top V\mathbf{e}\right)$.

Thus every QP over a polytope can be expressed as an StQP. This approach is of course only practical if the vertices V are known and k is not too large. This is the case of QPs over the ℓ^1 ball, where $V = [I|-I]$, I the $n \times n$ identity matrix and $\Delta \subset \mathbb{R}^{2n}$ and, more generally, for box-constrained QPs [17]. However, even for general QPs, where the constraints are expressed as $M = \{\mathbf{x} \in \mathbb{R}^n_+ \,:\, A\mathbf{x} = b\}$, we can use StQP as a relaxation without using all vertices (see [18] for details).

Acknowledgements

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250).

References

1. Pavan, M., Pelillo, M.: Dominant sets and pairwise clustering. IEEE Trans. Pattern Anal. Machine Intell. 29(1), 167–172 (2007)
2. Neumann, J., von Cramon, D.Y., Forstmann, B.U., Zysset, S., Lohmann, G.: The parcellation of cortical areas using replicator dynamics in fMRI. NeuroImage 32(1), 208–219 (2006)
3. Wang, M., Ye, Z.L., Wang, Y., Wang, S.X.: Dominant sets clustering for image retrieval. Signal Process. 88(11), 2843–2849 (2008)
4. Hamid, R., Johnson, A., Batta, S., Bobick, A., Isbell, C., Coleman, G.: Detection and explanation of anomalous activities: representing activities as bags of event n-grams. In: CVPR, vol. 1, pp. 20–25 (2005)
5. Frommlet, F.: Tag SNP selection based on clustering according to dominant sets found using replicator dynamics. Adv. in Data Analysis (in press, 2010)
6. Wei, Q.D., Hu, W.M., Zhang, X.Q., Luo, G.: Dominant sets-based action recognition using image sequence matching. In: ICIP, vol. 6, pp. 133–136 (2007)
7. Albarelli, A., Torsello, A., Rota Bulò, S., Pelillo, M.: Matching as a non-cooperative game. In: ICCV (2009)
8. Todorovic, S., Ahuja, N.: Region-based hierarchical image matching. Int. J. Comput. Vision 78(1), 47–66 (2008)

9. Torsello, A., Rota Bulò, S., Pelillo, M.: Grouping with asymmetric affinities: a game-theoretic perspective. In: CVPR, pp. 292–299 (2006)
10. Weibull, J.W.: Evolutionary game theory. Cambridge University Press, Cambridge (1995)
11. Bomze, I.M., Pötscher, B.M.: Game Theoretical Foundations of Evolutionary Stability. Springer, Heidelberg (1989)
12. Bomze, I.M., Weibull, J.W.: Does neutral stability imply Lyapunov stability? Games and Econ. Behaviour 11, 173–192 (1995)
13. Rota Bulò, S.: A game-theoretic framework for similarity-based data clustering. PhD thesis, University of Venice (2009)
14. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV, July 2001, vol. 2, pp. 416–423 (2001)
15. Pavan, M., Pelillo, M.: Efficient out-of-sample extension of dominant-set clusters. NIPS 17, 1057–1064 (2005)
16. Fowlkes, C., Belongie, S., Malik, F.C.J.: Spectral grouping using the Nyström method. IEEE Trans. Pattern Anal. Machine Intell. 26(2), 214–225 (2004)
17. Pardalos, P.M.: Quadratic problems defined on a convex hull of points. BIT Num. Math. 28(2), 323–329 (1988)
18. Bomze, I.M., Locatelli, M., Tardella, F.: Efficient and cheap bounds for (standard) quadratic optimization. Technical report, University "La Sapienza" of Rome (2005)

What Is the Complexity of a Network? The Heat Flow-Thermodynamic Depth Approach*

Francisco Escolano[1], Miguel A. Lozano[1],
Edwin R. Hancock[2], and Daniela Giorgi[3]

[1] University of Alicante
{sco,malozano}@dccia.ua.es
[2] University of York
erh@cs.york.ac.uk
[3] IMATI-CNR Genova
daniela@ge.imati.cnr.it

Abstract. In this paper we establish a formal link between network complexity in terms of Birkhoff-von Neumann decompositions and heat flow complexity (in terms of quantifying the heat flowing through the network at a given inverse temperature). We propose and proof characterization theorems and also two fluctuation laws for sets of networks. Such laws emerge from studying the implicacions of the Fluctuation Theorem in heat-flow characterization. Furthermore, we also define heat flow complexity in terms of thermodynamic depth, which results in a novel approach for characterizing networks and quantify their complexity In our experiments we characterize several protein-protein interaction (PPI) networks and then highlight their evolutive differences, in order to test the consistence of the proposed complexity measure in terms of the second law of thermodynamics.

1 Introduction

The quantification of the *intrinsic complexity* of networks has attracted significant attention, in a number of fields including complexity science, pattern recognition and machine learning, due to its fundamental practical importance. Some complexity characterizations rely on spectral graph theory (see [1] for applications in computational biology, [2] for biological, social and other kinds of networks, and [3][4] for applications to pattern recognition). The work presented herein concerns the global analysis of structural patterns but not their fine discriminability. For instance, two undirected complete graphs (the simplest ones according to our approach) of very different size should have a similar complexity; however in terms of their discrimination, they will be different for an inexact graph matching strategy. However, complexity can be used as a MDL-principled measure for graph learning. In addition, fine discrimiability methods like matching are not suitable for finding global characterizations of structural patterns

* Funded by the Project TIN2008- 04416/TIN of the Spanish Government.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 286–295, 2010.

like identifying clusters corresponding to sub-populations. In the particular case of Protein-Protein Interaction (PPI) networks, we have found in a preliminary study that networks with similar complexity are quite different in terms of edit distance. Therefore our contribution fits a global (low-frequency) methodology for analysis of graphs. In this regard, spectral graph theory is a recurrent formal tool. Recent extensive use of spectral graph theory is due to: a) that it explains some previous approaches (the number of spanning trees, path-length distribution, clusterization, and so on) from a random walks perspective and b) that it is flexible enough to allow the development of new characterizations. In this paper we explore the connection between convex polytopes (and those of the Birkhoff type in particular), heat kernels in graphs, the well known thermodynamic depth approach to complexity, and network complexity itself. Some work in this direction has been done recently [3], but no formal connections between polytopes and heat-flow characterization of structural entropy [3] has been developed so far. Our main contribution here is to formally specify the complexity profiles of both approaches to structural complexity, showing that they have a qualitatively similar behavior and that the complexity corresponding to the maximum entropy (ME) Birkhoff-von Neumann decomposition is derived from that corresponding to the maximum flow. Thus, the phase-transition point always exists and it is characterized by such maxima. Moreover we establish links between heat flow complexity, the fluctuation theorem and thermodynamic depth. We also apply this characterization to PPI networks.

2 Polytopal vs. Heat Flow Complexity

Theorem 1 *(*Birkhoff-von Neumann (BvN) *[5]). Let $\mathcal{B}_n$ be is the set of doubly stochastic matrices $B = [b_{ij}]_{n\times n}$ of dimension $n \times n$ (*Birkhoff polytope*). Then every doubly stochastic matrix (*DSM*) B can be expressed as a convex combination of permutation matrices (*PM*):*

$$B = \sum_\alpha p_\alpha P_\alpha,\ \forall B \in \mathcal{B}_n \text{ and } \begin{array}{c}\sum_\alpha p_\alpha = 1\\ p_\alpha \geq 0\ \forall \alpha\end{array}.$$

Thus $\mathcal{B}_n$ is the convex hull of the set of the $n\times n$ permutation matrices. However, the representation of a DSM in terms of many PMs is not unique because $\mathcal{B}_n$ is not a simplex. The barycenter of $\mathcal{B}_n$ is the van der Waerden constant matrix B_* with all entries equal to $1/n$.

Theorem 2 *(*Agrawal, Wang & Ye *[6]). The maximum entropy (ME) BvN decomposition of a DSM B is the solution to the left optimization problem below (primal) whose dual one is on the right:*

$$\begin{array}{ll}\min\ \ \sum_{\alpha\in\mathcal{S}_n} p_\alpha(\log p_\alpha - 1) & \max\ B : Y - 1\\ s.t. \sum_\alpha p_\alpha P_\alpha \leq B & s.t.\ \sum_\alpha e^{(Y:P_\alpha)} P_\alpha \leq B\\ \quad p_\alpha \geq 0 & 0 \geq Y_{ij} \geq -n\frac{\log n}{b_{min}}\ \forall i,j\end{array}$$

where $\mathcal{S}_n$ is the set of permutations of $\{1, 2, \ldots, n\}$, $X : Z = \sum_{ij} X_{ij} Z_{ij} = trace(XZ^T)$ is the Frobenius inner product, $Y \in R^{n\times}$, a matrix of Lagrange multipliers each corresponding to one constraint (component) in $B = \sum_\alpha p_\alpha P_\alpha$, and $b_{min} = \min\{B_{ij}\}$.

In [6] it is shown how to solve approximately the dual of the ME problem. In practice, however, instead of finding a unique representation for B it is preferable to obtain greedily just one of them. To that end, the constructive proof of the BvN theorem is used. This is the origin of *polytopal complexity* [3].

Definition 1 *(*Polytopal Complexity *[3]). Given $G = (V, E)$, an undirected and unweighted graph with diffusion kernel $K^\beta(G)$, and* BvN *decomposition $K^\beta(G) = \sum_{\alpha=1}^{\gamma} p_\alpha P_\alpha$, we define the* polytopal complexity of G *as the β-dependent function*

$$\mathcal{BC}^\beta(G) = \frac{H(\mathcal{P})}{\log_2 n} = \frac{\log_2 \gamma + D(\mathcal{P}||\mathcal{U}_\gamma)}{\log_2 n} \,, \tag{1}$$

where $\mathcal{P} = \{p_1, \ldots, p_\gamma\}$ is the probability density function (pdf) induced by the decomposition, $H(.)$ the entropy and $D(.)$ the Kullback-Leibler divergence $D(\mathcal{P}||\mathcal{Q}) = \sum_\alpha p_\alpha \log \frac{p_\alpha}{q_\alpha}$.

In [3] it is argued that the typical signature is heavy tailed, monotonically increasing from 0 to $\beta^+ \equiv \arg\max\{\mathcal{BC}^\beta(G)\}$ and either monotonically decreasing or stable from β^+ to ∞ where $\mathcal{BC}^\beta(G) = 1$ is reached. Thus, β^+ represents the most significant *topological phase transition* regarding the impact of the diffusion process in the topology of the input graph. However, no characterization theorem has been enunciated so far in order to validate the latter assumptions. In addition, in [4] it is showed that the $O(n^5)$ computational complexity of the greedy BvN decomposition for each β precludes the use of the descriptor for the practical analysis of complex networks. Thus, a new descriptor, qualitatively similar but more efficient than the current one, and also providing a simpler analytical framework, is needed.

Definition 2 *(*Heat Flow Complexity *[4]). Given $G = (V, E)$ with $|V| = n$ and adjacency matrix A. The diffusion kernel is $K^\beta(G) = \exp(-\beta\mathcal{L}) \equiv \Phi\Lambda\Phi^T$, being $\Lambda = diag(e^{-\beta\lambda_1}, e^{-\beta\lambda_2}, \ldots, e^{-\beta\lambda_n})$, and $\lambda_1 = 0 \leq \lambda_2 \leq \ldots \leq \lambda_n$ are the eigenvalues of $\mathcal{L}$. Therefore, the* total heat flowing through the graph *at a given β is:*

$$F^\beta(G) = \sum_{i=1}^{n} \sum_{j\neq i}^{n} \delta_{ij} \underbrace{\left(\sum_{k=1}^{n} \phi_k(i)\phi_k(j) e^{-\lambda_k \beta} \right)}_{K_{ij}^\beta} \,, \tag{2}$$

where $\delta_{ij} = 1$ iff $(i, j) \in E$. Then, heat flow complexity *is defined as:*

$$\mathcal{FC}^\beta(G) = \frac{\log_2(1 + F^\beta(G))}{\log_2 n} \,. \tag{3}$$

3 Characterization of Polytopal and Flow Complexity

3.1 Characterization of Phase Transition

Theorem 3 *(*Phase-Transition Point*). Let $G = (V, E)$ be a graph with $|V| = n$ and edge-set E. Then, there exists a unique finite inverse temperature $\beta^+ \geq 0$ so that β^+ is the maximal value for which the sum of the off-diagonal elements of the diffusion kernel (or Gram matrix) on graph G is less that the sum of the on-diagonal elements. In other words, there exists an unique $\beta^+ \geq 0$ so that $\sum_{i=1}^{n}\sum_{j\neq i}^{n} K_{ij}^{\beta^+} < trace(K^{\beta^+})$, and $\sum_{i=1}^{n}\sum_{j\neq i}^{n} K_{ij}^{\beta} \geq trace(K^{\beta})\ \forall\beta > \beta^+$.*

Proof. Let us analyze the behavior of the function $\Xi^{\beta} = trace(K^{\beta}) - \sum_{i=1}^{n}\sum_{j\neq i}^{n} K_{ij}^{\beta}$. The analysis of the limiting cases $K^0 = I_n$ and $K^{\infty} = \mathcal{B}_*$ yields $\Xi^0 = n$ and $\Xi^{\infty} = -n$. Actually $-n$ may be reached as soon as the kernel converges to $\mathcal{B}_*$ (reaches the *equilibrium point*). Local maxima of Ξ^{β} are precluded by the monotonic nature of the diffusion process and therefore Ξ^{β} is a monotonically decreasing function with a minimum at equilibrium. Thus, the PTP exists just before the zero-crossing $\Xi^{\beta} = 0$ and it is unique. □

The existence of a unique PTP is key to relating heat flow and maximal entropy.

Theorem 4 *(*Phase-Transition*). Let $\beta^+ > 0$ define a PTP. Then, the heat flow $F^{\beta^+}(G)$ corresponding to the PTP is maximal among all choices of β. Moreover, this implies that the entropy $H^{\beta^+}(\mathcal{P})$ with $\mathcal{P} = \{p_1, \ldots, p_{\gamma}\}$ corresponding to the maximal entropy BvN decomposition of $K^{\beta^+}(G) = \sum_{\alpha=1}^{\gamma} p_{\alpha}P_{\alpha}$ is maximal over β.*

Proof (Flow Maximality at PTP). Consider $\beta < \beta^+$ and suppose that $F^{\beta} > F^{\beta^+}$, that is, $\sum_{ij}\delta_{ij}K_{ij}^{\beta} > \sum_{ij}\delta_{ij}K_{ij}^{\beta^+}$. We can write $\sum_{ij}\delta_{ij}K_{ij} = A : K$, where A the adjacency matrix of G and $X : Y = \sum_{ij} X_{ij}Y_{ij}$ denotes here the Frobenius inner product. It follows that $A : K^{\beta} > A : K^{\beta^+}$. All the off-diagonal elements of K_{β} decrease at β, with respect to their values at β^+ due to the diffusion process. As a result, the sum of off-diagonal elements of K^{β} is smaller than the sum of off-diagonal elements of K^{β^+}. Moreover, as on-diagonal elements are zero on A, we have that $A : K^{\beta} \leq A : K^{\beta^+}$ which is a contradiction. Therefore $F^{\beta} \leq F^{\beta^+}$.

Consider now the case $\beta > \beta^+$ and also $F^{\beta} > F^{\beta^+}$. Then, we should have that $A : K^{\beta} > A : K^{\beta^+}$ which is consistent with the fact that the sum of off-diagonal elements is more and more greater or equal than the sum of on-diagonal elements as β increases. This is due to the fact that off-diagonal values which are not associated to an edge in the graph increase whereas on-diagonal ones decrease. However the individual values of both diagonal and off-diagonal elements are bounded by $1/n$, and tend to such value as β increases. Furthermore, when all values reach $1/n$ at a given inverse temperature, such equilibrium state remains constant for greater values of the inverse temperature. If the equilibrium is reached later than β, only off-diagonal elements which are not associated to an edge (but to a path) increase. However, edge-associated off-diagonal elements decrease which implies $A : K^{\beta} < A : K^{\beta^+}$, that is $F^{\beta} < F^{\beta^+}$ which is a contradiction. If β corresponds to an inverse temperature beyond the equilibrium value, then we have that $F_\beta = \frac{2|E|}{n}$ which must be greater than F_{β^+} (where the sum of the on-diagonal elements is greater than that of the off-diagonal elements) and the off-diagonal elements associated with edges have a greater value since $\beta^+ < \beta$[1]. Therefore we have again a contradiction. The limiting case is that equilibrium is reached at $\beta = \beta^+$. In that case we have also contradiction because $F^\beta = F\beta^+ = \frac{2|E|}{n}$. From the contradiction in the two cases $\beta < \beta^+$ and $\beta > \beta^+$, we have $F^{\beta^+} > F^\beta$ for all values of $\beta \in [0, +\infty)$.

Proof (Entropy Maximality at PTP). Let H^β the entropy corresponding to the maximum entropy BvN decomposition for a given β. Now, we have to prove that $F^{\beta^+} > F^\beta \Rightarrow H^{\beta^+} > H^\beta$, for any β. The maximum entropy BvN decomposition yields $p_\alpha = e^{Y:P_\alpha}$, and $Y \in \mathbb{R}^{n\times n}$ is the matrix of Lagrange multipliers satisfying the condition $0 \leq K_\beta : Y = trace(K_\beta Y^T) \leq -n\log n$ (see proof of Lemma 5 in [6]). Such a BvN decomposition is unique for the given value of β, and the Lagrange multipliers which correspond to dual variables associated to the $n\times n$ constraints $K^\beta(G) = \sum_{\alpha=1}^{\gamma} p_\alpha P_\alpha$. Let $k_{min} = \min_{ij}\{K_{ij}^\beta\}$ be the minimal component in K_β. Then, every multiplier satisfies the bound $0 \geq Y_{ij} \geq -\frac{n\log n}{k_{min}}$. Consequently, those kernel elements that are zero or close-to-zero may enlarge the bounds (see the dual problem) up to $-\infty$ (when $k_{min} = 0$). These large bounds imply that $p_\alpha \rightarrow 0$ for some value of α (the exponential argument in $e^{Y:P_\alpha}$ may be $-\infty$), but not necessarily to all of them because of the different structures of the associated permutation matrices P_α in each case. This occurs at every β for the same graph G. In the limiting cases of $\beta = 0$ and $\beta \rightarrow +\infty$ we have, respectively, $p_\alpha = 1$ for the unique $P_\alpha = I_n$ and $p_\alpha = 1/n$ (all kernel components are $1/n$) for the n permutation matrices, where $H^0 = 0$ and $H^{+\infty} = \log_2 n$. The respective flows are $F^0 = 0$ and $F^{+\infty} = \frac{2|E|}{n}$.

Proving that $H^\beta < H^{\beta^+}$ for each $\beta \neq \beta^+$ is equivalent to prove $-n\log n \leq K^\beta : Y^\beta < K^{\beta^+} : Y^{\beta^+} \leq 0$ for each $\beta \neq \beta^+$, since we are maximizing $K : Y - 1$ in the dual problem, being Y^β and Y^{β^+} respectively the optimal Lagrange multipliers corresponding to the maximum entropy BvN decompositions at β and β^+. This means that the multipliers (which are all negative) are set to their maximal (close-to-zero) values provided that the decomposition constraints are satisfied. Given their theoretical bounds $0 \geq Y_{ij}^{\beta^+} \geq \frac{-n\log n}{k_{min}^{\beta^+}}$ and $0 \geq Y_{ij}^\beta \geq \frac{-n\log n}{k_{min}^\beta}$, the Lagrange multipliers can be arbitrarily close to zero. Each multiplier is related to a kernel component (the Frobenius inner product is the sum of the elements of the matrix resulting from the Hadamard product) and both kernels are DSMs. Hence, we must only set $Y_{ij}^{\beta^+}$ and Y_{ij}^β to their minimal values when $K_{ij}^{\beta^+} = 0$ and $K_{ij}^\beta = 0$ so that each Frobenius product is maximized (given that p_α is defined by the exponential of $Y : P_\alpha$).

For $\beta < \beta^+$, As β^+ defines a PTP, we have that the sum of the $n^2 - n$ off-diagonal values in K^β is lower than the n on-diagonal elements. Therefore we obtain K^β :

[1] Furthermore, for large β we have that $K_\beta = e^{-\beta\lambda_2}\phi_2\phi_2^T$, where ϕ_2 is the Friedler vector.

$Y^\beta < K^{\beta^+} : Y^{\beta^+}$ which is due to the fact that, although the multipliers are chosen as close to zero as possible, the most negative multipliers must be assigned to the lower elements in K^β in order to maximize the Frobenius product. Although the less negative elements correspond with the (dominant) diagonal elements of K^β, they become more closer to zero than at β^+. There will be an increasing number of zero elements as $\beta \rightarrow 0$, since in these conditions we have $K^\beta = (I_n - \mathcal{L}\beta)$ which means that the on-diagonal elements will be closer to the unity, and we have the freedom to assign negative multipliers to increasingly small off-diagonal elements. The latter assigment yields a small $K_\beta : Y_\beta$. However, as we approach β^+, where off-diagonal elements start to dominate, it is more convenient to assign the close-to-zero multipliers to dominant elements and then the Frobenius product increases.

When $\beta > \beta^+$, the sum of off-diagonal values is greater to the sum of on-diagonal ones until the equilibrium point is reached. If in addition $A : K^{\beta^+} > A : K^\beta$ before equilibrium and recalling that on-diagonal elements at β are smaller than their values at β^+, we obtain $K^{\beta^+} : Y^{\beta^+} > K^\beta : Y^\beta$. This is due to: (i) that it is desirable to assign the closer-to-zero multipliers the off-diagonal elements, and the more negative ones to the diagonal in order to maximize the Frobenius product; (ii) that the latter assignment is increasingly infasible as β grows because of the increasing number of constraints over these multipliers as $\Xi^\beta = trace(K^\beta) - \sum_{i=1}^n \sum_{j\neq i}^n K_{ij}^\beta$ decreases. Under this latter condition, heat flow increases through the edges and establishes virtual paths (reachability) between those node pairs not connected by edges. As a result, there is an increase of the off-diagonal elements associated with indirect paths (rather than connecting edges) . An increasing number of close-to-zero multipliers are needed for the latter elements in order to maximize the Frobenius product. However, not all off-diagonal elements can have a close-to-zero multiplier and some of them will be very negative. If β is closer to β^+ than to the equilibrium point, the off-diagonal elements associated to indirect paths can be very negative and thus $K^{\beta^+} : Y^{\beta^+} > K^\beta : Y^\beta$. As β reaches the equilibrium point all the elements tend to $1/n$, which implies that all multipliers are almost equal but less or equal to any multiplier at β^+. Then, again $K^{\beta^+} : Y^{\beta^+} > K^\beta : Y^\beta$ even beyond the equilibrium point. Therefore, for $\beta \neq \beta^+$ we have $H^{\beta^+} \geq H^\beta$. □

3.2 The Fluctuation Laws

The fluctuation theorem (FT) states that the probability of *destroying entropy* in an isolated (macroscopic or microscopic) system decreases exponentially with time.Herein, as the β inverse temperature is assimilated to time t, and entropy H_β is assimilated to heat flow F_β, we do not have the case of *destroying entropy* in the sense of having a negative entropy. However, due to the existence of a PTP, for each flow trace F_β a *entropy production phase* $[0, \beta^+]$ and an *entropy stabilization/destruction phase* $(\beta^+, \beta^{max}]$. Entropy decayment after the PTP is due to the structure imposed by the network (but in the complete graph where there is no topological constrain) because structure means information. However, is there a formal relation between the rate of entropy production and that of entropy reduction in a set of networks representing, for instance, the same phenomenon like a PPI? We have found that (see the experimental section) for many sets of networks such a relation exists and also that it is linear. Moreover, besides linearity, there also exists an exponential decay.

Definition 3 *(*Fluctuation Laws*). Let $\Omega = \{G_i = (V_i, E_i)\}$ where for $G_i \in \Omega$ we have that $F_\beta(G_i)$ is the heat flow trace for $\beta \in [0, \beta^+]$ and $\langle \nabla F(G_i) \rangle_a^b$ is the average flow gradient between $\beta = a$ and $\beta = b$. Such set of networks satisties the* linear fluctuation law (LFL) *if exists $k > 0$ so that*

$$Pr\left(\langle \nabla F(G_i) \rangle_{\beta^+}^{\beta^{max}} = -k \langle \nabla F(G_i) \rangle_0^{\beta^+} \right) \approx 1,$$

that is,entropy variation decay is, with high probability, larger (in absolute value) as the entropy variation increase grows (the more entropy is produced at the beginning, the more is distroyed beyond the PTP) and variation decay is linear with respect to variation increasing along the population of networks. If in addition to satisfying LFL *, for every pair $(G_i, G_j), i \neq j$ so that $G_i, G_j \in \Omega$ and exists $\lambda > 0$ so that $Pr\left(\left| \langle \nabla F(G_i) \rangle_{\beta^+}^{\beta^{max}} - \langle \nabla F(G_j) \rangle_{\beta^+}^{\beta^{max}} \right| \right) = e^{-\lambda}$, we have that the set of networks satisfies the* linear fluctuation law with exponential decay (LFLED)*.*

4 Heat Flow - Thermodynamic Depth Complexity

The application of thermodynamic depth (TD) to characterize network complexity demands the formal specification of the micro-states whose history leads to the macro-state (of the network). Here we define such micro-states in terms of *expansion subgraphs.*

Definition 4 *(*Node History & Expansion Subgraphs*). Let $G = (V, E)$ with $|V| = n$. Then the* history of a node *$i \in V$ is $h_i(G) = \{e(i), e^2(i)), \ldots, e^p(i)\}$ where: $e(i) \subseteq G$ is the* first-order expansion subgraph *given by i and all $j \sim i$, $e^2(i) = e(e(i)) \subseteq G$ is the* second-order expansion *consisting on $z \sim j : j \in V_{e(i)}, z \notin V_{e(i)}$, and so on until p cannot be increased. If G is connected $e^p(i) = G$, otherwise $e^p(i)$ is the connected component to which i belongs.*

Every $h_i(G)$ defines a different causal trajectory leading to G itself, if it is connected, or to one of its connected components otherwise. Thus, in terms of TD the full graph G or the union of its connected components is the macro-state (macroscopic state). The *depth* of such macro-state relies on the variability of the causal trajectories leading to it. The higher the variability, the more complex it is to explain how the macro-state is reached and the deeper is this state. Therefore, in order to characterize each trajectory we combine the heat flow complexities of its expansion subgraphs by means of defining *minimal enclosing Bregman balls* (MEBB) [8]. Bregman divergences D_F define an asymmetric family of similarity measures, each one characterized by a strictly convex generator function $F : \mathcal{X} \rightarrow R^+$, where $\mathcal{X} \subseteq R^d$ is a convex domain, and d the data dimension (in this case the number of discretized β - inverse temperatures). Given two patterns (discretized functions in this case)

$\boldsymbol{f}$ and $\boldsymbol{g}$, $D_F(\boldsymbol{f}||\boldsymbol{g}) = F(\boldsymbol{f}) - F(\boldsymbol{g}) - (\boldsymbol{f} - \boldsymbol{f})^T\nabla F(\boldsymbol{f})$. Here, we use the I-Kullback-Leibler divergence $D_F(\boldsymbol{f}||\boldsymbol{g}) = \sum_{i=1}^{d} f_i \log\frac{f_i}{g_i} - \sum_{i=1}^{d} f_i + \sum_{i=1}^{d} g_i$ with $F(\boldsymbol{f}) = \sum_{i=1}^{d}(f_i \log f_i - f_i)$ (un-normalized Shannon entropy) which yields better results (more representative centroids of heat flow complexities) than other divergences/distorions like that of Itakura-Saito. When using the I-KL divergence in R^d, we have that $\nabla F(f_i) = \log f_i$ and also that $\nabla^{-1}F(f_i) = e^{f_i}$ (obviously the natural logarithm is assumed). Using these formal ingredients we define the *causal trajectory* in terms of MEBBs.

Definition 5 *(*Causal Trajectory*). Given $h_i(G)$, the heat flow complexity $\boldsymbol{f}_t = f(e^t(i))$ for the $t-th$ expansion of i, a generator F and a Bregman divergence D_F, the* causal trajectory *leading to G (or one of its connected components) from i is characterized by the center $\boldsymbol{c}_i \in R^d$ and radius $r_i \in R$ of the MEBB $\mathcal{B}^{\boldsymbol{c}_i, r_i} = \{\boldsymbol{f}_t \in \mathcal{X} : D_F(\boldsymbol{c}_i||\boldsymbol{f}_t) \leq r_i\}$.*

Solving for the center and radius implies finding $\boldsymbol{c}^*$ and r^* minimizing r subject to $D_F(\boldsymbol{c}_i||\boldsymbol{f}_t) \leq r\ \forall t \in \mathcal{X}$ with $|\mathcal{X}| = T$. Considering the Lagrange multipliers α_t we have that $\boldsymbol{c}^* = \nabla^{-1}F(\sum_{t=1}^{T} \alpha_t \boldsymbol{f}_t \nabla F(\boldsymbol{f}_t))$. The efficient algorithm in [8] estimates both the center and multipliers. This idea is closely related to Core Vector Machines [9], and it is interesting to focus on the non-zero multipliers (and their support vectors) used to compute the optimal radius. More precisely, the multipliers define a convex combination and we have $\alpha_t \propto D_F(\boldsymbol{c}^*||\boldsymbol{f}_t)$, and the radius is simply chosen as: $r^* = \max_{\alpha_t > 0} D_F(\boldsymbol{c}^*||\boldsymbol{f}_t)$.

Definition 6 *(*TD Network Depth*). Given $G = (V, E)$, with $|V| = n$ and all the n pairs $(\boldsymbol{c}_i, r_i)$, the* heat flow-thermodynamic depth complexity *of G is characterized by the MEBB $\mathcal{B}^{\boldsymbol{c},r} = \{\boldsymbol{c}_t \in \mathcal{X}_i : D_F(\boldsymbol{c}||\boldsymbol{c}_i) \leq r\}$ and $D_{min} = \min_{f \in \mathcal{B}^{\boldsymbol{c},r}} D_F(f^\infty||f)$, where $f^\infty = f(B_*) \in R^d$ is the van der Waerden complexity trace . As a result, the* TD depth of network *is given by $\mathcal{D}(G) = r \times D_{min}$.*

The above definitions of complexity and depth are highly consistent with summarizing node histories to find a global causal trajectory which is as tightly bounded as possible. Here, r quantifies the *historical uncertainty*: the smaller r the simpler (shallower) is G. However, this is not sufficient for structures because many networks with quite different complexities may have the same value of r. Therefore, we define the depth of the network complementing randomness as suggested in the thermodynamic depth approach. In our case, the projection of f^∞ on the MEBB preserves the definition of entropy in terms of the distance to the uniform distribution. The combinations or hierarchies of MEBBs have proved to be more effective than ball trees for nearest-neighbor retrieval [10]. In the computation of depths, the Legrendre duality (convex conjugate) is key because it establishes a one-to-one correspondence between the gradients ∇F and ∇F^{-1} due to the convexity of F. Therefore, the Bregman projection f of f^∞ on the the border of $\mathcal{B}^{\boldsymbol{c},r}$ lies on the curve $f_\theta^{-1} = \theta\nabla F(\boldsymbol{c}) + (1-\theta)\nabla F(f^\infty)$ with $\theta \in [0, 1]$ and $f_\theta = \nabla^{-1}F(f_\theta^{-1})$. The projection f be easily found (approximately) through bisection search on θ.

5 Experiments: TD of PPIs

We have designed an experimental section using PPIs extracted from STRING–8.2[2]. In a first experiment, we consider PPIs related to *histidine kinase*, a key protein in the development of signal transduction, corresponding to 10 species belonging to 10 phyla of bacteria. We select subjectively 3 PPIs (simple, complex and more-complex) from each species and compute their TDs. In 70% of the cases, TD matches intuition. When comparing with Estrada's spectral homogeneity descriptor [2] we also find that the ratio between intraclass and interclass variability is slightly better (smaller) for TD (0.6840 vs 0.7383). The second experiment consists of analyzing 222 PPIs, also related to histidine kinase, from 6 different groups (all the PPIs in the same group corresponds to the same species) with the following evolutive order (from older to more recent): *Aquifex* –4 PPIs, *Thermotoga*–4 PPIs, *Gram-Positive*–52 PPIs, *Cyanobacteria*–73 PPIs *Proteobacteria*–45 PPIs. There is an additional class (*Acidobacteria*—46 PPIs). Histogramming TDs reveals typically long tailed distributions with most of the TDs concentrated at a given point. Are these points ordered according to the evolutive order? This question can be answered by studying the cumulative distributions instead of the pdfs (Fig. 1-left/top). In such case, reaching the top (cumulative=1) soon indicates low TD whereas reaching it later indicates high TD. Then, it can be seen that the evolving complexity of the signal transduction mechanism driven by the histidine kinase is properly quantified by TD for

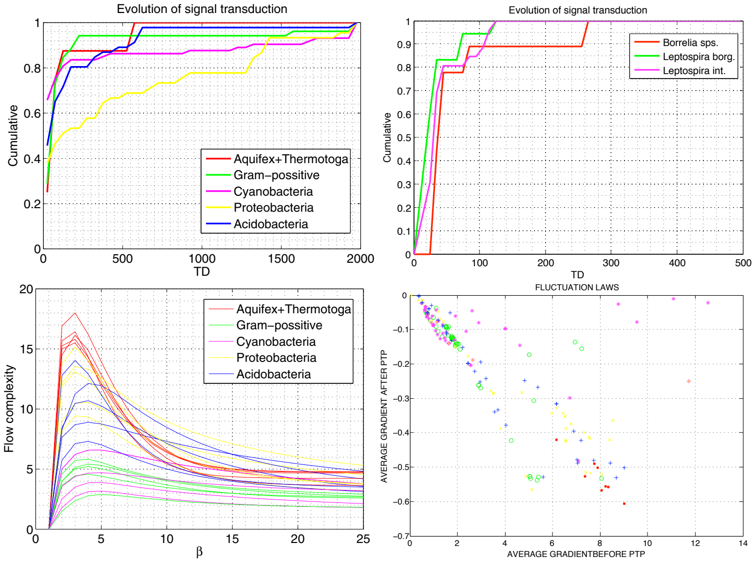


Fig. 1. PPI analysis with TD and Illustration of Fluctuation Laws (bottom-left)

[2] http://string.embl.de/

the 5 first phyla studied. However, the Acidobacterium sp. chosen seems older than Gram-Postive which seems not to be the case. In the bottom of Fig. 1-left/bottom we show some $\boldsymbol{c}_i$s of all classes, and their intraclass variability is low (similar shape). Thus, we can conclude that TD is a good principled tool for analying the complexity of networks. In a third experiment we analyze the cumulatives of three different species of the same phylum (76 PPIs of *Spirochaetes*) to check that the intra-species variability is low (Fig. 1-right/top). Finally, we show how the PPIs analyzed in the second experiment follow the fluctuation law, and some of them (we preserve the colors of Fig. 1-left/top) like *Cianobacteria* follow the LFLED.

6 Conclusions and Future Work

In this work, there are four contributions: a) the characterization heat flow complexity in terms of information theory, b) to define structural complexity in terms of Heat Flow-Thermodynamic Depth, c) to explore connections between the heat-flow thermodynamic depth and the fluctuation theorem and d) test the formal definition in terms of characterizing the evolution of Bacteria through quantifying the TD of their PPIs. Future work includes both exploring formal links with the Ihara Zeta function and studying different kind of networks.

References

[1] Banerjee, A., Jost, J.: Graph Spectra as a Systematic Tool in Computational Biology. Discrete Applied Mathematics 157(10), 27–40 (2009)

[2] Estrada, E.: Spectral Scaling and Good Expansion Properties in Complex Networks. Europhys. Lett. 73(4), 649 (2006)

[3] Escolano, F., Hancock, E.R., Lozano, M.A.: Birkhoff Polytopes, Heat kernels and Graph complexity. In: Proc. of ICPR, pp. 1–5 (2008)

[4] Escolano, F., Giorgi, D., Hancock, E.R., Lozano, M.A., Falcidieno, B.: Flow Complexity: Fast Polytopal Graph Complexity and 3D Object Clustering. In: Proc. of GbRPR, pp. 253–262 (2009)

[5] Birkhoff, G.D.: Tres Observaciones sobre el Algebra Lineal. Universidad Nacional de Tucuman Revista, Serie A 5, 147–151 (1946)

[6] Agrawal, S., Wang, Z., Ye, Y.: Parimutuel Betting on Permuations. In: Papadimitriou, C., Zhang, S. (eds.) WINE 2008. LNCS, vol. 5385, pp. 126–137. Springer, Heidelberg (2008)

[7] Jerrum, M., Sinclair, A., Vigoda, E.: A Polynomial-time Approximation Algorithm for the Permanent of a Matrix with Nonnegative Entries. Journal of the ACM 51(4), 671–697 (2004)

[8] Nock, R., Nielsen, F.: Fitting Smallest Enclosing Bregman Ball. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 649–656. Springer, Heidelberg (2005)

[9] Tsang, I.W., Kocsor, A., Kwok, J.T.: Simple Core Vector Machines with Enclosing Balls. In: Proc. of ICLM, pp. 911–918 (2007)

[10] Cayton, L.: Fast Nearest Neighbor Retrival with Bregman Divergences. In: Proc. of ICLM, pp. 112–119 (2008)

[11] Slater, P.B.: Maximum-Entropy Representations in Convex Polytopes. Env. and Planning 21, 1541–1546 (1989)

New Partially Labelled Tree Similarity Measure: A Case Study*

David Rizo and José M. Iñesta

Dept. Lenguajes y Sistemas Informáticos, Universidad de Alicante,
E-03080 Alicante, Spain
{drizo,inesta}@dlsi.ua.es

Abstract. Trees are a powerful data structure for representing data for which hierarchical relations can be defined. They have been applied in a number of fields like image analysis, natural language processing, protein structure, or music retrieval, to name a few. Procedures for comparing trees are very relevant in many task where tree representations are involved. The computation of these measures is usually a time consuming tasks and different authors have proposed algorithms that are able to compute them in a reasonable time, through approximated versions of the similarity measure. Other methods require that the trees are fully labelled for the distance to be computed. In this paper, a new measure is presented able to deal with trees labelled only at the leaves, that runs in $O(|T_A| \times |T_B|)$ time. Experiments and comparative results are provided.

Keywords: Tree edit distance, multimedia, music comparison and retrieval.

1 Introduction

The computation of a measure of the similarity between two trees is a subject of interest in very different areas where trees are suitable structures for data coding. Trees are able to code hierarchical relations in their structure in a natural way and they have been utilized in many tasks, like text document analysis [8], protein structure [11], image representation and coding [2], or music retrieval [9], to name just a few.

Different approaches have been proposed in order to perform this comparison. Some of them pose a restriction on how the comparison is performed, other establish valid mappings. While some methods pay more attention to the tree structure, others do it to the content of the nodes and leaves. Most of them are designed to work with fully labelled trees.

The method proposed in this paper is designed to work with partially labelled trees, more precisely with those labelled only at the leaves. This fact, places the focus more on the coded content and the relations within its context. One of the

* This work was funded by the Spanish *DRIMS project* (TIN2009-14247-C02), and the research programme *Consolider Ingenio 2010* (MIPRCV, CSD2007-00018).

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 296–305, 2010.

fields where this situation is relevant is music comparison and retrieval. Trees have been used for this task and a number of representation and comparison schemes have been applied based on tree edit distances [9] or probabilistic similarity schemes [3].

In any case, the computation of these measures is usually a time consuming task and different authors have proposed algorithms that are able to compute them in a reasonable time [12], through approximated versions of the similarity measure. In this paper, a new algorithm is presented, able to deal with trees labelled only at the leaves that runs in $O(|T_A| \times |T_B|)$ time, where $|T_x|$ stand for the number of nodes in tree T_x.

2 Tree Comparison Methods

A number of similarity measures for the ordered non-evolutionary trees have been defined in the literature. Some of them measure the sequence of operations needed to transform one tree in another one, others look for the longest common path from the root to a tree node, and there are methods that allow wildcards in the matching process in the so-called *variable-length doesn't care* (VLDC) distance. Several taxonomies of these measures have been proposed. The interested reader can look up a hierarchy of tree edit distance measures in [7] and [14], and a survey in [1].

2.1 Definitions and Notations

In this section, the terms and notations that will be used in this paper will be defined.

Let $T = (V, E, L)$ be a labelled tree formed by a finite non-empty set V of vertices, a finite set $E \subseteq V \times V$ of arcs, and a set L of labels for nodes. Each node contains a label, possibly empty. The labelling function will be defined by label: $V \rightarrow L$. The empty tree will be denoted as λ, and the empty label as ϵ.

This tree is said to be a *labelled rooted tree* if there is a distinguished node $r \in V$, called the *root* of the tree and denoted by root: $T \rightarrow V$ such that for all nodes $v \in V$, there is an only path from root r to node v. All the trees used in this report are labelled rooted trees, so both terms will be used interchangeably.

The *level* or *depth* of a node $v \in V$, denoted by depth: $V \rightarrow \mathbb{N} \cup \{0\}$, is the length of the unique path from the root node root(T) to node v. The *height* denoted by h: $T \rightarrow \mathbb{N} \cup \{0\}$ is defined as $\mathrm{h}(T) = \max_{v \in V}\{\mathrm{depth}(v)\}$.

Let two nodes $v, w \in V$, v is said to be the parent of w, if $(v, w) \in E$ and $\mathrm{depth}(w) = \mathit{depth}(v) + 1$. The parent of a node will be obtained by the function par: $V \rightarrow V$. The node w is said to be a child of v. Let's define the function children: $V \rightarrow 2^V$ as the set of all children of node v. $\mathrm{children}(v) = \{w | \mathrm{par}(w) = v\}$.

The arity, rank, or outdegree of a node $v \in V$, denoted by rank: $V \rightarrow \mathbb{N} \cup \{0\}$, is the number of children of a node. $\mathrm{rank}(v) = |\mathrm{children}(v)|$. When applied to a tree T, $\mathrm{rank}(T) = \max_{v \in V}\{\mathrm{rank}(v)\}$.

A node $v \in V$ is said to be a *leaf* if $\text{rank}(v) = 0$. A boolean function leaf : $V \to \mathbb{B}$ is defined to denote it. Similarly, leaves(T) is the set of nodes of that tree that have no children: $\{v \in V \,|\, \text{leaf}(v) = \text{true}\}$.

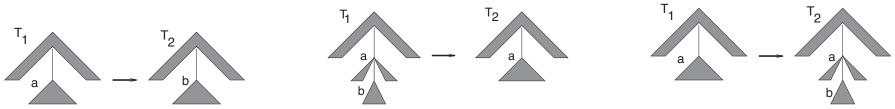
An *ordered tree* is a tree where the relative order of its children is fixed for each node. It allows us to define the function child: $\mathbb{N} \times V \to V$, such that $\text{child}_i(v)$ is just before $\text{child}_j(v)$ iff $i = j - 1$, $\forall i, j \in \mathbb{N}$.

The postorder numbering of a tree consists of giving the visit order of each node of the tree following a postorder traversal of the tree. To uniquely identify nodes in a tree T, lets define $T[i] \in \mathbb{N}$ as the ith node in a postorder numbering, beginning from 1.

A forest is a disjoint union of trees, and an ordered forest has the property that its components follow an order being this way a sequence of trees that will be denoted as $\mathcal{T}^+$. The operation $T[\text{child}_i(T)..\,\text{child}_j(T)]$ is the set composed by the children of T from positions i to j, both included, and it forms a forest. For abbreviating the notation, $T[i..j]$ will be used in the sequel to denote this operation.

Tree Edit Distance. The classical edit distance between two trees is the minimal cost to transform one input tree into an output one by edit operations. An *edit operation* over two trees $T_A = (V, E, L)$ and $T_B = (V', E', L')$ is any of the following:

- *relabel* the label l of a node $v \in V$ by the label l' of another node $w \in V'$, denoted by (v, w) (Fig. 1a).
- *deletion* of a non-root node $v \in V$, denoted by (v, λ), consists of deleting it, making the children of v become the children of par(v), just in the position that was occupied by v, preserving this way the left to right ordering of leaves (Fig. 1b).
- *insertion* of a non-root node $w \in V'$, denoted by denoted by (λ, w). Given a sequence $w_i \cdots w_j$ of subtrees of a common parent w, the insertion of node w' makes those $w_i \cdots w_j$ subtrees children of w', and w' child of w (Fig. 1c).



(a) Substitute operation **(b)** Delete operation **(c)** Insert operation

Fig. 1. Tree edit operations (from [5])

To each operation, a so-called *edit cost* $c_t\colon V \times V' \to \mathbb{R}$ is assigned based on that of the edit cost of the symbols at labels, $c\colon L \times L \to \mathbb{R}$ that depends on the given application. Therefore, $c_t(a, b)$ denotes the cost of applying the edit operation (v, w) where v is an input node and w is an output node. If $v = \lambda$, the operation denotes an insertion, if $w = \lambda$ the operation is a deletion. Note that the operation (λ, λ) is not allowed.

Definition 1. *An edit script* $e_t = e_{t_1} \cdots e_{t_n}$ *is a sequence of edit operations* $e_{t_i} = (a_i, b_i) \in (V \cup \{\lambda\}) \times (V \cup \{\lambda\})$ *allowing the transformation of a tree* X *into a tree* Y*. The cost of an edit script* $\pi_t(e_t)$ *is the sum of the costs of the edit operations involved in the script:* $\pi(e_t) = \sum_{i=1}^{n} c_t(e_{t_i})$.

Definition 2. *Let* $S_t(X, Y)$ *be the set of all the scripts that enable the emission of* Y *given* X*, the edit distance between* X *and* Y *is defined by:* $d_t(X, Y) = min_{e_t \in S_t(X,Y)} \pi(e_t)$.

2.2 Review of Tree Edit Distances

The first author to give a solution to the general tree edit problem was Tai [13], proposing an algorithm with time complexity $O(|T_A| \times |T_B| \times \text{depth}(T_A)^2 \times \text{depth}(T_B)^2)$, where $|T_i|$ denotes the number of nodes in tree T_i. This algorithm was improved by Shasha and Zhang [15] giving a dynamic programming algorithm with time complexity $O(|T_A| \times |T_B| \times \min(\text{depth}(T_A), |\text{leaves}(T_A)|) \times \min(\text{depth}(T_B), |\text{leaves}(T_B)|))$.

The alignment distance is a restricted version of the edit distance based on forcing the application of all insertions before any deletions. Hence, the edit distance is always lower or equal than the optimal alignment [1]. It seems that alignment charges more for the structural dissimilarity at the top levels of the trees than at the lower levels, whereas edit treats all the levels the same [6]. In that work, an algorithm to solve the problem in $O(|T_A| \times |T_B| \times (\text{rank}(T_A) + \text{rank}(T_B))^2)$ time and $O(|T_A| \times |T_B| \times (\text{rank}(T_A) + \text{rank}(T_B)))$ space is given.

Another interesting variant of the edit distance was introduced by Selkow [12], where deletions and insertions are constrained to leaves. Thus, in order to delete an inner node, all its descendants must be deleted before. This algorithm has its strength in its low temporal cost $O(|T_A| \times |T_B|)$.

Finally, the bottom-up distance between two non-empty rooted trees T_A and T_B and is equal to $1 - f / \max(|T_A|, |T_B|)$, where f is the size of a largest common forest of T_A and T_B [14] . Valiente [14] reported a time complexity $O(|T_A| + |T_B|)$. However, this complexity is actually $O(|T_A| \times |T_B| \times \log(T_A + T_B))$, because in the original paper the computing of the bottom-up mapping is not included in the complexity calculation[1].

2.3 Proposed Partially Labelled Tree Comparison Algorithm

The presented similarity measures between trees are designed to work with fully labelled trees. In order to apply those algorithms to trees labelled only at leaves, the non-labelled inner nodes can be assigned a special label "empty". However it is expected that they don't work as well as they do with fully labelled trees.

[1] The nested loop in the mapping function (lines from 3 to 12 of the algorithm included in [14]) that traverses in level-order all the nodes of both trees leads to $O(|T_A| \times |T_B| \times \log(T_A + T_B))$, where the logarithm corresponds to the map operations on a $(|T_A| + |T_B|)$ size map inside the double loop.

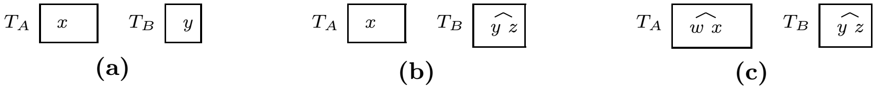


Fig. 2. Similarity function s_p representative cases

In order to overcome this situation two approaches are possible. The first one consists of labelling all nodes using any bottom-up propagation scheme based on the application domain specific knowledge . The main drawback to that option is that any intermediate process might add noise to the resulting trees. The second approach is the definition of a similarity function designed just to compare those partially labelled trees.

The *partially labelled tree comparison algorithm* s_p is based on the assumption that the similarity value between a labelled leaf and a non-labelled inner node should be the average of chances of finding that leaf in the descendants of that inner node. Fig. 2a shows the simplest case of having two leaf trees: $s_p(T_A, T_B) = \delta(x, y)$, where $\delta(x, y) = 1 \iff x = y$, and 0 elsewhere. For comparing the trees shown in Fig. 2b, the chances of finding the label x in T_B are computed as $s_p(T_A, T_B) = (\delta(x, y) + \delta(x, z))/2$. If instead of being a label, y were another tree, the function should be computed recursively. Finally, when none of the trees is composed by a single leaf (Fig. 2c), the similarity of the ordered forests wx and yz can be computed like an edit distance between sequences wx and yz where each symbol is a tree.

This similarity method omits the accounting of the insertion or deletion of nodes and just measures the chance of finding coincident labels, giving more importance to the information hierarchically contained in the tree than to the tree structure.

Being designed for working with partially labelled trees, however, we can slightly adapt the original idea to work also with fully labelled trees. The case of comparing a leaf to a non-leaf tree (Fig. 3a), is computed as $s_p(T_A, T_B) = (\delta(x, b) + \delta(x, y) + \delta(x, z))/3$. Likewise, the similarity $s_p(T_A, T_B)$ between two fully labelled trees (Fig. 3b) is computed as the edit distance between sequences wx and yz, where each symbol is a tree, plus the similarity between labels a and b.

Let $s_p \colon T \times T \to \mathbb{R}$ be a similarity function between trees and $sf_p \colon \mathcal{T}^+ \times \mathcal{T}^+$ be a similarity function between forests. Let us also use rlabel: $T \to L$ that returns the label of the root of the tree, and R_m as an abbreviation for rank(T_m). The similarity between two trees, fully or partially labelled, is defined as:

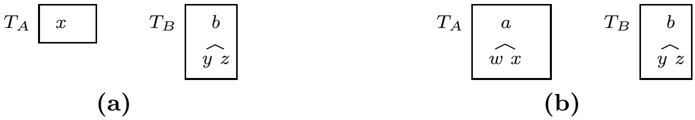


Fig. 3. Similarity function s_p working on fully labelled trees

Definition 3

$$(i) \quad s_p(T_A, T_B) = \begin{cases} \delta(\mathrm{rlabel}(T_A), \mathrm{rlabel}(T_B)) & : \textit{if } \mathrm{leaf}(T_A) \wedge \mathrm{leaf}(T_B) \quad (1) \\ \frac{\delta(\mathrm{rlabel}(T_A),\mathrm{rlabel}(T_B)) + \sum_{j=1}^{R_B} s_p(T_A, \mathrm{child}_j(T_B))}{1+R_B} & : \textit{if } \mathrm{leaf}(T_A) \wedge \neg\,\mathrm{leaf}(T_B) \quad (2) \\ \frac{\delta(\mathrm{rlabel}(T_A),\mathrm{rlabel}(T_B)) + \sum_{i=1}^{R_A} s_p(\mathrm{child}_i(T_A), T_B)}{1+R_A} & : \textit{if } \neg\,\mathrm{leaf}(T_A) \wedge \mathrm{leaf}(T_B) \quad (3) \\ \frac{\delta(\mathrm{rlabel}(T_A),\mathrm{rlabel}(T_B)) + sf_p(T_A, T_B)}{\max(R_A, R_B)+1} & : \textit{otherwise} \quad (4) \end{cases}$$

$$(ii) \quad sf_p(\lambda, \lambda) = 0$$

$$(iii) \quad sf_p(i..i', \lambda) = sf_p(i..i'-1, \lambda)$$

$$(iv) \quad sf_p(\lambda, j..j') = sf_p(\lambda, j..j'-1)$$

$$(v) \quad sf_p(i..i', j..j') = \max \begin{cases} sf_p(i..i'-1, j..j') \\ sf_p(i..i', j..j'-1) \\ sf_p(i..i'-1, j..j'-1) + s_p(T_A[i'], T_B[j']) \end{cases}$$

The simplest situation in Fig. 2a is solved by case (i)-(1). Cases (i)-(2) and (i)-(3) solve the problems depicted in Fig. 2b and 3a. Finally, (i)-(4) computes the similarity for Fig. 3b. After comparing the roots, the ordered forests composed by the tree children (Fig. 2c) are compared with the similarity function between forests sf_p in the the indirect recurrence (ii) to (v).

Complexity of the partial edit distance. In order to calculate the time complexity of s_p and sf_p, the functions $\mathcal{T}_s$ and $\mathcal{T}_{sf}$ will be used respectively. In both cases the size of the problem is the number of nodes of the compared trees: $(|T_A|, |T_B|)$.

$$\mathcal{T}_s(|T_A|, |T_B|) = \begin{cases} 1 & : \quad \textit{if } |T_A| = 1 \wedge |T_B| = 1 \\ R_B \times \mathcal{T}_s(|T_A|, |T_B| / \mathrm{rank}(T_B)) & : \quad \textit{if } |T_A| = 1 \wedge |T_B| > 1 \\ R_A \times \mathcal{T}_s(|T_A| / \mathrm{rank}(T_A), |T_B|) & : \quad \textit{if } |T_A| > 1 \wedge |T_B| = 1 \\ \mathcal{T}_{sf}(|T_A|, |T_B|) & : \quad \textit{if } |T_A| > 1 \wedge |T_B| > 1 \end{cases}$$

The function sf_p can be solved using a dynamic programming scheme as the used for any edit distance. On a classical edit distance, where the substitution cost has constant complexity, given the problem size $(|T_A|, |T_B|)$, the complexity is $O(|T_A| \times |T_B|)$ because its implementation is a simple double loop traversing a $|T_A| \times |T_B|$ matrix. However, in our case, in each step of that iteration, the s_p is called. Under these assumptions, the temporal complexity of the algorithm is obtained:

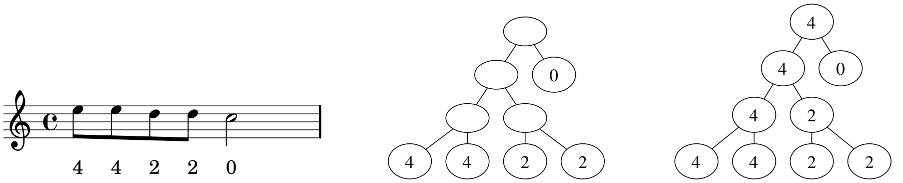
$$\mathcal{T}_s(|T_A|,|T_B|) = c + \mathcal{T}_{sf}(|T_A|-1,|T_B|-1)$$
$$\mathcal{T}_{sf}(|T_A|,|T_B|) = \sum_{i=1}^{R_A}\sum_{j=1}^{R_B}\mathcal{T}_s(\frac{|T_A|}{R_A},\frac{|T_B|}{R_B}) = R_A \times R_B \times \mathcal{T}_s(\frac{|T_A|}{R_A},\frac{|T_B|}{R_B})$$

and it can be shown that this time is $O(|T_A| \times |T_B|)$.

3 Experiments

The experiments are devised to assess the suitability of the proposed algorithm working with both partially and fully labelled tree corpora, and compare it with classical tree comparison algorithms.

In the selected case study, the main goal is to identify a melody from a set of all the different variations played by the musicians. In our experiments, we use tree representations of monophonic music pieces [10] (Fig. 4c). The node labels are symbols from a pitch description alphabet Σ_p. In this paper, the interval modulo 12 from the tonic of the song is utilized as pitch descriptor (Fig. 4a): $\Sigma_p = \{p \mid 0 \leq p \leq 11\} \cup \{-1\}$, where -1 is used to encode rests. For measuring the similarity between the melody and each of the variations, the different tree comparison algorithms have been used.



(a) The figures below the score indicate the Σ_p (interval from tonic that is C in this case).

(b) Corresponding tree representation.

(c) Corresponding propagated tree.

Fig. 4. A short melody sample and its representation as a tree

Initially, only leaf nodes are labelled. Then, in order to reduce tree sizes and improve performing times, a pruning operation at level $\mathscr{L}$ can be applied that propagates bottom-up leaf labels until getting all leaves at level $\mathscr{L}$ or less labelled. For deciding which label to propagate, a melodic analysis [4] is applied to decide which notes are more important and then are more suitable to be promoted. Optionally, if we want to label all nodes in the tree, the same propagation method must be followed until reaching the root.

Corpora. In our experiments, we used a corpus consisting of a set of 420 monophonic 8-12 bar incipits of 20 worldwide well known tunes of different musical genres[2]. For each song, a canonic version was created by writing the score in a musical

[2] The MIDI data set is available upon request to the authors.

notation application and exported to MIDI and MP3 format. The MP3 files were given to three amateur and two professional musicians who listened to the songs (mainly to identify the part of the tune to be played) and played with MIDI controllers the same tune several times with different embellishments and capturing performance errors. This way, for each of the 20 original scores, 20 different variations have been built.

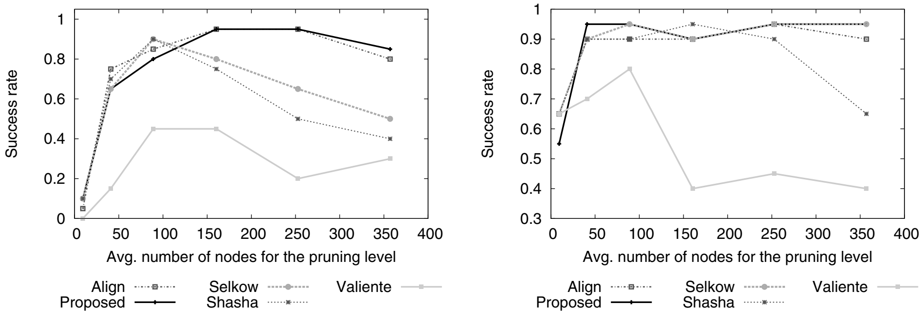
Melody classification accuracy. The experiments have been performed using a query / candidates scheme, i.e., given a corpus, for each song the query prototype is compared to all the scores in the dataset. The similarity values are considered as distances and, following a nearest neighbor (NN) rule, the class of the file closest to the query will be taken as the retrieved song. This answer is correct if it corresponds to the same song of the query. Any other situation is considered as an error.

The success rate is measured as the ratio of correct answers to all the queries. Running times are measured in milliseconds taking into account only the test phase, leaving aside the construction of the representations that may be done off-line. All experiments have been performed using a MacBook Pro machine with 4 Gb RAM and 2 Intel(R) Core 2 Duo(R) CPU running at 2.26GHz, with a Java virtual machine 1.6.

Results. The experiments have been performed feeding all algorithms with several versions of the trees that have been pruned from level $\mathscr{L} = 1$ to $\mathscr{L} = 6$ (which is the maximum depth found in the corpus). For each pruning level the average number of nodes has been extracted to be used as x-axis in the plots.

The results plotted in Fig. 5a show that the proposed algorithm behaves the best for not-pruned trees or $\mathscr{L} = 6$ (see results for $x > 350$). For pruned trees, it behaves also the best in average.

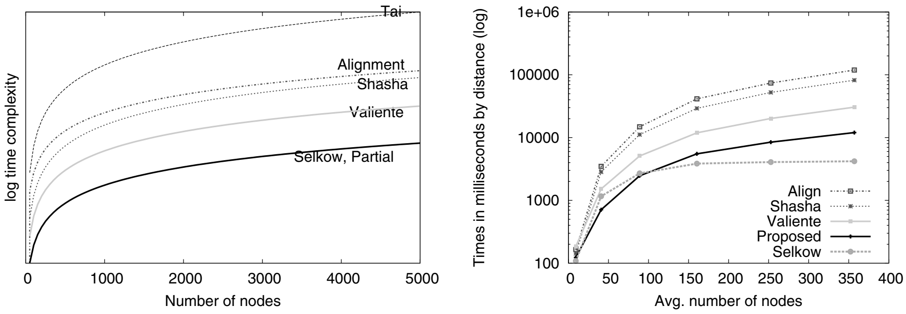
When working with fully labelled trees (see Fig. 5b), the success rates of the proposed method are comparable to the success rates of the Selkow and the Alignment distance. The plot in Fig. 6a shows the theoretical evolution of computing



(a) Partially labelled trees

(b) Fully labelled trees

Fig. 5. Success rates of proposed method compared to classical tree edit distances



(a) Complexities theoretical processing times

(b) Actual processing times

Fig. 6. Time processing evolution of algorithm

times of the main tree edit distance algorithms given their time complexity, and Fig. 6b describes the actual processing times of those distances in our experiments. The actual times confirm the prediction from the theoretical complexities, being our proposed algorithm the second faster one.

Thus, it seems that the proposed similarity is suitable for its purpose, being able to compare successfully both trees labelled only at leaves and fully labelled trees better than the other methods in terms of trade-off between time and success rate.

4 Conclusions

In this paper a new similarity tree algorithm has been introduced for working with trees labelled only at the leaves. It has been applied to a real application: the similarity computation between monophonic symbolic music pieces encoded with both partially and fully labelled trees. The application has been tested using the proposed algorithm and classical tree similarity algorithms from the literature. From the results, it seems that the proposed algorithm outperforms the other ones in a trade-off among computing time and success rates.

In the future the algorithm must be applied to other applications that require this kind of measure to compare trees. Currently we are applying the same methodology to bigger corpora encoded with different pitch encodings and other propagation schemes, being the partial results obtained as good as those shown in this paper.

References

1. Bille, P.: A survey on tree edit distance and related problems. Theorical Computer Science 337(1-3), 217–239 (2005)
2. Finkel, R.A., Bentley, J.L.: Quad trees: A data structure for retrieval on composite keys. Acta Inf. 4, 1–9 (1974)

3. Habrard, A., Iñesta, J.M., Rizo, D., Sebban, M.: Melody recognition with learned edit distances. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 86–96. Springer, Heidelberg (2008)
4. Illescas, P.R., Rizo, D., Iñesta, J.M.: Harmonic, melodic, and functional automatic analysis. In: Proceedings of the 2007 International Computer Music Conferrence, vol. I, pp. 165–168 (2007)
5. Isert, C.: The editing distance between trees. Technical report, Institut für Informatik, Technische Universität München (1999)
6. Jiang, T., Wang, L., Zhang, K.: Alignment of trees – an alternative to tree edit. Theoretical Computer Science 143(1), 137–148 (1995)
7. Kuboyama, T., Shin, K., Miyahara, T.: A hierarchy of tree edit distance measures. Theoretical Computer Science and its Applications (2005)
8. Marcus, M., Kim, G., Marcinkiewicz, M.A., Macintyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B.: The penn treebank: Annotating predicate argument structure. In: ARPA Human Language Technology Workshop, pp. 114–119 (1994)
9. Rizo, D., Lemström, K., Iñesta, J.M.: Tree representation in combined polyphonic music comparison. In: Ystad, S., Kronland-Martinet, R., Jensen, K. (eds.) CMMR 2008. LNCS, vol. 5493, pp. 177–195. Springer, Heidelberg (2009)
10. Rizo, D., Moreno-Seco, F., JIñesta, J.M.: Tree-structured representation of musical information. In: Perales, F.J., Campilho, A.C., Pérez, N., Sanfeliu, A. (eds.) IbPRIA 2003. LNCS (LNAI), vol. 2652, pp. 838–846. Springer, Heidelberg (2003)
11. Russell, R.B., Barton, G.J.: Multiple protein sequence alignment from tertiary structure comparison: Assignment of global and residue confidence levels. Proteins: Structure, Function, and Bioinformatics 14, 309–323 (2004)
12. Selkow, S.M.: The tree-to-tree editing problem. Information Processing Letters 6(6), 184–186 (1977)
13. Tai, K.-C.: The tree-to-tree correction problem. J. ACM 26(3), 422–433 (1979)
14. Valiente, G.: An efficient bottom-up distance between trees. International Symposium on String Processing and Information Retrieval, 0,212 (2001)
15. Zhang, K., Shasha, D.: Simple fast algorithms for the editing distance between trees and related problems. SIAM J. Comput. 18(6), 1245–1262 (1989)

Complete Search Space Exploration for SITG Inside Probability

Guillem Gascó, Joan-Andreu Sánchez, and José-Miguel Benedí

Institut Tecnològic d'Informàtica, Universitat Politècnica de València
Camí de Vera s/n, València, 46022, Spain
ggasco@iti.upv.es, {jandreu,jbenedi}@dsic.upv.es

Abstract. Stochastic Inversion Transduction Grammars are a very powerful formalism in Machine Translation that allow to parse a string pair with efficient Dynamic Programming algorithms. The usual parsing algorithms that have been previously defined cannot explore the complete search space. In this work, we propose important modifications that consider the whole search space. We formally prove the correctness of the new algorithm. Experimental work shows important improvements in the probabilistic estimation of the models when using the new algorithm.

1 Introduction

Stochastic Inversion Transduction Grammars (SITGs) were introduced in [1] for describing structurally correlated pairs of languages. SITGs can be used to simultaneously analyze two strings from different languages and to correlate them. SITGs have been used in the last few years for Machine Translation (MT), especially for pairs of languages that are sufficiently non-monotonic. Several works have explored its use for MT [2,3,4,5].

An efficient Dynamic Programming parsing algorithm for SITGs was presented in [2]. This algorithm is similar to the CKY algorithm for Probabilistic Context Free Grammars. The parsing algorithm does not allow the association of two items that have the empty string in one of their sides. This limitation restricts the search space,and thus, it prevents exploring some valid parse trees. Expressiveness capacity of SITGs by using Wu's parsing algorithm has been recently studied in [6,7].

In this paper, we propose a new version of the *Inside* parsing algorithm for SITGs that allows to consider all valid parse trees. Then, we also present the formal proof of the correctness, and a set of experiments to demonstrate the usefulness of the new valid parse trees.

2 Inside Probability with SITG

A SITG in Chomsky Normal Form [2] can be defined as a set of lexical rules that are noted as $A \rightarrow a/\epsilon$, $A \rightarrow \epsilon/b$, $A \rightarrow a/b$; direct syntactic rules that are noted as $A \rightarrow [BC]$; and inverse syntactic rules that are noted as $A \rightarrow$

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 306–315, 2010.

$\langle BC\rangle$, where A, B, C are non-terminal symbols, a, b are terminal symbols, ϵ is the empty string, and each rule has a probability value p attached. The sum of the probabilities of the rules with the same non-terminal in the left side must be equal to 1. When a direct syntactic rule is used in parsing, both strings are parsed with the syntactic rule $A \rightarrow BC$. When an inverse rule is used in parsing, one string is parsed with the syntactic rule $A \rightarrow BC$, and the other string is parsed with the syntactic rule $A \rightarrow CB$.

The *inside* probability of a substring pair $(x_{i+1} \ldots x_{i+j}, y_{k+1} \ldots y_{k+l})$ from the non-terminal symbol A is defined as follows:

$$\mathcal{E}_{i,i+j,k,k+l}[A] = p(A \overset{*}{\Rightarrow} x_{i+1} \cdots x_{i+j}/y_{k+1} \cdots y_{k+l}) \ , \tag{1}$$

where j and l represent the size of the subproblems. In this way, the probability of the string pair $(x_1 \ldots x_{|x|}, y_1 \ldots y_{|y|})$ is $\mathcal{E}_{0,|x|,0,|y|}[S]$.

Let $\mathcal{G}$ be a SITG, and let $(x_1 \ldots x_{|x|}, y_1 \ldots y_{|y|})$ be a string pair. In general, we can efficiently calculate the probability of this pair by means of a simple modification of the well-known CKY-based *inside* algorithm [8,2]. This algorithm is essentially a Dynamic Programming method, which is based on the construction of a triangular $(n+1) \times (n+1)$ probabilistic parse matrix $\mathcal{E}$. Following a notation very close to [2], each element of $\mathcal{E}$ is a probabilistic nonterminal vector, where their components are computed for all $A \in N$ as:

1. Initialization

$$\mathcal{E}_{i,i+1,k,k+1}[A] = p(A \rightarrow x_{i+1}/y_{k+1}) \qquad 0 \leq i < |x| \ 0 \leq k < |y| \tag{2}$$

$$\mathcal{E}_{i,i+1,k,k}[A] = p(A \rightarrow x_{i+1}/\epsilon) \qquad 0 \leq i < |x| \ 0 \leq k \leq |y| \tag{3}$$

$$\mathcal{E}_{i,i,k,k+1}[A] = p(A \rightarrow \epsilon/y_{k+1}) \qquad 0 \leq i < |x| \ 0 \leq k \leq |y| \tag{4}$$

2. Recursion

$$\text{For all } A \in N \text{ and } i, j, k, l \text{ such that} \begin{cases} 0 \leq i \leq |x|, \ 0 \leq j \leq |x| - i \\ 0 \leq k \leq |y|, \ 0 \leq l \leq |y| - k \\ j + l \geq 2, \end{cases} \tag{5}$$

$$\mathcal{E}_{i,i+j,k,k+l}[A] = \mathcal{E}^{[]}_{i,i+j,k,k+l}[A] + \mathcal{E}^{\langle\rangle}_{i,i+j,k,k+l}[A]$$

where

$$\mathcal{E}^{[]}_{i,i+j,k,k+l}[A] = \sum_{\substack{B,C \in N \\ 1 \leq I \leq j, \ 1 \leq K \leq l \\ ((j-I)+(l-K)) \times (I+K) \neq 0}} p(A \rightarrow [BC]) \ \mathcal{E}_{i,i+I,k,k+K}[B] \ \mathcal{E}_{i+I,i+j,k+K,k+l}[C] \tag{6}$$

$$\mathcal{E}^{\langle\rangle}_{i,i+j,k,k+l}[A] = \sum_{\substack{B,C \in N \\ 1 \leq I \leq j, \ 1 \leq K \leq l \\ ((j-I)K) \times (I+(l-K)) \neq 0}} p(A \rightarrow \langle BC\rangle) \ \mathcal{E}_{i,i+I,k+K,k+l}[B] \ \mathcal{E}_{i+I,i+j,k,k+K}[C] \tag{7}$$

The main differences of this algorithm with regard to the Wu's original algorithm are:

- The restriction $j + l \geq 2$ in (5) substitutes the restriction $j + l > 2$ in Wu's algorithm, and
- The restrictions $((j - I) + (l - K)) \times (I + K) \neq 0$ in (6) and $((j - I) + K) \times (I + (l - K)) \neq 0$ in (7) substitute the restriction $I(j - I) + K(l - K) \neq 0$ in Wu's algorithm.

These modifications allow us to consider some parse trees that the original algorithm ignore. Thus, for example, consider a SITG composed by the following rules (the probabilities of the rules have been omited): $(S \rightarrow [SS], S \rightarrow \langle SS \rangle, S \rightarrow \epsilon/b, S \rightarrow a/\epsilon, S \rightarrow a/b)$. If the string pair is (a, b), this SITG could parse this string pair with the parse trees that can be seen in Fig. 1.

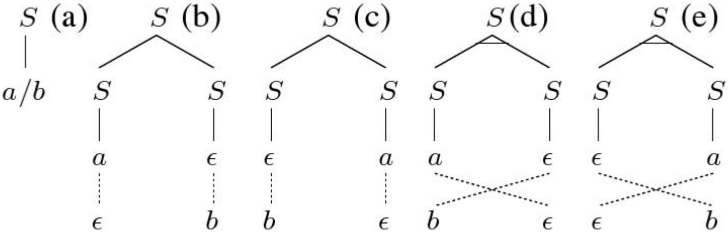


Fig. 1. Parse trees for input pair (a, b) that are taken into account in the search process with the modifications

However, the original algorithm would use just parse tree (a) of Fig. 1. The original algorithm is not able to obtain parse trees (b-e) due to the restriction $j + l > 2$. This restriction does not allow the algorithm to consider subproblems in which each substring has length 1 which have not been previously considered in the initialization step.

In fact, this situation appears for other string pairs (see Fig. 2) in which a string in one side is associated with the empty string in the other side through rules that are not lexical rules. For example, in Fig. 2b, substring aa could be associated with ϵ. However, this parse tree cannot be considered with the original algorithm due to the search restrictions that it applied.

Although the modifications to the algorithm allow it to explore more parse trees, the time complexity is the same as in the original algorithm: $O(N^3|x|^3|y|^3)$ where N is the number of non-terminal symbols, $|x|$ is the length of the source language string, and $|y|$ is the length of the target language string.

In order to prove the correctness of the modified algorithm we show that the *inside* probability of a bilingual string computed using it is the correct and complete probability of the string.

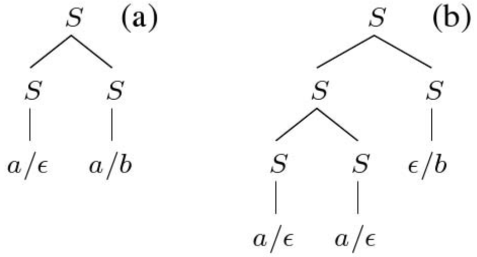


Fig. 2. Parse tree (a) can be obtained with Wu's algorithm for $aa\#b$, but parse tree (b) was not considered

Theorem 1. *If the* inside *algorithm is applied to the string pair* $(x_1 \ldots x_{|x|}, y_1 \ldots y_{|y|})$ *with a SITG* $\mathcal{G}$*, then the probabilistic parse matrix* $\mathcal{E}$ *collects correctly the probability of this string pair.*

`Proof`

Let $\mathcal{G}$ be a SITG, $p(A \overset{+}{\Rightarrow} x_{i+1} \ldots x_{i+j}/y_{k+1} \ldots y_{k+l})$ is the inside probability of the substring pair $(x_{i+1} \ldots x_{i+j}, y_{k+1} \ldots y_{k+l})$.

If the size of subproblems is equal to 1, we can consider the following cases:

- If $j = 1$ and $l = 0$ then, by (3), we have:

$$p(A \overset{+}{\Rightarrow} x_{i+1}/\epsilon) = p(A \rightarrow x_{i+1}/\epsilon) = \mathcal{E}_{i,i+1,k,k}[A]$$

  with $0 \leq i < |x|$, $0 \leq k < |y|$.
- If $j = 0$ and $l = 1$ then, by (4), we have:

$$p(A \overset{+}{\Rightarrow} \epsilon/y_{k+1}) = p(A \rightarrow \epsilon/y_{k+1}) = \mathcal{E}_{i,i,k,k+1}[A]$$

  with $0 \leq i < |x|$, $0 \leq k < |y|$.
- If $j = 1$ and $l = 1$ then, the probability of the substring pair (x_{i+1}, y_{i+1}) is computed with the following possibilities, as illustrated in Fig. 1:

$$\begin{aligned}
p(A \overset{+}{\Rightarrow} x_{i+1}/y_{k+1}) = {} & p(A \rightarrow x_{i+1}/y_{k+1}) \\
& + \sum_{B,C} p(A \rightarrow [BC]) p(B \overset{+}{\Rightarrow} x_{i+1}/\epsilon) p(C \overset{+}{\Rightarrow} \epsilon/y_{k+1}) \\
& + \sum_{B,C} p(A \rightarrow [BC]) p(B \overset{+}{\Rightarrow} \epsilon/y_{k+1}) p(C \overset{+}{\Rightarrow} x_{i+1}/\epsilon) \\
& + \sum_{B,C} p(A \rightarrow \langle BC \rangle) p(B \overset{+}{\Rightarrow} x_{i+1}/\epsilon) p(C \overset{+}{\Rightarrow} \epsilon/y_{k+1}) \\
& + \sum_{B,C} p(A \rightarrow \langle BC \rangle) p(B \overset{+}{\Rightarrow} \epsilon/y_{k+1}) p(C \overset{+}{\Rightarrow} x_{i+1}/\epsilon)
\end{aligned}$$

with $0 \leq i < |x|$, $0 \leq k < |y|$. Considering the expressions (2), (3) and (4), we have:

$$\begin{aligned} p(A \overset{+}{\Rightarrow} x_{i+1}/y_{k+1}) = & \mathcal{E}_{i,i+1,k,k+1}[A] \\ & + \sum_{B,C} p(A \rightarrow [BC])\mathcal{E}_{i,i+1,k,k}[B]\mathcal{E}_{i,i,k,k+1}[C] \\ & + \sum_{B,C} p(A \rightarrow [BC])\mathcal{E}_{i,i,k,k+1}[B]\mathcal{E}_{i,i+1,k,k}[C] \\ & + \sum_{B,C} p(A \rightarrow \langle BC\rangle)\mathcal{E}_{i,i+1,k,k}[B]\mathcal{E}_{i,i,k,k+1}[C] \\ & + \sum_{B,C} p(A \rightarrow \langle BC\rangle)\mathcal{E}_{i,i,k,k+1}[B]\mathcal{E}_{i,i+1,k,k}[C] \end{aligned}$$

It is important to note that in Wu's version [2], only the first term is possible, since the rest of the terms are prohibited because it imposes the restriction $j + l > 2$.

Furthermore, in our case, the last 4 terms correspond to the general term of the algorithm for $j = 1, l = 0$ and $j = 0, l = 1$ both for the direct rules and the inverse rules.

- Finally, the possibility $j = 0$ and $l = 0$ is excluded given that there are no rules like $p(A \rightarrow \epsilon/\epsilon)$ in the model.

For subproblems of size greater than 1, and in a similar way than in [2], the probability of $p(A \overset{+}{\Rightarrow} x_{i+1} \dots x_{i+j}/y_{k+1} \dots y_{k+l})$ can be solved considering rules (both direct and inverse) and a cutoff points as follows:

$$\begin{aligned} & p(A \overset{+}{\Rightarrow} x_{i+1} \dots x_{i+j}/y_{k+1} \dots y_{k+l}) = \\ & \sum_{B,C} p(A \rightarrow [BC]) \\ & \sum_{\substack{1 \leq I \leq j \\ 1 \leq K \leq l}} p(B \overset{+}{\Rightarrow} x_{i+1} \dots x_{i+I}/y_{k+1} \dots y_{k+K}) p(C \overset{+}{\Rightarrow} x_{i+I+1} \dots x_{i+j}/y_{k+K+1} \dots y_{k+l}) \\ & + \sum_{B,C} p(A \rightarrow \langle BC\rangle) \\ & \sum_{\substack{1 \leq I \leq j \\ 1 \leq K \leq l}} p(B \overset{+}{\Rightarrow} x_{i+1} \dots x_{i+I+1}/y_{k+K+1} \dots y_{k+l}) p(C \overset{+}{\Rightarrow} x_{i+I+1} \dots x_{i+j}/y_{k+1} \dots y_{k+K}) \end{aligned}$$

With $0 \leq i \leq |x|$, $0 \leq j \leq |x| - i$, $0 \leq k \leq |y|$, $0 \leq l \leq |y| - k$. Considering the definition (1) and the general term of the algorithm, the previous expression can be rewritten as:

$$
\begin{aligned}
&p(A \overset{+}{\Rightarrow} x_{i+1} \dots x_{i+j}/y_{k+1} \dots y_{k+l}) = \\
&\sum_{B,C} p(A \to [BC]) \sum_{\substack{1 \le I \le j \\ 1 \le K \le l}} \mathcal{E}_{i,i+I,k,k+K}[B]\mathcal{E}_{i+I,i+j,k+K,k+l}[C] \\
&+ \sum_{B,C} p(A \to \langle BC \rangle) \sum_{\substack{1 \le I \le j \\ 1 \le K \le l}} \mathcal{E}_{i,i+I,k+K,k+l}[B]\mathcal{E}_{i+I,i+j,k,k+K}[C] \\
&= \mathcal{E}_{i,i+j,k,k+l}[A] \qquad \square
\end{aligned}
$$

Corollary 1. *The probability of the pair string* $(x_1 \dots x_{|x|}, y_1 \dots y_{|y|})$ *can be computed by means of the probabilistic parse matrix* $\mathcal{E}$ *in the following terms:*

$$p(S \overset{+}{\Rightarrow} x_1 \dots x_{|x|}/y_1 \dots y_{|y|}) = \mathcal{E}_{0,|x|,0,|y|}[S] \qquad \square$$

3 Experiments

In this section we present several experiments in order to show the performance of the modified SITG parsing algorithm and compare it to the original algorithm. To stress the differences between both algorithms we used the Viterbi parsing algorithm instead of the Inside algorithm. However, similar assumptions can be done for the inside algorithm. Viterbi parsing algorithm computes the most likely parse tree for a given string pair and its probability. The modified version of the Viterbi parsing algorithm [7] can be obtained using maximizations instead of sums in the expressions that have been explained in the previous section.

Experiments were carried out over two different bilingual corpora, the Chinese-English BTEC part of the IWSLT2009 and the French-English Hansard Corpus. The former is a small corpus and we used it to test the differences of both Viterbi parsing algorithms with two languages with a very distinct syntax structure. The later is a larger corpus and is used to test the impact of the modified algorithm for languages with a similar syntactic structure and for large corpora. We explored also the impact of the use of bracketing information in both algorithms. For that purpose we used a parsing strategy similar to the one used in [3], for the corpus partially or fully bracketed. It must be noted that the bracketing information restricts the search to only those parse trees that are consistent with the bracketing. In order to obtain the bracketing information for the corpora, we used several language versions (Chinese, French and English) of the Berkeley Parser [9].

3.1 IWSLT 2009 Corpus

The BTEC part of the IWSLT 2009 corpus [10] is a set of parallel travel sentences in Chinese and English. For the experiments of this work we used the training and the test partitions. Table 1 shows the statistics of this corpus.

As previously mentioned, the original algorithm does not explore all the possible parse trees for a given sentence. In some cases, the algorithm misses the parse tree with the highest probability. As proved in Section 2, the modified algorithm explores the whole search space and, thus, it finds always the most probable tree. In this experiment, we computed the percentage of sentences, for which the parse tree obtained with the modified algorithm have a higher probability than the one obtained with the original algorithm[1]. In other words, the number of times the original algorithm could not explore the best tree. For this purpose, we used a SITG obtained following the method explained in [3]. In addition, for some sentences the original algorithm could not find any parse tree while the modified could.

Table 1. Statistics for IWSLT 2009 Chinese-English BTEC corpus

Corpus Set	Statistic	Chinese	English
	Sentences	42,655	
Training	Words	330,163	380,431
	Vocabulary Size	8,773	8,387
	Sentences	511	
Test	Words	3,352	3,821
	Vocabulary Size	888	813

Table 2 shows the results of the experiment for the non-bracketed corpus (Ch-En), the corpus with only the Chinese side bracketed ([Ch]-En), the corpus with only the English side bracketed (Ch-[En]) and the corpus bracketed in both sides ([Ch]-[En]).

Table 2. Percentage of sentences in IWSLT Corpus for which the original algorithm does not find the parse tree with the highest probability and percentage of sentences not parsed by the original algorithm

Experiment	% of sentences with a different parse tree	% of sentences not parsed with the original algorithm
Ch - En	36.25%	0.24%
[Ch] - En	37.21%	1.4%
Ch - [En]	36.97%	1.02%
[Ch] - [En]	40.93%	3.92%

It must be noted that there was a high percentage of sentences for which the original algorithm could not find the tree with the highest probability. This percentage was even higher when we used bracketing information. The percentage of sentences that could not be parsed with the original algorithm using bracketing in both sides was almost 4%.

[1] Note that the contrary is not possible.

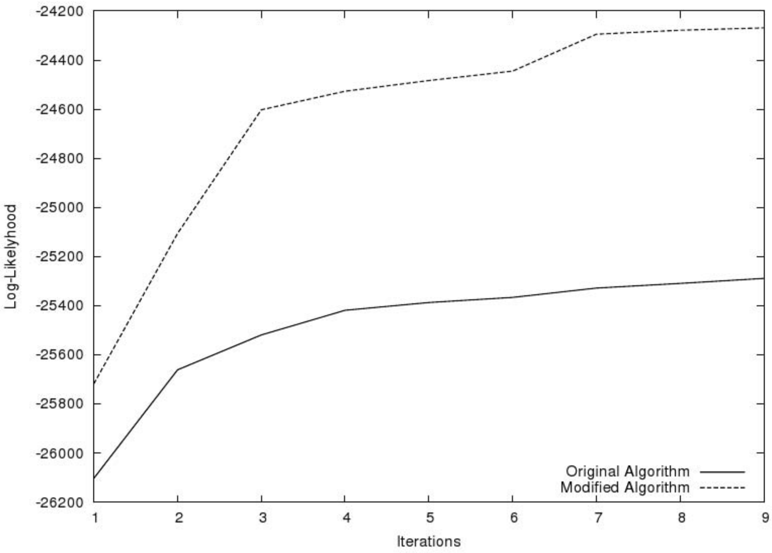


Fig. 3. Log-likelihood of the SITG for several iterations of the Viterbi reestimation using both algorithms on the IWSLT test set

For all the iterations, the reestimation with the modified algorithm results in a more adjusted grammar.

The second experiment tried to determine the importance of the differences between the use of the original or the modified algorithm in the process of SITG reestimation. We performed several iterations of the Viterbi reestimation with each algorithm and we then computed the log-likelihood for the SITG resulting in each iteration over the test set; that is, the logarithm of the product of the probabilities of the SITG parse trees for all the sentences of the test. It is worth noting that for the computation of the log-likelihood we only used those sentences that could be parsed by both algorithms. Figure 3 shows the log-likelihood for each of the iterations and each of the algorithms.

3.2 Hansard Corpus

The Hansard corpus [11] is a set of parallel texts in English and Canadian French, extracted from official records of the Canadian Parliament. Due to the high computational cost of the SITG parsing algorithms and in order to get a faster process, we only used the sentences of length lower than 40 words in each of the languages. The statistics of the resulting corpus are shown in Table 3.

Table 3. Statistics for Hansard French-English corpus (less than 40 words)

Corpus Set	Statistic	French	English
Training	Sentences	997,823	
	Words	16,547,387	14,266,620
	Vocabulary Size	68,431	49,892

Table 4. Percentage of sentences of the Hansard corpus for which the original algorithm does not find the parse tree with the highest probability

Experiment	% of sentences with a different parse tree
Fr - En	27.73%
[Fr] - En	28.06%
Fr - [En]	28.51%
[Fr] - [En]	30.56%

The experiment performed with this corpus is equivalent to the first one explained in the previous subsection. We computed the percentage of times the original algorithm could not find the tree with the highest probability using or not bracketing information. Table 4 shows the results of the experiment.

Compared to the experiment with the IWSLT Corpus, the percentage of sentences with a different parse tree is lower. This behavior may be due to two factors: the similarity in the syntactic structure of both languages and/or the size of the corpus that allows for a better reestimation of the SITG. However, it is still high, almost one third of the sentences of the corpus. The behavior of the algorithms, in respect with the bracketing information is the same as in the IWSLT corpus: the more restricted the search space is, the more differences have the resulting parse trees.

4 Conclusions

SITGs have proven to be a powerful tool in Syntax Machine Translation. However, the parsing algorithms that have been previously proposed do not explore all the possible parse trees. This work propose a modified parsing algorithm that is able to explore the whole search space. We prooved the completeness of the new search. The experiments carried out over two different corpora show that there is a high percentage of sentences for wich the original algorithm cannot find the tree with the highest probability and, in some cases, it cannot find any parse tree at all. In addition, the use of the modified algorithm for reestimation results in better SITGs. As future work, we plan to study the impact of these modifications on the use of SITGs for Machine Translation and the inside-outside SITG reestimation algorithm.

Acknowledgments

Work supported by the EC (FEDER-FSE), the Spanish Government (MICINN, MITyC, "Plan E", under grants MIPRCV "Consolider Ingenio 2010" CSD2007-00018, iTrans2 TIN2009-14511 and erudito.com TSI-020110-2009-439) and the Generalitat Valenciana (grant Prometeo/2009/014 and BFPI/2007/117).

References

1. Wu, D.: Stochastic inversion transduction grammars with application to segmentation, bracketing, and alignment of parallel corpora. In: Proc. of the 14th International Conference on Artificial Intelligence, Montreal, vol. 2, pp. 1328–1335 (1995)
2. Wu, D.: Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics 23(3), 377–404 (1997)
3. Sánchez, J., Benedí, J.: Stochastic inversion transduction grammars for obtaining word phrases for phrase-based statistical machine translation. In: Proc. of Workshop on Statistical Machine Translation. HLT-NAACL 2006, New York, USA, June 2006, pp. 130–133 (2006)
4. Huang, S., Zhou, B.: An em algorithm for scfg in formal syntax-based translation. In: ICASSP, Taiwan, China, April 2009, pp. 4813–4816 (2009)
5. Gascó, G., Sánchez, J.A.: Syntax augmented inversion transduction grammars for machine translation. In: Gelbukh, A. (ed.) CICLing 2010. LNCS, vol. 6008, pp. 427–437. Springer, Heidelberg (2010)
6. Soggard, A., Wu, D.: Empirical lower bounds on translation unit error rate for the full class of inversion transduction grammars. In: Proc. 11th Internatnional Conference on Parsing Technologies, Paris, October 2009, pp. 33–36 (2009)
7. Gascó, G., Sánchez, J., Benedí, J.: Enlarged search space for sitg parsing. In: Proc. 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT), Los Angeles, USA (June 2010)
8. Wu, D.: Trainable coarse bilingual grammars for parallel text bracketing. In: Proceedings of the Third Annual Workshop on Very Large Corpora, pp. 69–81 (1995)
9. Petrov, S., Barrett, L., Thibaux, R., Klein, D.: Learning accurate, compact, and interpretable tree annotation. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, pp. 433–440. Association for Computational Linguistics (2006)
10. Paul, M.: Overview of the IWSLT 2009 Evaluation Campaign. In: Proc. of the International Workshop on Spoken Language Translation, Tokyo, Japan, pp. 1–18 (2009)
11. Germann, U.: Aligned hansards of the 36th parliament of canada (2001), http://www.isi.edu/natural-language/download/hansard/

Commute-Time Convolution Kernels for Graph Clustering

Normawati A. Rahman and Edwin R. Hancock

Department of Computer Science, University of York,
Heslington, York, YO10 5DD, UK

Abstract. Commute time has proved to be a powerful attribute for clustering and characterising graph structure, and which is easily computed from the Laplacian spectrum. Moreover, commute time is robust to deletions of random edges and noisy edge weights. In this paper, we explore the idea of using convolution kernel to compare the distributions of commute time over pairs of graphs. We commence by computing the commute time distance in graphs. We then use a Gaussian convolution kernel to compare distributions. We use kernel kmeans for clustering and use kernel PCA for illustration using the COIL object recognition database.

Keywords: commute times, laplacian, graph kernel, convolution kernel.

1 Introduction

There has recently been a concerted effort in the literature to extend the kernel paradigm from pattern-vectors to relational structures such as graphs, trees and strings. In particular, various types of graph kernel have been suggested, with specific goals in mind [5,8,9,11,13,14,15,18,24]. Generally speaking, there are two sources of information that can be exploited in the construction of a graph kernel. Firstly, there is information concerning the structure of the graph. This can be encapsulated in a number of ways. However, one of the most powerful is to use information concerning the distribution of path length or the frequency of different cycle lengths. The second source of information is conveyed by the labels or attributes on the nodes or edges of a graph. Finally, there are correspondences between the nodes of the graphs being compared. There have been many examples of the use of kernels in conjunction with graphs for instance Smola et al [2], develop path-length kernels and use them from comparing molecular structures. Gartner et al [11], use kernels for mining graphs from large databases. Bunke and Riesen [3] have shown how kernel methods can be used to transform pattern analysis problems using graphs into equivalent statistical pattern analysis tasks.

These three sources of information are obtained at different cost and play different roles. For instance, reliable node correspondences are both difficult and costly to locate. If correspondences are to hand, then the attributes can provide powerful discriminating information. In fact, the use of correspondences can be viewed as implicitly vectorising the available attribute information for the graphs

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 316–323, 2010.

under study. Relational structure, on the other hand is something that is intrinsic to graphs, and distinguishes them from data in the form of vectors or strings.

The aim in this paper is to explore the extent to which structural information alone can be used to construct a kernel, and the kernel used for the purposes of graph clustering. We require a structural characterisation which is both economically computed and robust to minor perturbations in graph structure due to noise. The characterisation also needs to be fine enough so as to distinguish reasonably subtle changes in structure. Path lengths provide one candidate which have been extensively explored in the graph kernels literature. Examples include the path-length kernel [1,23] and the diffusion map [16]. However, commute time [19] provides an interesting alternative that captures the features of these two alternatives in a robust way. The commute time is the expected number of steps for a random walk to travel from one node of a graph to another, and then return again. The quantity is averaged over all possible paths. As a result it is relatively robust to edge deletion. It can also be shown to average the diffusion map over all possible diffusion lengths. Moreover, it is a metric and can be simply computed from the Laplacian spectrum in a time that is cubic in the number of nodes in the graph.

To avoid the need for explicit correspondences, we make use of the convolution kernel. Our idea is to compute the commute time between all pairs of nodes in a graph using the Laplacian spectrum, yielding a commute time matrix. We then compare the elements of the commute time matrices using the convolution kernel. This avoids explicit element by element correspondences, which would normally require the estimation of a permutation matrix between the nodes of the graph. Instead the convolution matrix weights against commute times that differ in magnitude. Viewed in this way the convolution kernel has features reminiscent of the computation of Hausdorff distance between graphs [12].

2 Commute Time on Graph

We commence by briefly reviewing the relationship between the Laplacian spectrum and the commute time. Consider the weighted graph $G = (V, E, \Omega)$ where V is the set of nodes, $E \subseteq V \times V$ is the set of edges and $\Omega : E \rightarrow [0, 1]$ is the set of weights associated with the edges. The adjacency matrix of the graph has elements

$$A_{uv} = \begin{cases} 1 & \text{if } (u, v) \in E \\ 0 & \text{otherwise,} \end{cases}$$

Let $T = diag(d_u; u \in V)$ be the diagonal weighted degree matrix with elements $\sum_{v=1}^{n} A(u, v)$, where $n = |V|$. The Laplacian matrix is given by $L = T - A$. The normalized weighted Laplacian matrix is defined to be $\mathrm{L} = T^{-1/2}(T - A)T^{-1/2}$, and has elements

$$\mathrm{L}_{uv} = \begin{cases} 1 & \text{if}\, u = v \\ -\frac{w_{u,v}}{\sqrt{d_u d_v}} & \text{if}\, u \neq v \,\text{and}\, (u, v) \in E \\ 0 & \text{otherwise,} \end{cases}$$

The spectral decomposition of the normalized Laplacian is $\mathrm{L} = \Phi \Lambda \Phi^T$, where $\Lambda = diag(\lambda_1, \lambda_2, ...\lambda_n)$ is the diagonal matrix with the eigenvalues as elements satisfying the ordering $0 = \lambda_1 \leq \lambda_2 ... \leq \lambda_{|V|}$ and $\Phi = (\phi_1 \mid \phi_2 \mid ... \mid \phi_{|V|})$ is the matrix with the correspondingly ordered eigenvectors as columns.

The Green's function Γ is the pseudo inverse of the normalized Laplacian matrix, which is computed by discarding the eigenvector associated with the zero eigenvalue, i.e.

$$\Gamma(u,v) = \sum_{i=2}^{|V|} \frac{1}{\lambda_i} \phi_i(u) \phi_i(v) \tag{1}$$

The expected number of steps taken by a random walk to reach node v, commencing from node u is defined as the hitting time $O(u,v)$, and the commute time $CT(u,v)$ is the expected time for a random walk to travel from node u to reach node v and then return. Thus, $CT(u,v) = O(u,v) + O(v,u)$, and it has been shown in [4],that it can be computed using the Green's function Γ by,

$$CT(u,v) = vol(T^{-1/2}(\Gamma(u,u) + \Gamma(v,v) - 2\Gamma(u,v))T^{-1/2}) \tag{2}$$

where vol is the volume of the graph. It has been proven in [4], that by substituting the spectral expression for the Green's function into the definition of the commute time, it is straightforward to show that in terms of the eigenvalues and eigenvectors of the normalized Laplacian [19],

$$CT(u,v) = vol \sum_{i=2}^{|V|} \frac{1}{\lambda_i} \left(\frac{\phi_1(u)}{\sqrt{d_u}} - \frac{\phi_i(v)}{\sqrt{d_v}} \right)^2 \tag{3}$$

Thus, the only operation required to compute the commute time matrix is the extraction of the eigenvectors and eigenvalues of the normalized Laplacian L. The commute time is a metric on the graph. In the next section, we'll show how we use the commute time matrix to compare the distribution of this metric over the graphs.

3 Convolution Kernels on Graph

Our objective in this paper is to use the graph kernel to compare the distribution of the commute time over the graphs and hence gauge graph similarity. One way to do this is by constructing a convolution graph kernel between the two graphs being considered.

Consider two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ with commute time matrices CT_1 and CT_2, representing the distributions of commute time within each graph. We wish to work without locating explicit correspondences between nodes of the two graphs. Hence, we make use of the convolution kernel to compare the commute-time matrices.

Convolution kernels were introduced by Haussler [10]. Consider two pattern-vectors $\boldsymbol{x} = x_1, ...x_D$ and $\boldsymbol{y} = y_1, ...y_D$. Suppose that for each corresponding pairs of elements (x_d, y_d) where $1 \leq d \leq D$, we have a kernel K_d that can be used to measure their similarity, i.e. $K_d(x_d, y_d)$. The similarity $K(x, y)$ between the two pattern-vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is given by the following generalized convolution

$$K(x, y) = \int_{\boldsymbol{x}} \int_{\boldsymbol{y}} \prod_{d=1}^{D} K_d(x_d y_d) d\boldsymbol{y} d\boldsymbol{x} \tag{4}$$

We can use any positive definite kernel with this definition. Here we use the Gaussian kernel to compare the commute time matrices for the graphs on an element by element basis. As a result the convolution kernel is

$$K(G_1, G_2) = \frac{1}{|V_1|} \frac{1}{|V_2|} \sum_{(a,b)\in V_1 \times V_1} \sum_{(\alpha,\beta)\in V_2 \times V_2} \exp^{-\frac{|CT_1(a,b)-CT_2(\alpha,\beta)|^2}{2\sigma^2}} \tag{5}$$

We have applied kernel PCA [21] to the convolution kernel for visualisation. Kernel PCA is the extension of PCA to a kernel feature space. Kernel kmeans has been well used for clustering. Here, we would like to cluster our kernel matrix based on kernel kmeans algorithm. Kernel kmeans [20] is a generalization of the standards kmeans algorithm where data points from input space are mapped into higher dimensional feature space through a nonlinear transformation ϕ and then kmeans is applied in the feature space. Clustering a test set of objects using the kernel kmeans approach involves partitioning the dataset into M disjoint clusters.

4 Experiment

In this section, we provide some experimental evaluation of our proposed method. In our experiments, we use two datasets. The first of these is the COIL [17] database of 72 views of each of a number of objects (see Figure 1). The second is a database containing 72 views of four toys (example images are shown in Figure 2). The datasets contain multiple images of objects as the camera pans around the object. Corner features for the COIL data and SIFT features for the toys, are extracted from the images and Delaunay graphs representing the arrangement of feature points are constructed.

We commence the experiment by computing the commute time distance for the graphs based on equation (3). With the value of the commute time at hand, we construct the kernel matrix based on equation (5). Then the data is embedded in a low dimensional space using kernel PCA and the kernel k-means algorithm used for clustering.

We apply kernel PCA to the output of the convolution kernel using the algorithm described in [22] and obtain an embedding of the graph data. Figure 3 shows the results obtained. Here the different coloured points refer to different objects. We also compare our method with the shortest path kernel on a graph and the modified Hausdorff distance. The path kernels used were constructed by computing the product of kernel on edges encountered along the walk. Here, we only take the edge weights into consideration in order to compute the path kernel. This is because we work with unlabeled graphs in the implementation of

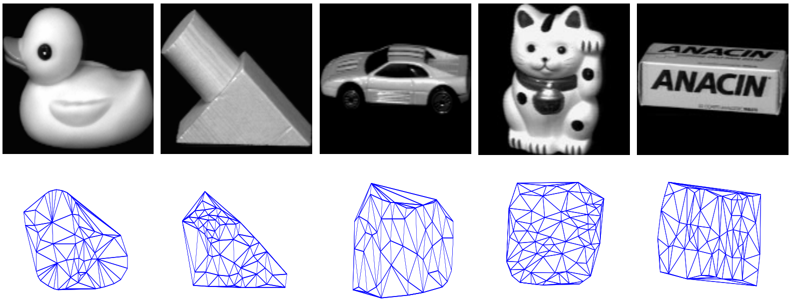


Fig. 1. Coil images that are used for the experiment and the corresponding delaunay graphs

Fig. 2. Examples images from the toys database

our kernel based on commute time matrix, and we wish to maintain comparability. The results for the path-length kernel are shown in Figure 4. The results obtained using the modified Hausdorff are shown in Figure 5. Here we compute distances between the commute time matrices using the modified Hausdorff distance reported in [7] and then embed the graphs into a low dimensional pattern space using multidimensional scaling [6]. Here we compare the results for both unweighted and weighted graphs.

From visual inspection of the results, there is some suggestion that the convolution kernel gives better results than the alternative two methods. Specifically, the different objects are separated into clearer clusters. To be more quantitative, we use the kernel K-means to perform clustering. Here we consider the first 5 objects from the COIL database. We use Rand index to validate the cluster. Rand index measures the consistency of a given clustering, therefore higher values indicate better clustering. We regard all pairs of objects (o_i, o_j) with $o_i \neq o_j$ for computing the Rand index. We denote pairs of object (o_i, o_j) that belong to same class and to same cluster with N_1, whereas N_2 denotes the number of pairs that neither belong to the same class nor to the same cluster. The number of pairs belonging to the same class but not to the same cluster is denoted as N_3, while N_4 represents the number of pairs belonging to different class but same cluster. The Rand index is thus defined as Rand index $= (N_1 + N_2)/(N_1 + N_2 + N_3 + N_4)$.

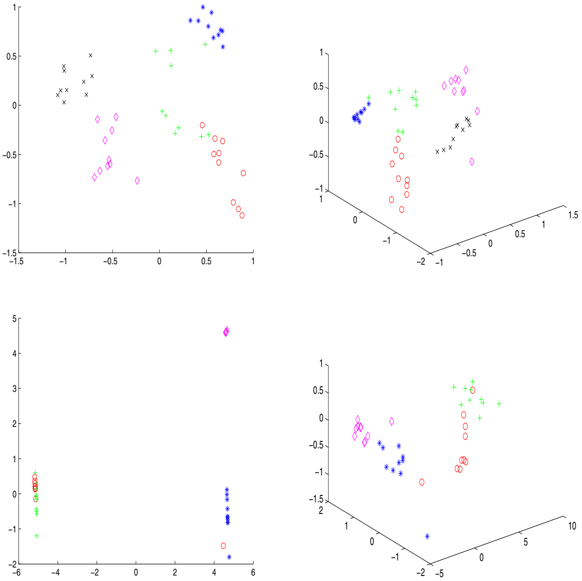


Fig. 3. Kernel PCA embedding of Gaussian convolution kernel using commute time as features.Top: Coil database. Bottom: Toys Database.

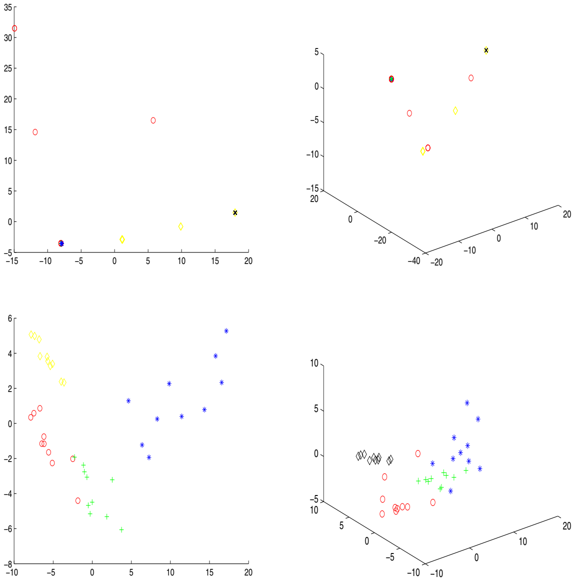


Fig. 4. Kernel PCA embedding of shortest path kernel. Top: Coil database. Bottom: Toys Database.

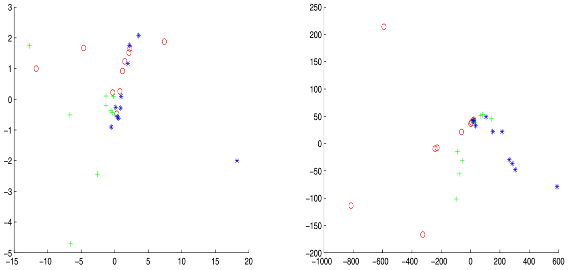


Fig. 5. Modified Hausdorff distance of Commute Time using MDS embedding. Left:Unweighted graph. Right: Weighted graph.

Table 1. Rand index for different value of σ

Rand index			
Data	$\sigma = 0.1$	$\sigma = 1$	$\sigma = 10$
COIL data using commute time kernel	0.7954	0.7286	0.6587
COIL data using path length kernel	0.7202	0.6714	0.6714
toys data using commute time kernel	0.7670	0.6330	0.6330
toys data using path length kernel	0.6770	0.6670	0.6670

Table 1 shows the Rand index for the COIL and toys dataset computed using commute time kernel and path length kernel with different value for σ.

From table 1, we can see that the performance of commute time kernel is comparable to shortest path kernel if not better as it gives higher values when σ is close to 0.1 for both COIL and toys datasets.

5 Conclusion

We have shown how the distribution of the commute time between pairs of vertices in the graph can be used to compute a measure of similarity between graphs using the convolution kernel. We have illustrated the effectiveness of the similarities on graph clustering and classification experiments. The commute time between vertices can be used as a more robust measure of vertex affinity than the path length distance as it is more robust to errors in edge weight structure.

References

1. Kriegel, H.P., Borgwardt, K.M.: Shortest path kernels on graphs. In: Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM 2005), pp. 74–81 (2005)
2. Ong, C.S., Schönauer, S., Vishwanathan, S.V.N., Smola, A.J., Kriegel, H.-P., Borgwardt, K.M.: Protein function prediction via graph kernels. In: In Proceedings of Intelligent Systems in Molecular Biology (ISMB), Detroit, USA (2005)

3. Bunke, K., Riesen, H.: A family of novel graph kernels for structural pattern recognition. LNCS, pp. 20–31. Springer, Heidelberg (2007)
4. Yau, S.T., Chung, F.: Discrete green's function. Journal of Combinatorial Theory Series A 91(1-2), 191–214 (2004)
5. Duffy, N., Collins, M.: Convolution kernels for natural language. Advances in Neural Information Processing Systems 14 (2002)
6. Cox, M.A.A., Cox, T.F.: Multidimensional Scaling. Chapman and Hall, Boca Raton (2001)
7. Jain, A., Dubuisson, M.: A modified haursdoff distance for object matching. In: Proc. 12th Int. Conf. Pattern Recognition, pp. 566–568 (1994)
8. Flach, P., Wrobel, S., Gartner, T.: On graph kernels: Hardness results and efficient alternatives. In: Scholkopf, B., Warmuth, M. (eds.) Sixteen Annual Conference on Computational Learning Theory and Seventh Kernel Workshop, COLT (2003)
9. Gartner, T.: Exponential and geometric kernels for graphs. In: NIP 2002 Workshop on Unreal Data, Volume Principles of Modelling Nonvectorial Data (2002)
10. Haussler, D.: Convolution kernels on discrete structure. Technical report, UCSC-CRL-99-10, UC Sansta Cruz (1999)
11. Gartner, T., Wrobel-S. Hovarth, T.: Cyclic pattern kernels for predictive graph mining. In: Proceedings of International Conference on Knowloedge Discovery and Data Mining (KDD), pp. 158–167 (2004)
12. Hancock, E.R., Huet, B.: Relational object recognition from large structural libraries. Pattern Recognition 35(9), 1895–1915 (2002)
13. Tsuda, K., Inokuchi-A. Kashima, H.: Marginalized kernels between labelled graphs. In: In Proceedings of the 20th International Conference on Machine Learning (ICML), Washington, DC, United States (2003)
14. Saunders, C., Shawe-Taylor, -J., Christianini, N., Watkins, C., Lodhi, H.: Text classification using string kernels. Journal of Machine Learning Research 2, 419–444 (2002)
15. Ueda, N., Akutsu, -T., -Perret, J.-L., Vert, J.-P., Mahe, P.: Extensions of marginalized graph kernels. In: ICML (2004)
16. Lafon, S., coifman, -R.R., -Kevrekidis, I.G., Nadler, B.: Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In: Advances in Neural Information Processing Systems, vol. 18 (2005)
17. Nayar, S., Murase, -H., Nene, S.: Columbia object image library: Coil (1996)
18. Bunke, H., Neuhaus, M.: Edit distance based kernel functions for structural pattern classification. Pattern Recognition 39(10), 1852–1863 (2006)
19. Hancock, E.R., Qiu, H.: Clustering & embedding using commute times. IEEE Transactions on Pattern Analysis & Machine Intelligence(PAMI) 29(11), 1873–1890 (2007)
20. Smola, A., Muller, -K.R., Scholkopf, B.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10, 1299–1319 (1998)
21. Smola, A.J., Muller, -K.-R., Scholkopf, B.: Kernel principal component analysis. In: Scholkopf, B., Burges, C.J.C., Smola, A.J. (eds.) Advances in Kernels Methods-Support Vector Learning, pp. 327–352 (1999)
22. Christianini, N., Shawe-Taylor, J.: Kernel Methods for Pattern Analysis. Cambridge University Press, New York (2004)
23. Borgwardt, K.M., Schraudolph, -N.N., Vishwanathan, S.: Fast computation of graph kernels. In: Adv. NIPS (2007)
24. Smola, -J., Vishwanathan, S.V.N.: Fast kernels for strings and tree matching. Advances in Neural Information Processing Systems 15 (2003)

Non-Euclidean Dissimilarities: Causes and Informativeness

Robert P.W. Duin[1] and Elżbieta Pękalska[2]

[1] Faculty of Electrical Engineering, Mathematics and Computer Sciences,
Delft University of Technology, The Netherlands
r.duin@ieee.org
[2] School of Computer Science, University of Manchester, United Kingdom
pekalska@cs.man.ac.uk

Abstract. In the process of designing pattern recognition systems one may choose a representation based on pairwise dissimilarities between objects. This is especially appealing when a set of discriminative features is difficult to find. Various classification systems have been studied for such a dissimilarity representation: the direct use of the nearest neighbor rule, the postulation of a dissimilarity space and an embedding to a virtual, underlying feature vector space.

It appears in several applications that the dissimilarity measures constructed by experts tend to have a *non*-Euclidean behavior. In this paper we first analyze the causes of such choices and then experimentally verify that the non-Euclidean property of the measure can be informative[1].

1 Introduction

Dissimilarities are a natural way to represent objects. Some consider them as more fundamental than features [1]. This paper studies particular aspects of dissimilarities. First, we analyze why *non*-Euclidean dissimilarities arise in recognition. Then, we discuss how non-Euclidean relations can become informative.

Dissimilarities have been studied in [2] for both supervised and unsupervised learning as an alternative to the use of features in building representations. They are especially useful in two contexts. First, when no clear properties are available to become features and, secondly, when objects can be compared globally such as shapes in images, time signals or spectra. Classifiers relying on dissimilarity relations can outperform nearest neighbor approaches or template matching.

There are two main approaches for building vector spaces from dissimilarities. One postulates a Euclidean space, the so-called dissimilarity space, in which features are defined by dissimilarities to a representation set of objects. The other relies on a linear embedding of the given dissimilarity matrix. The first is very general and can always be used. It demands a proper choice of the representation

[1] We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250) as well as the Engineering and Physical Sciences Research Council in the UK.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 324–333, 2010.

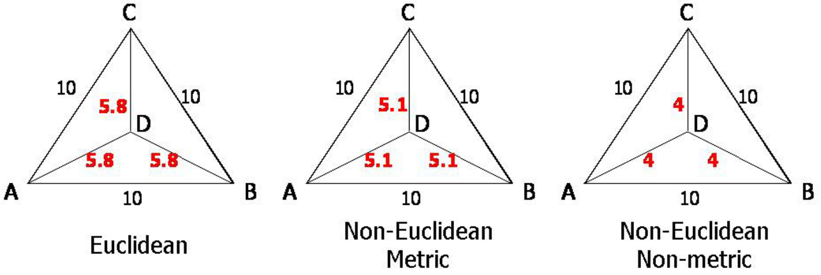


Fig. 1. Illustration of the difference between Euclidean, metric, but non-Euclidean and non-metric dissimilarities. If the distances between the four points A, B, C and D are given as in the left plot, then an exact 2-dimensional Euclidean embedding is possible. If the distances are given as given as in the middle plot, the triangle inequality is obeyed. So the given distances are metric, but no isometric Euclidean embedding exists. The distances in the right plot are non-Euclidean as well as non-metric.

set, a problem similar to feature selection [3]. In the selection of the representation set the intrinsic nature of the dissimilarities may be used, e.g. such that the closer or the more likely objects belong to the same class. In the construction of classifiers in this space, dissimilarities may be used in the same way as features. This, however, neglects their original character of pairwise dissimilarities.

On the contrary, in the embedding of a dissimilarity matrix to a space with a given metric, the nature of dissimilarities is preserved. It is natural to search for an embedding to a Euclidean space as the Euclidean metric is assumed either implicitly or explicitly in many classification systems. It appears however that in many applications *non*-Euclidean and even non-metric dissimilarities are used due to their good performance in template matching. (See Fig. 1 to understand the difference between non-Euclidean and non-metric dissimilarities.) An early example is given by Dubuisson and Jain [4] who showed that in a set of image object matching examples the non-metric modified Hausdorff distance outperforms the original metric Hausdorff distance. Non-Euclidean distances can only be approximately embedded in a Euclidean space. Goldfarb showed how a so-called Pseudo-Euclidean (PE) embedding can be found [5] for any symmetric dissimilarity matrix. It is error free, but requires a different distance measure. Some classifiers can be defined in this space, such as the nearest mean, nearest neighbor, Parzen classifier, LDA and QDA. The relation of the latter three with densities is not clear yet, as the concept of probability density distributions has not been well defined for the PE space.

The question on usefulness or non-importance of non-Euclidean distances has been around for some time. Goldfarb did not find good applications for the PE space and abandoned all vector space approaches [5]. Instead, he focussed on the Evolving Transformation System (ETS) which by a structural representation aimed to model relations between objects [6], but for which it was difficult to find classifiers [7]. This is common for structural approaches, but is solved in a heuristic way by the use of a dissimilarity space, in which the non-Euclidean nature of the object relations in neglected. In [8] some studies are presented indicating that non-Euclideaness of the data (i.e. the deviation from Euclidean distances) might contribute to the classification performance. In [9] Euclidean corrections of non-Euclidean data are discussed.

Finding classifiers for non-Euclidean dissimilarities is directly related to study of indefinite kernels, important for the optimization of Support Vector Machines (SVMs). The quadratic programming solution used for SVMs is not guaranteed to be optimal for kernels that violate the Mercer conditions. As the inner product definition for the pseudo-Euclidean space leads to indefinite kernels, the construction of SVMs in such a space is thereby hampered. For that reason there is a strong tendency in the machine learning community to design positive semidefinite (psd), i.e. Mercer, kernels. On the other hand, since non-Euclidean dissimilarities are frequently used in pattern recognition applications, it is relevant to know how to deal with them. Should we avoid them, correct them into Euclidean distances to make them suitable for the full set of traditional classification tools, or keep them as they are and design special classifiers for non-Euclidean data?

In this paper we want to contribute to this discussion in two ways. In Section 3 we will analyze the causes behind non-Euclidean dissimilarities. In Section 4 we will argue why non-Euclidean dissimilarities can be informative and we will present some examples. First, however, the dissimilarity space and PE embedded space will be briefly introduced in Section 2.

2 Dissimilarity Representations

The dissimilarity representation has extensively been discussed, e.g. in [2] or [10], so we will only focus here on aspects that are essential for this paper.

Traditionally, dissimilarity measures were often optimized for the nearest neighbor classification performance. In addition, they were also widely used in hierarchical cluster analysis. Later, the resulting dissimilarity matrices served for the construction of vector spaces and the computation of classifiers. Only more recently proximity measures have been designed for classifiers that are more general than the nearest neighbor rule. These are usually similarities and kernels (but not dissimilarities), used in combination with SVMs. So, research on the design of dissimilarity measures such that they fit to a wider range of classifiers is still in an early stage. Consequently, we will restrict ourselves in this paper to the common practice of measures optimized for nearest neighbor classifiers. New objects are thereby classified just on the basis of pairwise comparisons. They are not represented in a vector space. An additional step is necessary to create such a space, and as a result, this will allow the use of other classifiers. The two ways investigated so far are the dissimilarity space and PE embedded space.

2.1 Dissimilarity Space

Let $\mathcal{X} = \{o_1, \ldots, o_n\}$ be a training set of objects o_i. These are not necessarily vectors but can be real world objects, or e.g. images or time signals. Given a dissimilarity function and/or dissimilarity data, we define a data-dependent mapping $D(\cdot, R) : \mathcal{X} \rightarrow \mathbb{R}^k$ from $\mathcal{X}$ to the so-called *dissimilarity space* [11,12,13]. The

k-element set R consists of objects that are representative for the problem. This set, the representation or prototype set, may be a subset of $\mathcal{X}$. In the dissimilarity space each dimension $D(\cdot, p_i)$ describes a dissimilarity to a prototype p_i from R. Here we will choose $R := \mathcal{X}$. As a result, every object is described by an n-dimensional vector $D(o, \mathcal{X}) = [d(o, o_1) \ \dots \ d(o, o_n)]^T$, which are just the rows of the given dissimilarity matrix D. The resulting vector space is endowed with the traditional inner product and the Euclidean metric. As we have n training objects in an n-dimensional space, a classifier such as SVM is needed to handle this situation.

2.2 Pseudo-Euclidean Embedded Space

A Pseudo-Euclidean (PE) space $\mathcal{E} = \mathbb{R}^{(p,q)} = \mathbb{R}^p \oplus \mathbb{R}^q$ is a vector space with a non-degenerate indefinite inner product $\langle \cdot, \cdot \rangle_{\mathcal{E}}$ such that $\langle \cdot, \cdot \rangle_{\mathcal{E}}$ is positive definite on $\mathbb{R}^p$ and negative definite on $\mathbb{R}^q$ [5,2]. The inner product in $\mathbb{R}^{(p,q)}$ is defined (using an orthonormal basis) as $\langle x, y \rangle_{\mathcal{E}} = x^T \mathcal{J}_{pq} y$, where $\mathcal{J}_{pq} = [I_{p \times p} \ 0; \ 0 \ -I_{q \times q}]$ and I is the identity matrix. As a result, $d^2_{\mathcal{E}}(x, y) = (x - y)^T \mathcal{J}_{pq} (x - y)$.

Any symmetric $n \times n$ dissimilarity matrix D can be embedded into a $(n-1)$-dimensional PE space [5,2]. The eigenvalue decomposition needed for the embedding results in p positive and q negative eigenvalues λ_j, $p+q = n-1$, and the corresponding eigenvectors. To inspect the amount of non-Euclidean influence in the derived PE space, we use the negative eigenfraction (*NEF*)

$$NEF = \sum_{j=p+1}^{p+q} |\lambda_j| / \sum_{i=1}^{p+q} |\lambda_i| \in [0, 1] \tag{1}$$

as a measure for the non-Euclidean behavior of the dissimilarity matrix.

If the negative eigenvalues are considered as the result of noise or errors, they may be neglected. As a result, a 'corrected' dissimilarity matrix D_p may be computed by using a positive subspace $\mathbb{R}^p$ of the embedded space $\mathbb{R}^{(p,q)}$:

$$d^2_{\mathcal{E}p}(x, y) = (x_p - y_p)^T (x_p - y_p), \tag{2}$$

where x_p, y_p are projections of the vectors x, y from $\mathbb{R}^{(p,q)}$ onto the subspace $\mathbb{R}^p$ and all diagonal values of $\mathcal{J}_{pq}$ become $+1$. In order to investigate a possible contribution of the negative eigenvalues, the residue can be computed by:

$$d^2_{\mathcal{E}q}(x, y) = -(x_q - y_q)^T (x_q - y_q) \tag{3}$$

where x_q, y_q are projections of the vectors x, y from $\mathbb{R}^{(p,q)}$ onto the negative subspace $\mathbb{R}^q$ and all diagonal values of $\mathcal{J}_{pq}$ become -1. The complete dissimilarity matrix D can thereby be decomposed as

$$D^{*2} = D_p^{*2} - D_q^{*2} \tag{4}$$

in which the values of D_q^{*2} are positive and ^{*2} denotes an element-wise squaring. n-dimensional dissimilarity spaces may also be defined for D_p and D_q.

3 Causes of Non-Euclidean Dissimilarity Measures

In the previous section two procedures for deriving vector spaces are presented. One is general, but neglects the pairwise dissimilarity characteristics. The other is specific but suffers from the possible non-Euclidean relations. If we want to make use of the specific dissimilarity character, but suffer from the non-Euclidean behavior, it is important to analyze why this happens. Should we avoid it, should we correct it, or should we design special classifiers that deal with it?

First, it should be emphasized how common non-Euclidean measures are. In [2] an extensive overview of such measures has been given, but in many occasions we have encountered that this fact is not fully recognized. Almost all probabilistic distance measures are non-Euclidean. This implies that by dealing with object invariants, the dissimilarity matrix resulting from the overlap between the object pdfs is non-Euclidean. Also the Mahalanobis class distance as well as the related Fisher criterion are non-Euclidean. Consequently many non-Euclidean distance measures are used in cluster analysis and in the analysis of spectra in chemometrics and hyperspectral image analysis.

In shape recognition, various dissimilarity measures are used based on the weighted edit distance, on variants of the Hausdorff distance or on non-linear morphing. Usual parameters are optimized within an application w.r.t. the performance based on template matching and other nearest neighbor classifiers [14]. Almost all have non-Euclidean behavior and some are even non-metric [4].

In the design and optimization of the dissimilarity measures for template matching, their Euclidean behavior is not an issue. With the popularity of support vector machines (SVMs), it has become important to design kernels (similarities) which fulfill the Mercer conditions. This is equivalent to a possibility of an isometric Euclidean embedding of such a kernel (or dissimilarities). Next subsections discuss reasons that give rise to violations of these conditions leading to non-Euclidean dissimilarities or indefinite kernels.

3.1 Non-intrinsic Non-Euclidean Dissimilarities

Below we identify some non-intrinsic causes for non-Euclidean dissimilarities.

Numeric inaccuracies. Non-Euclidean dissimilarities arise due the numeric inaccuracies caused by the use of a finite word length. If the intrinsic dimensionality of the data is lower than the sample size, eigenvalues that should be zero during embedding, may become negative due to numeric inaccuracies. It is thereby advisable to neglect dimensions (features) that correspond to very small positive and negative eigenvalues.

Overestimation of large distances. Complex measures are used when dissimilarities are derived from raw data such as (objects in) images. They may define the distance between two objects as the length of the path that transforms one object into the other. Examples are the weighted edit distance [15] and deformable templates [16]. In the optimization procedure that minimizes the path length, the procedure may approximate the transformation costs from above. As a consequence, too large distances are found. If the distance measure is Euclidean, such errors make it non-Euclidean or even non-metric.

Underestimation of small distances. The underestimation of small distances has the same result as the overestimation of large distances. It may happen when the pairwise comparison of objects is based on different properties for every pair, like in studies on consumer preference data. Another example is the comparison of partially occluded objects in computer vision.

3.2 Intrinsic Non-Euclidean Dissimilarities

The causes discussed in the above may be judged as accidental. They result either from computational or observational problems. If better computers and observations were available, they would disappear. Now we will focuss on dissimilarity measures for which this will not happen. We will present three possibilities.

Non-Euclidean Dissimilarities. As already indicated at the start of this section, there can be arguments from the application side to use another metric than the Euclidean one. An example is the l_1-distance between energy spectra as it is related to energy differences. Although the l_2-norm is very convenient for computational reasons and it is rotation invariant in a Euclidean space, the l_1-norm may naturally arise from the demands in applications.

Invariants. A very fundamental reason is related to the occurrence of invariants. Frequently, one is not interested in the dissimilarity between objects A and B, but between their equivalence classes i.e. sets of objects $A(\theta)$ and $B(\theta)$ in which θ controls an invariant. One may define the dissimilarity between the A and B as the minimum difference between the sets defined by all their invariants.

$$d^*(A,B) = \min_{\theta_A} \min_{\theta_B} (d(A(\theta_A), B(\theta_B))) \tag{5}$$

This measure is non-metric: the triangle inequality may be violated as for different pairs of objects different values of θ are found minimizing (5).

Sets of vectors. Complicated objects like multi-region images may be represented by sets of vectors. Distance measures between such sets have already been studied for a long time in cluster analysis. Many are non-Euclidean or even non-metric, e.g. the single linkage procedures. It is defined as the distance between the two most neighboring points of the two clusters being compared, is non-metric. It even holds that if $d(A,B) = 0$, then it does not follow that $A \equiv B$.

For the single linkage dissimilarity measure it can be understood why the dissimilarity space may be useful. Given a set of such dissimilarities between clouds of vectors, it can be concluded that two clouds are similar if the entire sets of dissimilarities with all other clouds are about equal. If just their mutual dissimilarity is (close to) zero, they may still be very different.

The problem with the single linkage dissimilarity measure between two sets of vectors points to a more general problem in relating sets and even objects. In [17] an attempt has been made to define a proper Mercer kernel between two sets of vectors. Such sets are in that paper compared by the Hellinger distance derived from the Bhattacharyya's affinity between two pdfs $p_A(x)$ and $p_B(x)$ found for the two vector sets A and B:

$$d(A,B) = \left[\int \left(\sqrt{p_A(x)} - \sqrt{p_B(x)} \right)^2 \right]^{1/2}. \tag{6}$$

The authors state that by expressing $p(x)$ in any orthogonal basis of functions, the resulting kernel K is automatically positive semidefinite (psd). This is only correct, if all vector sets $A, B, ...$ to which the kernel is applied have the same basis. If different bases are derived in a pairwise comparison of sets, the kernel will become indefinite.

This makes clear that indefinite relations may arise in any pairwise comparison of real world objects if they are first represented in some joint space for the two objects, followed by a dissimilarity measure. These joint spaces may be different for different pairs! Consequently, the total set of dissimilarities will likely have a non-Euclidean behaior, even if a single comparison is defined as Euclidean, as in (6).

4 Informativeness

Are non-Euclidean dissimilarity measures informative? How should this question be answered? It is different than the question whether non-Euclidean measures are better than Euclidean ones. This second question can certainly not be answered in general. After we study a set of individual problems and compare a large set of dissimilarity measures we may find that for some problems of interest the best measure is non-Euclidean. Such a result is always temporal. A new Euclidean measure may later be found that outperforms the earlier ones.

The question of informativeness however may be answered in an absolute sense. Even if a particular measure is not the best one, its non-Euclidean characteristic can be informative as by removing it, performance deteriorates. Should this result also be found by a classifier in the non-Euclidean space? If an Euclidean correction can be found for an initial non-Euclidean representation that enables the construction of a good classifier, is the non-Euclidean dissimilarity measure then informative? We answer this question positively as any transformation can be included in the classifier and thereby effectively a classifier for the non-Euclidean representation has been found.

We will therefore state that the non-Euclidean character of a dissimilarity measure is non-informative if the classification result improves by removing its non-Euclidean characteristic. The answer may be different for different classifiers. The traditional way of removing the non-Euclidean characteristic is by neglecting the negative eigenvectors in the pseudo-Euclidean embedding. This is represented by the recomputed dissimilarities in the positive part of the pseudo-Euclidean space, D_p in (4). The dissimilarity representation based on D_q isolates the non-Euclidean characteristic of the given dissimilarity matrix D and can be used as a check to see whether there is any class separability visibility in the removed part of the embedding.

Table 1. Classification errors of the linear SVM for several representations using leave-one-out crossvalidation

	size	classes	Non-Metric	NEF	Rand Err	Original, D	Positive, D_p	Negative, D_q
Chickenpieces45	446	5	0	0.156	0.791	**0.022**	0.132	0.175
Chickenpieces60	446	5	0	0.162	0.791	**0.020**	0.067	0.173
Chickenpieces90	446	5	0	0.152	0.791	**0.022**	0.052	0.148
Chickenpieces120	446	5	0	0.130	0.791	**0.034**	0.108	0.148
FlowCyto	612	3	1e-5	0.244	0.598	0.103	**0.100**	0.327
WoodyPlants50	791	14	5e-4	0.229	0.928	**0.075**	0.076	0.442
CatCortex	65	4	2e-3	0.208	0.738	**0.046**	0.077	0.662
Protein	213	4	0	0.001	0.718	0.005	**0.000**	0.634
Balls3D	200	2	3e-4	0.001	0.500	**0.470**	0.495	0.000
GaussM1	500	2	0	0.262	0.500	**0.202**	**0.202**	0.228
GaussM02	500	2	5e-4	0.393	0.500	0.204	**0.174**	0.252
CoilYork	288	4	8e-8	0.258	0.750	**0.267**	0.313	0.618
CoilDelftSame	288	4	0	0.027	0.750	**0.413**	0.417	0.597
CoilDelftDiff	288	4	8e-8	0.128	0.750	**0.347**	0.358	0.691
NewsGroups	600	4	4e-5	0.202	0.733	**0.198**	0.213	0.435
BrainMRI	124	2	5e-5	0.112	0.499	0.226	**0.218**	0.556
Pedestrians	689	3	4e-8	0.111	0.348	**0.010**	0.015	0.030

We analyze a set of public domain dissimilarity matrices used in various applications, as well as a few artificially generated ones. The details of the sets are available from the D3.3 deliverable of the EU SIMBAD project[2]. See Table 1 for some properties: *size* (number of objects), (number of) *classes*, *non-metric* (fraction of triangle violations), *NEF* (negative eigenfraction) and *Rand Err* (classification error by random assignment). Every dissimilarity matrix is made symmetric by averaging with its transpose and normalized by the average off-diagonal dissimilarity. We compute the linear SVM in the dissimilarity spaces based on the original, 'positive' and 'negative' dissimilarities D, D_p and D_q. Error estimates are based on leave-one-out crossvalidation. These experiments are done in a transductive way: test objects are included in the derivation of the embedding as well as the dissimilarity representations.

The four Chickenpieces datasets are the averages of 11 dissimilarity matrices derived from a weighted edit distance between blobs [15]. FlowCyto is the average of four specific histogram dissimilarities including an automatic calibration correction. WoodyPlants is a subset of the shape dissimilarities between leaves of woody plants [10]. We used classes with more than 50 objects. Catcortex is based on the connection strength between 65 cortical areas of a cat, [12]. Protein measures protein sequence differences using an evolutionary distance measure [18]. Balls3D is an artificial dataset based on the surface distances of randomly

[2] http://simbad-fp7.eu/

positioned balls of two classes having a slightly different radius. GaussM1 and GaussM02 are based on two 20-dimensionally normally distributed sets of objects for which dissimilarities are computed using the Minkowsky distances 1 respectively 0.2. The three Coil datasets are based on the same sets of SIFT points in COIL images compared by different graph distances. BrainMRI is the average of 182 dissimilarity measures obtained from MRI brain images. Pedestrians is a set of dissimilarities between detected objects (possibly pedestrians) in street images of the classes 'pedestrian', 'car', 'other'. They are based on cloud distances between sets of feature points derived from single images.

5 Discussion and Conclusions

In this paper we identify a number of causes that give rise to non-Euclidean and non-metric dissimilarities and we wonder whether they can play an informative role for classification purposes. In the above table some phenomena can be observed that illustrate these issues and answer some questions.

- From the negative eigenfraction column (NEF) it can be understood that all datasets are non-Euclidean. Protein set has a nearly Euclidean measure as it has just a very small contribution from the negative eigenvalues.
- A number of datasets is metric. Chickenpieces, as we used the dataset here, based on averages of weighted edit-distances between contours, should be metric as the edit-distance searches for the smallest edit path. In the individual dissimilarities matrices some violations can be observed due to approximations in the path optimization procedure. After averaging this is solved. Interesting is that this procedure improves the results significantly. The performances found in the dissimilarity space are to our knowledge the best ever published for this dataset .
- The original, uncorrected, pseudo-Euclidean dissimilarities are the best (in bold) in many cases. For these the deletion of the negative eigenvectors works counter-productive.
- For the other datasets the Euclidean correction works out well.
- However, in almost all cases the negative part of the space alone shows some separability of the classes (compare with the random assignment error), proving that it contains some information.
- In a few cases the negative space shows very good results, e.g. Pedestrians.
- In the Balls3D example all information is concentrated in the negative space.

In conclusion it is stated that the non-Euclidean characteristic of dissimilarity data, resulting from the search of the best representation for nearest neighbor assignment should not be directly removed from the representation as by using the positive space only. This space performs often similar or worse compared to the original dissimilarities. The negative space itself, concentrating all non-Euclidean characteristics, yields usually a better than random performance and surprisingly leads to a very good result in some problems. From these two observations, removing the negative space often deteriorates results and the negative space alone shows a better than random performance, it is concluded that negative space and thereby the non-Euclideaness of the data is informative. It should be realized that these conclusions are classifier dependent.

References

1. Edelman, S.: Representation and Recognition in Vision. MIT Press, Cambridge (1999)
2. Pękalska, E., Duin, R.: The Dissimilarity Representation for Pattern Recognition. In: Foundations and Applications. World Scientific, Singapore (2005)
3. Pękalska, E., Duin, R., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recognition 39(2), 189–208 (2006)
4. Dubuisson, M., Jain, A.: Modified Hausdorff distance for object matching. In: Int. Conference on Pattern Recognition, vol. 1, pp. 566–568 (1994)
5. Goldfarb, L.: A new approach to pattern recognition. In: Kanal, L., Rosenfeld, A. (eds.) Progress in Pattern Recognition, vol. 2, pp. 241–402. Elsevier, Amsterdam (1985)
6. Goldfarb, L., Gay, D., Golubitsky, O., Korkin, D.: What is a structural representation? second version. Technical Report TR04-165, University of New Brunswick, Fredericton, Canada (2004)
7. Duin, R.P.W.: Structural class representation and pattern recognition by ets; a commentary. Technical report, Delft University of Technology, Pattern Recognition Laboratory (2006)
8. Pękalska, E., Harol, A., Duin, R., Spillmann, B., Bunke, H.: Non-Euclidean or non-metric measures can be informative. In: S+SSPR, pp. 871–880 (2006)
9. Duin, R.P.W., Pekalska, E., Harol, A., Lee, W.J., Bunke, H.: On euclidean corrections for non-euclidean dissimilarities. In: Structural, Syntactic, and Statistical Pattern Recognition, pp. 551–561 (2008)
10. Jacobs, D., Weinshall, D., Gdalyahu, Y.: Classification with Non-Metric Distances: Image Retrieval and Class Representation. IEEE TPAMI 22(6), 583–600 (2000)
11. Duin, R., de Ridder, D., Tax, D.: Experiments with object based discriminant functions; a featureless approach to pattern recognition. Pattern Recognition Letters 18(11-13), 1159–1166 (1997)
12. Graepel, T., Herbrich, R., Bollmann-Sdorra, P., Obermayer, K.: Classification on pairwise proximity data. In: Advances in Neural Information System Processing, vol. 11, pp. 438–444 (1999)
13. Pękalska, E., Paclík, P., Duin, R.: A Generalized Kernel Approach to Dissimilarity Based Classification. J. of Machine Learning Research 2(2), 175–211 (2002)
14. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph. Pattern Recognition Letters 19(3-4), 255–259 (1998)
15. Bunke, H., Bühler, U.: Applications of approximate string matching to 2D shape recognition. Pattern recognition 26(12), 1797–1812 (1993)
16. Jain, A.K., Zongker, D.E.: Representation and recognition of handwritten digits using deformable templates. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(12), 1386–1391 (1997)
17. Kondor, R.I., Jebara, T.: A kernel between sets of vectors. In: Fawcett, T., Mishra, N. (eds.) ICML, pp. 361–368. AAAI Press, Menlo Park (2003)
18. Graepel, T., Herbrich, R., Schölkopf, B., Smola, A., Bartlett, P., Müller, K.R., Obermayer, K., Williamson, R.: Classification on proximity data with LP-machines. In: ICANN, pp. 304–309 (1999)

Non-parametric Mixture Models for Clustering*

Pavan Kumar Mallapragada, Rong Jin, and Anil Jain

Department of Computer Science and Engineering,
Michigan State University, East Lansing, MI - 48824

Abstract. Mixture models have been widely used for data clustering. However, commonly used mixture models are generally of a parametric form (e.g., mixture of Gaussian distributions or GMM), which significantly limits their capacity in fitting diverse multidimensional data distributions encountered in practice. We propose a non-parametric mixture model (NMM) for data clustering in order to detect clusters generated from arbitrary unknown distributions, using non-parametric kernel density estimates. The proposed model is non-parametric since the generative distribution of each data point depends only on the rest of the data points and the chosen kernel. A leave-one-out likelihood maximization is performed to estimate the parameters of the model. The NMM approach, when applied to cluster high dimensional text datasets significantly outperforms the state-of-the-art and classical approaches such as K-means, Gaussian Mixture Models, spectral clustering and linkage methods.

1 Introduction

Data clustering aims to partition a given set of n objects represented either as points in a d dimensional space or as an $n \times n$ similarity matrix. The lack of a universal definition of a cluster, and its task or data dependent nature has resulted in publication of a very large number of clustering algorithms, each with different assumptions about the cluster structure [1]. Broadly, the proposed approaches can be classified into *parametric* vs. *non parametric* approaches. Parametric approaches impose a structure on the data, where as non-parametric methods infer the underlying structure from the data itself.

Probabilistic finite mixture modeling [2,3] is one of the most popular parametric clustering methods. Several probabilistic models like Gaussian Mixture Model (GMM) [3] and Latent Dirichlet Allocation [4] have been shown to be successful in a wide variety of applications concerning the analysis of continuous and discrete data, respectively. Probabilistic models are advantageous since they provide principled ways to address issues like the number of clusters, missing feature values, etc. Parametric mixture models are effective only when the

* The research was partially supported by ONR grant no. N000140710225, NSF grant no. IIS-0643494. Part of Anil Jain's research was supported by WCU(World Class University) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology(R31-2008-000-10008-0).

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 334–343, 2010.

underlying distribution of the data is either known, or can be closely approximated by the distribution assumed by the model. This is a major shortcoming since it is well known that clusters in real data are not always of the same shape and rarely follow a "nice" distribution like Gaussian [5]. In a general setting, each cluster may follow its own unknown distribution, which limits the performance of parametric mixture models. Similar shortcomings can be attributed to squared error based clustering algorithms such as K-means, which is one of the most popular clustering algorithms due to its ease of implementation and reasonable empirical performance [1].

The limitations of parametric mixture models can be overcome by the use of algorithms that exploit non-parametric density estimation methods. Several non-parametric clustering algorithms, for instance, Jarvis-Patrick [6], DBSCAN [7] and Mean-shift [8], have been proposed[1]. These methods first find a single kernel-density estimate of the entire data, and then detect clusters by identifying modes or regions of high density in the estimated density [8]. Despite their success, most of these approaches are not always successful in finding clusters in high-dimensional datasets, since it is difficult to define the neighborhood of a data point in a high-dimensional space when the available sample size is small [9]. For this reason, almost all non-parameteric density based algorithms have been applied only to low-dimensional clustering problems such as image segmentation [8,10]. Further, it is not possible to a priori specify the desired number of clusters in these methods.

In this paper, we assume that each cluster is generated by its own density function that is unknown. The density function of each cluster may be arbitrary and multimodal and hence it is modeled using a non-parametric kernel density estimate. The overall data is modeled as a mixture of the individual cluster density functions. Since the proposed approach, unlike other non-parametric algorithms (e.g., Spectral clustering), constructs an explicit probabilistic model for each cluster, it can naturally handle out-of-sample[2] clustering by computing the posterior probabilities for new data points. In summary, we emphasize that:

- The proposed approach is a non-parametric probabilistic model for data clustering, and offers several advantages compared to non-probabilistic models since (a) it allows for probabilistic assignments of data points to different clusters, unlike other non-parametric models (b) it can effectively explore probabilistic tools such as Dirichlet process and Gaussian process for non-parametric priors, and (c) the model naturally supports out of sample cluster assignments, unlike other non-parametric models.
- Contrary to most existing mixture models, the proposed approach does not make any explicit assumption about the parametric form of the underlying density function, and can model clusters following arbitrary densities.

[1] Although spectral clustering and linkage methods can be viewed as non-parametric methods, they are not discussed since they are not probabilistic models.

[2] A clustering algorithm can perform *out-of-sample* clustering if it can assign a cluster label to a data point unseen during the learning phase.

We show the performance of the proposed clustering algorithm on high-dimensional text datasets. Experiments demonstrate that, compared to several widely used clustering algorithms such as K-means and Spectral clustering, the proposed algorithm performs significantly better when data is of high dimensionality and is embedded in a low dimensional manifold.

2 Non-parametric Mixture Model

2.1 Model Description

Let $\mathcal{D} = \{x_1, \ldots, x_n\}$ be a collection of n data points to be clustered, where each $x_i \in \mathbb{R}^d$ is a vector of d dimensions. Let G be the number of clusters. We aim to fit the data points in $\mathcal{D}$ by a non-parametric mixture model. Let $\kappa(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ be the kernel function for density estimation. We further assume that the kernel function is stationary, i.e., $\kappa(x_i, x_j) = \kappa_s(x_i - x_j)$, where $\int \kappa_s(x)dx = 1$. We denote by the matrix $K = [\kappa(x_i, x_j)]_{n\times n} \in \mathbb{R}_+^{n\times n}$ the pairwise kernel similarity for data points in $\mathcal{D}$.

Let $\{c_g\}, g = 1, \ldots, G$ be the set of G clusters that forms a partition of $\mathcal{D}$. We specify the conditional density function $p_g(x|c_g, \mathcal{D})$ for each cluster c_g as follows:

$$p_g(x|c_g, \mathcal{D}) = \frac{1}{|c_g|} \sum_{x_i \in c_g} \kappa(x, x_i) \tag{1}$$

where $|c_g|$ is the number of samples in cluster c_g, and $\sum_g |c_g| = n$. The unconditional (on clusters) density $p(x|\mathcal{D})$ is then written as

$$p(x|\mathcal{D}) = \sum_{g=1}^{G} \pi_g p_g(x|c_g, \mathcal{D}) \tag{2}$$

where $\pi_g = P(c_g)$ is the mixture coefficient for cluster c_g. We generalize the cluster conditional density $p(x|c_g, \mathcal{D})$ in (1) by considering soft cluster participation, i.e each data point x_i contributes $q_i^g \in [0, 1]$ to the kernel density estimate of the cluster c_g.

$$p_g(x|c_g, \mathcal{D}) = \sum_{i=1}^{n} q_i^g \kappa(x_i, x), \text{where} \sum_{i=1}^{n} q_i^g = 1. \tag{3}$$

We refer to $q^g = (q_1^g, \ldots, q_n^g)$ as the *profile vector* for cluster c_g, and $Q = (q^1, \ldots, q^G)$ as the *profile matrix*. The objective of our clustering model is to learn the profile matrix Q for data set $\mathcal{D}$. We emphasize that due to the normalization step, i.e., $\sum_{j=1}^{n} q_j^g = 1$, q_j^g can no longer be interpreted as the probability of assigning x_j to cluster c_g. Instead, it only indicates the relative importance of x_j to the density function for cluster c_g. The density function in (3) is also referred to as the density estimate in “dual form” [11].

2.2 Estimation of Profile Matrix Q

To estimate the profile matrix Q, we follow the idea of maximum likelihood, i.e., find the matrix Q by solving the optimization problem $\max_Q \sum_{i=1}^n \log p(x_i|\mathcal{D})$. One major problem with this approach is that, when estimating $p(x_i|\mathcal{D})$, x_i is already an observed data point in $\mathcal{D}$ that is used to construct the density function $P(x_i|\mathcal{D})$. As a result, simply maximizing the likelihood of data may lead to an overestimation of the parameter Q, a problem that is often referred to as overfitting in machine learning [12]. We resolve this problem by replacing $p(x_i|\mathcal{D})$ with its leave-one-out (LOO) estimate [13].

Let $p_i(x_i|c_g, \mathcal{D}_{-i})$ be the LOO conditional probability for each held out sample x_i, conditioned on the clusters and the rest of the data:

$$p_i(x_i|c_g, \mathcal{D}_{-i}) = \frac{1}{\sum_{j=1}^n (1-\delta_{j,i}) q_j^g} \sum_{j=1}^n (1-\delta_{j,i}) q_j^g K_{i,j}, \tag{4}$$

where $D_{-i} = \mathcal{D} \backslash \{x_i\}$ denotes the subset of $\mathcal{D}$ that excludes sample x_i. Using the LOO cluster conditional probability $p_i(x_i|c_g, \mathcal{D}_{-i})$, we further define the LOO unconditional (on cluster) density for each held out sample x_i as follows:

$$p_i(x_i|\mathcal{D}_{-i}) = \sum_{g=1}^G \gamma_i^g p_i(x_i|c_g, D_{-i}). \tag{5}$$

where $\gamma_i^g = P(c_g|\mathcal{D}_{-i})$, and $\sum_g \gamma_i^g = 1, \forall i = 1, \dots, n$. Note that unlike the mixture model in (2) where the same set of mixture coefficients $\{\pi_g\}_{g=1}^G$ is used for any x_i, the mixture coefficients $\{\gamma_i^g\}_{g=1}^G$ depend on sample x_i, due to the leave-one-out estimation. We denote by $\gamma_i = (\gamma_i^1, \cdots, \gamma_i^G)$ and $\mathit{\Gamma} = (\gamma_1, \dots, \gamma_n)^\top \in \mathbb{R}_+^{n \times G}$.

To improve the robustness of estimation, we introduce a Gaussian prior for profile matrix Q, i.e.,

$$p(Q) \propto \exp\left(-\lambda \sum_i \sum_g [q_i^g]^2\right), \tag{6}$$

where λ is a hyperparameter that will be determined empirically. For notational convenience, we set $K_{i,i} = 0$ in Eq (4). Now, using the condition $\sum_{i=1}^n q_i^g = 1$, the LOO log-likelihood of data, denoted by $\ell_{LOO}(\mathcal{D}; Q, \mathit{\Gamma})$, can be expressed as follows

$$\begin{aligned} \ell_{LOO}(\mathcal{D}; Q, \mathit{\Gamma}) &= \log p(Q) + \sum_{i=1}^n \log p_i(x_i|\mathcal{D}_{-i}) \\ &= -\lambda \sum_{i=1}^n \sum_{g=1}^G (q_i^g)^2 + \sum_{i=1}^n \log\left(\sum_g \gamma_i^g \frac{\sum_{j=1}^n K_{i,j} q_j^g}{1 - q_i^g}\right). \end{aligned} \tag{7}$$

The parameters in the above simplified model are γ_i^g and q_i^g, for $i = 1, \cdots, n$ and $g = 1, \cdots, G$. They are estimated by maximizing the LOO log-likelihood $\ell_{LOO}(\mathcal{D}; Q, \varGamma)$. The optimal values of Q and $\varGamma$ can be obtained by solving the following optimization problem:

$$\{Q^*, \varGamma^*\} = \arg\max_{Q,\varGamma} \ell_{LOO}(\mathcal{D}; Q, \varGamma) \tag{8}$$

The optimization procedure is described in the following section.

2.3 Optimization Methodology

To determine the optimal values of $\varGamma$ and Q that maximize the log-likelihood in Eq (8), we apply an alternating optimization strategy [14]. At each iteration, we first optimize $\varGamma$ with fixed Q, and then optimize Q with fixed $\varGamma$, as summarized below. For a fixed Q, the LOO log-likelihood of a sample x_i is maximized when

$$\gamma_i^g = \delta(g, \arg\max_{g'} p_i(x_i | c_{g'}, \mathcal{D}_{-i})). \tag{9}$$

The variable γ_i^g is closely related to the posterior distribution $\Pr(c_g|x_i)$, and therefore can be interpreted as the cluster label of the i-th sample, i.e., $\gamma_i^g = 1$ if $x_i \in c_g$ and 0, otherwise.

Algorithm 1. $[Q, \varGamma]$ = NonParametricMixtureFit$(\mathcal{D}, G, \lambda, \sigma)$

```
Input: Dataset $\mathcal{D}$, no. of clusters $G$, parameters $\lambda$ and $\sigma$
Output: Cluster labels $\varGamma$ and the profile matrix $Q$
Compute the kernel matrix $K$ for the points in $\mathcal{D}$ with bandwidth $\sigma$. Normalize $K$
  such that $\sum_j K_{ij} = 1$.
Set the iteration $t \leftarrow 0$.
Initialize $Q^{(t)} \leftarrow Q_0$, such that $Q_0 \succcurlyeq 0$, $Q_0^T 1_n = 1_G$.
repeat
    $t \leftarrow t + 1$;
    Compute the $\gamma_i^g$ using Eq (9)
    By fixing the values of $\gamma_i^g$, obtain $Q^{(t)}$ by minimizing Eq (7).
    $\varDelta Q \leftarrow Q^{(t)} - Q^{(t-1)}$.
until $||\varDelta Q||_2^2 \leq \epsilon$, ($\epsilon$ is pre-set to a desired precision)
return $Q, \varGamma$
```

It is difficult to directly optimize the log-likelihood in Eq (7) with respect to Q. Therefore, we minimize a convex variational upper bound on the negative log-likelihood for efficient inference. At each iteration, we maintain a touch point between the bound and the negative log-likelihood function, which guarantees convergence to at least a local minima [15]. The procedure for finding Q and $\varGamma$ that maximize the log-likelihood in Eq (7) is summarized in Algorithm 1. Upon convergence, the value of γ_i determines the cluster label for x_i.

2.4 Implementation Details

Normalization is one of the key issues in kernel density estimation. Conventionally, the kernel function is normalized over the entire domain of the data, $\kappa_\sigma(\mathbf{x}) = (\pi\sigma)^{-d} \exp\left(-||\mathbf{x}||^2/2\sigma^2\right)$. However, the σ^{-d} term may be close to 0 ($\sigma < 1$) or be very large ($\sigma > 1$). This may cause serious numerical problems in density estimation for high-dimensional data (large values of d) with small sample size. To overcome this problem, we normalize the kernel matrix such that each row sums to 1, i.e. $\sum_j K_{i,j} = 1$. This nullifies the effect of dimension on the estimation process, and therefore is useful in handling sparse datasets.

The heuristic used in spectral clustering [16] to select the value for σ is also effective in estimating kernel width. Empirical results show that the clustering performance is not very sensitive to the choice of the kernel width σ. The parameter λ also is not very critical and is chosen to be sufficiently small; in all of our experiments we choose $\lambda = 10^{-4}$, which results in mild smoothing of the q_i^g values, and avoids any numerical instability in the algorithm due to the logarithm. The number of variables to be solved for is of $O(nG)$, similar to that of spectral clustering. On the other hand, Gaussian mixture models solve for $O(d^2)$ number of variables which is large, especially for high-dimensional sparse datasets (specifically when $(n+1)G < \left(dG + \frac{d(d+1)}{2}\right)$, as shown in Table 1).

3 Results and Discussion

The proposed non-parametric mixture fitting algorithm is evaluated on text datasets derived from the 20-newsgroups[3] dataset [17].

3.1 Baseline Methods

The proposed non-parametric mixture algorithm is compared with three classes of well known clustering algorithms: (a) K-means and Gaussian mixture model (GMM) with diagonal and full covariance matrices, (b) one kernel-based algorithm: NJW spectral clustering [19], and (c) three non-parametric hierarchical clustering algorithms: Single Link, Complete Link and Average Link. For (a) and (c), we use the implementations from the Matlab's Statistics Toolbox. For the linkage based methods, the number of clusters is externally specified. We chose the state-of-the-art spectral clustering algorithm implementation based on [19]. Each algorithm is run 10 times and the mean performance value is reported in Table 1, with the best performance shown in bold face. Comparison with Meanshift, or related algorithms is difficult as the datasets are high-dimensional and further, it is not possible to specify the number of clusters in these algorithms. Since the number of dimensions is greater than the number of data points, GMM is not succesful for this data.

At each run, the proposed algorithm, K-means and Spectral clustering were initialized with 5 different starting points; only the best performance is reported.

[3] http://people.csail.mit.edu/jrennie/20Newsgroups/

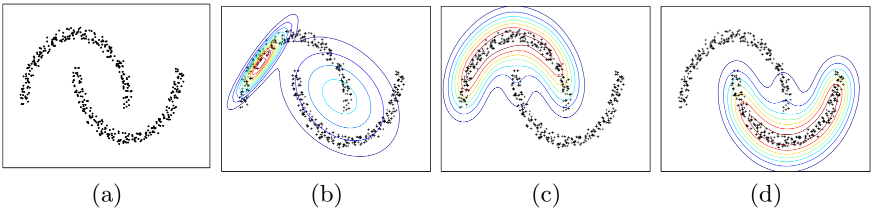


Fig. 1. Illustration of the non-parametric mixture approach and Gaussian mixture model (GMM) on the "two-moon" dataset. (a) Input data with two clusters. (b) Gaussian mixture model with two components. (c) and (d) the iso-contour plot of non-parameteric estimates of the class conditional densities for each cluster. The warmer the color, the higher the probability.

Due to the space limitation, we only show the best performance among the three hierarchical linkage based algorithms, without specifying which algorithm achieved it.

3.2 Synthetic Datasets

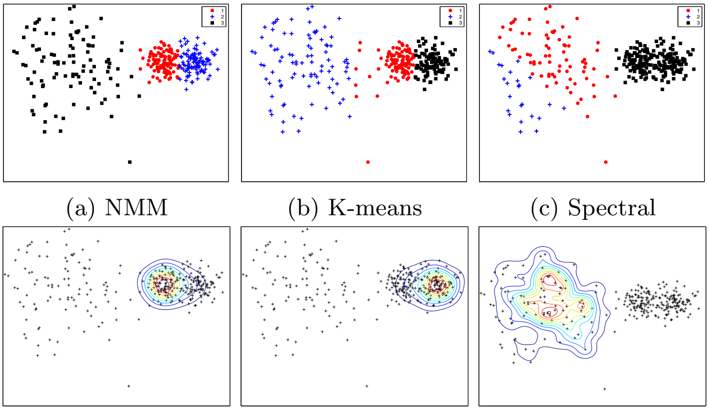
The proposed algorithm aims at identifying clusters of arbitrary shapes, while estimating their conditional density. Figure 1 illustrates the performance of NMM on a dataset not suitable for GMM. Figure 1(a) shows the input data. Figure 1(b) is shown to contrast the proposed non-parametric mixture approach against the parametric Gaussian mixture model (GMM) with the number of mixture components set to two. Figures 1(c) and (d) show the class conditional densities for each of the two clusters. The proposed algorithm is able to recover the underlying clusters, as well as estimate the associated conditional densities, which is not possible for GMM as shown in Figure 1(b).

Figure 2 illustrates the performance of the proposed algorithm on a dataset that is known to be difficult for spectral clustering [18]. Both K-means and spectral clustering fail to recover the clusters due to the difference in the variance of the spherical clusters. The proposed algorithm however, is purely local, in that the cluster label of a point is affected only by the cluster labels of neighboring points. The clusters, therefore, are recovered nearly perfectly by the proposed algorithm as shown in Figure 2(a) and the cluster conditional densities are shown in Figures 2(d)-(f).

3.3 Text Datasets

We use eight high dimensional text datasets to show the efficacy of the algorithm. These datasets are popularly used in document clustering [20].

Table 1 shows that the proposed non-parametric mixture (NMM) algorithm performs significantly better (paired t-test, 95% confidence) than the other clustering methods on all the high dimensional text datasets, except for



(a) NMM (b) K-means (c) Spectral

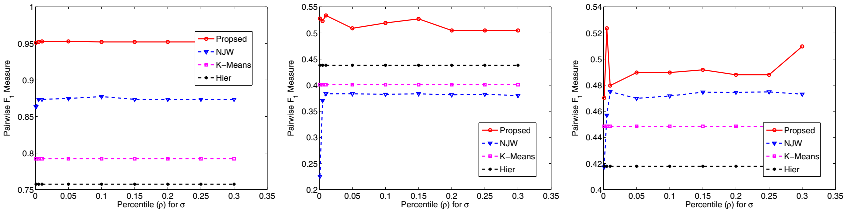
(d) (e) (f)

Fig. 2. Illustration of the non-parametric mixture approach, K-means and spectral clustering on the example dataset from [18]. Input data contains 100 points each from three spherical two-dimensional Gaussian clusters with means (0,0), (6,0) and (8,0) and variances $4I_2$,$0.4I_2$ and $0.4I_2$ respectively. Spectral clustering and NMM use $\sigma = 0.95$. (a) NMM (b) K-means (c) Spectral clustering. Plots (d)-(f) show the cluster-conditional densities estimated by the proposed NMM.

Table 1. Mean pairwise F_1 value for different clustering algorithms over 10 runs of each algorithm on eight high-dimensional text datasets. The kernel width is chosen as the 5^{th} percentile of the pairwise Euclidean distances for Kernel based algorithms. The best performance for each dataset is shown in bold. The name of the dataset, number of samples (n), dimensions (d), and the number of target clusters (G) are shown in the first 4 columns, respectively. The last column shows the best F_1 value achieved by Single (S), Complete (C) and Average (A) link algorithms.

				Proposed	K-means	NJW-Spec	Linkage
Dataset	n	d	G				max(S,C,A)
cmu-different-1000	2975	7657	3	**95.86**	87.74	94.37	40.31
cmu-similar-1000	2789	6665	3	**67.04**	49.86	45.16	37.28
cmu-same-1000	2906	4248	3	**73.79**	49.40	48.04	30.01
cmu-different-100	300	3251	3	**95.27**	79.22	87.47	75.74
cmu-similar-100	288	3225	3	**50.89**	40.10	38.35	43.82
cmu-same-100	295	1864	3	**48.97**	44.85	46.99	41.79
cmu-classic300	300	2372	3	85.32	**86.32**	86.02	80.61
cmu-classic400	400	2897	3	**61.26**	60.13	51.01	53.31

`cmu-classic-300`, where its performance is slightly inferior to K-means. Since the datasets are high-dimensional, and non-spherical, the proposed approach outperforms K-means on most of the datasets. Spectral clustering considers only the top $G-1$ eigenvectors for clustering a dataset into G clusters; the superior performance of the proposed NMM can be attributed to its utilization of the complete



(a) Different-100 (b) Similar-100 (c) Same-100

Fig. 3. Performance of the non-parametric mixture model on three text datasets, with varying value of the percentile (ρ) for choosing the kernel bandwidth (σ). The proposed algorithm is compared with NJW (Spectral clustering), K-means and the best of three linkage based methods.

kernel matrix without discarding any portion of it. These datasets could not be clustered by GMM (Gaussian mixture models) since they are prone to numerical estimation problems when the number of dimensions is larger than the number of samples.

3.4 Sensitivity to Parameters

There are two parameters in the non-parametric mixture clustering algorithm: the regularizer weight λ and the kernel width σ. The parameter σ is set to the ρ^{th} percentile of the pairwise Euclidean distances among the data points. A useful range for ρ is 5-10%, as suggested in [16]. Figure 3 shows the performance of the proposed algorithm in comparison to K-means, spectral clustering and hierarchical clustering on three text datasets. These plots show that there exists a wide range of kernel bandwidth values for which the proposed algorithm performs significantly better than the competing methods. For some datasets (e.g., Different-100 and Classic-400), the algorithm is more stable compared to that of other datasets. We observed that the algorithm is not sensitive to the value of λ, over the range $(10^{-4}, 10^{4})$. While the performance is the same for almost all the values of λ, the parameter λ does play a role in determining the sparsity of the profile matrix. As λ increases, the profile of data points between the clusters tends to get smoother. The key role of λ is to provide numerical stability to the algorithm.

4 Conclusions and Future Work

We have proposed a non-parametric mixture model for data clustering. It is a probablistic model that clusters the data by fitting a kernel density estimate to each cluster. Experimental results show that the non-parametric mixture model based clustering outperforms some of the well known clustering algorithms on the task of document clustering, which can be characterized as high dimensional sparse data. The non-parametric mixture model opens up a wide range of possible theoretical analysis related to clustering, which is a part of our ongoing work.

Automatic kernel bandwidth selection, scalability of the algorithm and application to other sparse data domains (e.g., bioinformatics) are possible extensions.

References

1. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recognition Letters 31, 651–666 (2010)
2. McLachlan, G.L., Peel, D.: Finite Mixture Models. Wiley, Chichester (2000)
3. Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of finite mixture models. TPAMI 24, 381–396 (2002)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
5. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs (1988)
6. Jarvis, R.A., Patrick, E.A.: Clustering using a similarity measure based on shared near neighbors. IEEE Transactions on Computers 22 (1973)
7. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. KDD, pp. 226–231 (1996)
8. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 603–619 (2002)
9. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
10. Andreetto, M., Zelnik Manor, L., Perona, P.: Non-parametric probabilistic image segmentation. In: Proceedings of the ICCV, pp. 1–8 (2007)
11. Shawe-Taylor, J., Dolia, A.N.: A framework for probability density estimation. In: Proc. AISTATS, pp. 468–475 (2007)
12. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley, Chichester (2001)
13. Wand, M.P., Jones, M.C.: Kernel Smoothing (Monographs on Statistics and Applied Probability, December 1994. Chapman & Hall/CRC, Boca Raton (1994)
14. Csiszar, I., Tusnady, G.: Information geometry and alternating minimization procedures. Statistics and Decision (1984)
15. Jaakkola, T.S.: Tutorial on variational approximation methods. In: Advanced Mean Field Methods: Theory and Practice, pp. 129–159 (2000)
16. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 888–905 (2000)
17. Slonim, N., Tishby, N.: Agglomerative information bottleneck. In: Advances in NIPS (2000)
18. Nadler, B., Galun, M.: Fundamental limitations of spectral clustering. In: NIPS 19, Citeseer, pp. 1017–1025 (2007)
19. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in NIPS, pp. 849–856. MIT Press, Cambridge (2001)
20. Banerjee, A., Langford, J.: An objective evaluation criterion for clustering. In: Proceedings of the KDD, pp. 515–520 (2004)

A Probabilistic Approach to Spectral Unmixing

Cong Phuoc Huynh[1] and Antonio Robles-Kelly[1,2]

[1] School of Engineering, Australian National University, Canberra ACT 0200, Australia
[2] National ICT Australia (NICTA)*, Locked Bag 8001, Canberra ACT 2601, Australia

Abstract. In this paper, we present a statistical approach to spectral unmixing with unknown endmember spectra and unknown illuminant power spectrum. The method presented here is quite general in nature, being applicable to settings in which sub-pixel information is required. The method is formulated as a simultaneous process of illuminant power spectrum prediction and basis material reflectance decomposition via a statistical approach based upon deterministic annealing and the maximum entropy principle. As a result, the method presented here is related to soft clustering tasks with a strategy for avoiding local minima. Furthermore, the final endmembers depend on the similarity between pixel reflectance spectra. Hence, the method does not require a preset number of material clusters or spectral signatures as input. We show the utility of our method on trichromatic and hyperspectral imagery and compare our results to those yielded by alternatives elsewhere in the literature.

1 Introduction

Spectral unmixing is commonly stated as the problem of decomposing an input spectral signal into relative portions of known spectra of endmembers. The endmembers can be any man-made or naturally occurring materials such as water, metals, etc. The input data varies in many forms, such as radiance or reflectance spectra, or hyperspectral images. The problem of unmixing applies to all those cases where a capability to provide subpixel detail is needed, such as geosciences, food quality assessment and process control. Moreover, unmixing can be viewed as a pattern recognition task related to soft-clustering with known or unknown endmembers.

Current unmixing methods assume availability of the endmember spectra [1]. This yields a setting in which cumbersome labelling of the endmember data is effected through expert intervention. Added to the complexity of endmember labeling is the fact that, often, illumination is a confounding factor in determining the intrinsic surface material reflectance. As a result, in general, unmixing can be viewed as a dual challenge. Firstly, endmembers shall, in case of necessity, be identified automatically. Secondly, reflectance has to be recovered devoid of illumination and scene geometry. The former of these can be viewed as an instance of blind-source or unsupervised clustering techniques. The latter is a photometric invariance problem.

* NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 344–353, 2010.

Spectral unmixing with automatic endmember extraction is closely related to the simultaneous estimation of illuminant power spectrum and material reflectance. Many of the methods elsewhere in the literature hinge on the notion that the simultaneous recovery of the reflectance and illuminant power spectrum requires an inference process driven by statistical techniques. In [2], Stainvas and Lowe proposed a Markov Random Field to separate illumination from reflectance from the input images. On the other hand, the physics-based approach in [3] for image colour understanding alternately forms hypotheses of colour clusters from local image data and verifies whether these hypotheses fit the input image. More recently, in [4], Tajima performed an imperfect segmentation of the object colours in images by recursively subdividing the scene's color space through the use of Principal Component Analysis. Li *et al.* [5] proposed an energy minimisation approach to estimating both the illumination and reflectance from images.

Note that the methods above aim at tackling either the colour constancy or the image segmentation problem and do not intend to recover subpixel information. Here, we present an approach to spectral unmixing with unknown lighting conditions and unknown endmember signatures. Unlike previous approaches related to the field of spectral unmixing [6] and photometric invariants [7], our method does not assume known power spectrum or colour of the illuminant. We adopt a probabilistic treatment of the problem which allows for a soft clustering operation on the pixel reflectance spectra. Thus, the method is quite general in the sense that it is applicable to any number of colour channels assuming no prior knowledge of the illumination condition as well as the surface reflectance.

The paper is organised as follows. In section 2, we cast the problem of simultaneous spectral unmixing and illumination recovery in a probabilistic framework based on the dichromatic reflection model [8] and the maximum entropy principle [9]. In this section, we also describe a deterministic annealing approach to solve the problem for a single spectral radiance image with unknown lighting condition and endmembers. Section 3 illustrates the utility of our method on real-world multispectral and trichromatic images.

2 Probabilistic Formulation

This section provides a probabilistic formulation of the problem of spectral unmixing on multispectral imagery. To commence, we pose the problem in the general case as a minimisation one governed by the interaction between image pixels and endmembers. This yields a general formulation for spectral radiance images with no prior knowledge of the lighting condition. By making use of the dichromatic model [8] and the maximum entropy principle [9], our integrated spectral unmixing and illumination estimation algorithm involves three interleaved steps

1. From the input spectral radiance image, find an optimal set of dichromatic hyperplanes representing the current endmembers and material association probabilities per pixel.
2. Estimate the illumination power spectrum making use of the least-squares intersection between the dichromatic hyperplanes.
3. Recover the endmembers from the reflectance image. The reflectance image is obtained via the normalisation of the input radiance image with respect to the current estimate of the illumination power.

2.1 General Unmixing Formulation

Our general problem is formulated as follows. Given an input multispectral image $\mathcal{I}$, we aim to recover the basis material reflectance, *i.e.* endmember signatures, as well as their relative proportions at each pixel in a single trichromatic or multi-band image. Let $\mathcal{M}$ be the set of unknown endmembers in the scene under study. Here we take a probabilistic viewpoint on the problem by equating material composition to the notion of association probability relating the input signal at a pixel u to a basis material $M \in \mathcal{M}$, which we denote as $p(M|u)$. The problem also involves a definition of the affinity between the input signal at a pixel u and a basis material M, which we denote as $d(u, M)$. The spectral unmixing statement is then cast as minimising the expected affinity between the given image $\mathcal{I}$ and the endmembers.

Thus, we aim to find a distribution of material association probabilities $\mathcal{P} = \{p(M|u)|M \in \mathcal{M}, u \in \mathcal{I}\}$ that minimises the total expected pixel-material affinity

$$C_{Total} = \sum_{u \in \mathcal{I}} \sum_{M \in \mathcal{M}} p(M|u) d(u, M) \tag{1}$$

subject to the law of total probability $\sum_{M \in \mathcal{M}} p(M|u) = 1 \forall u \in \mathcal{I}$.

Note that the formulation in Equation 1 is reminiscent of a soft clustering problem. Here we aim to find the optimal association of image pixels with a set of materials which minimises the cost function. Since the above formulation often favours single-material composition per pixel as each pixel is finally associated with its closest material with probability one, we restate the problem as that of finding a distribution of material association $\mathcal{P}$ that minimises the above cost function subject to the maximum entropy criterion [10]. The justification of the additional constraint originates from the maximum entropy principle [9], which states that amongst all the probability distributions that satisfy a set of constraints, the one with the maximum entropy is preferred.

To apply this principle to our problem, let us fix the expected affinity level at hand. While several distributions of material association satisfy this level of expected affinity, choosing non-maximal entropy distributions would imply making rather restrictive assumptions on the problem. As a result, only the one with the maximum entropy shall require no further constraints.

By making use of the entropy

$$H(\mathcal{P}) = -\sum_{u \in \mathcal{I}} \sum_{M \in \mathcal{M}} p(M|u) \log p(M|u) \tag{2}$$

to quantify the level of uncertainty of the material association distribution we reformulate the expected material affinity as

$$C_{Entropy} = C_{Total} - \mathcal{L} \tag{3}$$

where

$$\mathcal{L} = TH(\mathcal{P}) + \sum_{u \in \mathcal{I}} \alpha(u) \left(\sum_{M \in \mathcal{M}} p(M|u) - 1 \right) \tag{4}$$

in which $T \geq 0$ and $\alpha(u)$ are Lagrange multipliers. Note that $T \geq 0$ weighs the level of randomness of the material association probabilities whereas $\alpha(u)$ enforces the total probability constraint for every image pixel u.

2.2 Illumination Spectrum Estimation

With the expression in Equation 3 at hand, we now proceed to integrate the dichromatic reflection theory introduced by Shafer [8] into the general problem formulation in Section 2.1.

The dichromatic model for a scene illuminated by a single illumination source relates the observed image radiance to the illuminant power spectrum and material reflectance. According to the dichromatic model, the observed colour or spectral radiance power at a point in the scene is a linear combination of the body reflection and interface reflection. The former component is affected by the material reflectance while the latter one is purely governed by the illuminant power spectrum. Therefore, the spectral radiance spectrum at a surface point composed of a single material belongs to a two-dimensional linear subspace spanned by the illuminant spectrum and the diffuse radiance spectrum of the material. We refer to this subspace as the dichromatic hyperplane.

In the most general case where each scene location is made of a mixture of materials, the pixel radiance spectrum does not necessarily lie in any of the dichromatic hyperplanes corresponding to the endmembers. Therefore, it is natural to quantify the notion of pixel-material affinity as the distances between the pixel radiance spectrum and the dichromatic planes corresponding to the endmembers. A zero-distance means purity in terms of material composition. The further the distance, the lower the proportion of the basis material or endmember.

To define the spectral unmixing problem using the dichromatic reflection model we require some formalism. Let us consider a multispectral imaging sensor that samples the spectral dimension of incoming light at wavelengths $\lambda_1, \ldots \lambda_K$. The input radiance at pixel u and wavelength λ_i is denoted as $I(u, \lambda_i)$ and the spectral component of the illumination power at wavelength λ_i is $L(\lambda_i)$. For brevity, we adopt thes vectorial notations of the illumination spectrum, the input radiance spectrum at each pixel u and the material reflectance spectrum of material M, which we denote $\mathbf{L}$, $\mathbf{I}(u)$ and $\mathbf{S}(M)$, respectively.

As discussed above, we characterise the combination of illumination power spectrum and material basis in a scene as a set of basis dichromatic planes, each of which captures all the possible radiance spectra reflected from a point made of a single basis material. Each of these two-dimensional planes can be further specified by two basis vectors. Note that the choice of basis vectors is, in general, arbitrary. Let us denote the dichromatic hyperplane for the endmember material M as $\mathcal{Q}(M)$, with two basis column-vectors $\mathbf{z}_1(M), \mathbf{z}_2(M)$.

With these ingredients, the affinity between the pixel u and the basis material M is quantified as the orthogonal distance between a K-dimensional point representing its radiance spectrum $\mathbf{I}(u)$ and the hyperplane $\mathcal{Q}(M)$. Since the linear projection matrix onto $\mathcal{Q}(M)$ is defined as $Q(M) = A(M)(A(M)^T A(M))^{-1} A(M)^T$, where $A(M) = [\mathbf{z}_1(M), \mathbf{z}_2(M)]$, the affinity distance is therefore defined as the squared L^2-norm of the hyperplane $d(u, M) = ||\mathbf{I}(u) - Q(M)\mathbf{I}(u)||^2$

The unmixing problem on spectral radiance images becomes that of seeking for an optimal set of linear projection matrices $\{Q(M), M \in \mathcal{M}\}$ corresponding to the endmembers and the material association probabilities $p(M|u)$ that minimise the following function

$$C_{Light} = \sum_{u \in \mathcal{I}} \sum_{M \in \mathcal{M}} p(M|u) \|\mathbf{I}(u) - Q(M)\mathbf{I}(u)\|^2 - \mathcal{L} \tag{5}$$

For the recovery of the optimal dichromatic hyperplanes, we fix the material association probability distribution. In this situation, we recast the problem as that of finding the optimal basis vectors $\mathbf{z}_1(M), \mathbf{z}_2(M)$ for each endmember M so as to minimise the expected material affinity given by the first term on the right-hand side of Equation 5

$$\sum_{u \in \mathcal{I}} p(M|u) \|\mathbf{I}(u) - Q(M)\mathbf{I}(u)\|^2 = \sum_{u \in \mathcal{I}} \| \sqrt{p(M|u)}\mathbf{I}(u) - A(M)b(u, M)\|^2 \tag{6}$$

where $A(M) = [\mathbf{z}_1(M), \mathbf{z}_2(M)]$ and $b(u, M) \triangleq \sqrt{p(M|u)}(A(M)^T A(M))^{-1} A(M)^T \mathbf{I}(u)$ is a two-element column vector.

We note that the right-hand side of Equation 6 is the Frobenius norm of the matrix $\mathbb{I} - \mathbb{J}$, where $u_1, u_2, \ldots u_N$ are all the image pixels and

$$\mathbb{I} = \left[\sqrt{p(M|u_1)}\mathbf{I}(u_1), \sqrt{p(M|u_2)}\mathbf{I}(u_2), \ldots, \sqrt{p(M|u_N)}\mathbf{I}(u_N) \right]$$
$$\mathbb{J} = A(M)[b(u_1, M)b(u_2, M), \ldots, b(u_N, M)]$$

Since $rank(\mathbb{J}) \leq rank(A(M)) = 2$, the problem above amounts to finding a matrix $\mathbb{J}$ with rank at most 2 that best approximates the known matrix $\mathbb{I}$. We achieve this via a Singular Value Decomposition operation such that $\mathbb{I} = U \Sigma V$, where U and V are the left and right singular matrices of $\mathbb{I}$ and Σ is a diagonal matrix containing its singular values. The solution to this problem is then given by $\mathbb{J} = U \Sigma^* V$, in which the only non-zero singular values of Σ^* are the two leading singular values of Σ. The vectors $\mathbf{z}_1(M), \mathbf{z}_2(M)$ correspond to the two leading eigenvectors of $\mathbb{I}$, *i.e.* those corresponding to the non-zero singular values in Σ^*. With the basis vectors $\mathbf{z}_1(M), \mathbf{z}_2(M)$ at hand, we can estimate the illumination power spectrum as a least-squares intersection between dichromatic hyperplanes making use of the algorithm in [11].

2.3 Endmembers from Image Reflectance

With the illuminant power spectrum at hand, we can obtain the reflectance image from the input radiance image by illumination normalisation. Let the reflectance spectrum at each image pixel u be a wavelength-indexed vector $\mathbf{R}(u) = [R(u, \lambda_1), \ldots, R(u, \lambda_K)]^T$, where $R(u, \lambda_1)$ is given by $R(u, \lambda) = \frac{I(u,\lambda)}{L(\lambda)}$. Note that the affinity distance between a pixel reflectance spectrum and a material reflectance spectrum can be defined based on their Euclidean angle. Mathematically, the distance is given as follows

$$d(u, M) = 1 - \frac{\tilde{\mathbf{R}}(u)^T \mathbf{S}(M)}{\|\mathbf{S}(M)\|} \tag{7}$$

where $\tilde{\mathbf{R}}(u)$ has been obtained by normalising $\mathbf{R}(u)$ to unit L^2-norm.

With this affinity distance, our unmixing problem becomes that to find a set of basis material reflectance spectra and a distribution of material association probabilities

$p(M|u)$ for each pixel u and material M that minimise the following cost function

$$C_{Reflectance} = \sum_{u \in \mathcal{I}} \sum_{M \in \mathcal{M}} p(M|u) \left(1 - \frac{\tilde{\mathbf{R}}(u)^T \mathbf{S}(M)}{\|\mathbf{S}(M)\|} \right) - \mathcal{L} \tag{8}$$

We now derive the optimal set of endmember spectra so as to minimise the cost function $C_{Reflectance}$ in Equation 8. To minimise the cost function, we compute the derivatives of $C_{Reflectance}$ with respect to the endmember reflectance $\mathbf{S}(M)$ to yield

$$\frac{\partial C_{Reflectance}}{\partial \mathbf{S}(M)} = - \sum_{u \in \mathcal{I}} p(M|u) \frac{\|\mathbf{S}(M)\|^2 \tilde{\mathbf{R}}(u) - (\tilde{\mathbf{R}}(u)^T \mathbf{S}(M))\mathbf{S}(M)}{\|\mathbf{S}(M)\|^3}$$

Setting this derivative to zero, we obtain

$$\mathbf{S}(M) \propto \sum_{u \in \mathcal{I}} p(M|u) \tilde{\mathbf{R}}(u) \tag{9}$$

2.4 Material Association Probability Recovery

Note that, in the cost functions associated to the steps above, we require the material association probability $p(M|u)$ to be at hand. In this section, we describe how $p(M|u)$ can be recovered via deterministic annealing. A major advantage of the deterministic annealing approach is that it mitigates attraction to local minima. In addition, deterministic annealing converges faster than stochastic or simulated annealing [12].

The deterministic annealing approach casts the Lagrangian multiplier T in the role of the system temperature in an analogous annealing process used in statistical physics. Initially, the whole process starts at a high temperature. At each temperature, the system eventually converges to a thermal equilibrium. After reaching this state, the system experiences a "phase transition" as the temperature is lowered. The optimal parameters corresponding to the equilibrium state are tracked through such phase transitions. At zero temperature, we can directly minimise the expected pixel-material affinity $C_{Entropy}$ to obtain the final material association probabilities $\mathcal{P}$ and the endmember reflectance.

The recovery of the endmembers in step 3 described in Section 2 is, hence, somewhat similar to a soft-clustering process. At the beginning, this process is initialised at a high temperature with a single endmember by assuming all the image pixels are made of the same material. As the temperature is lowered, the set of endmembers grows. This, in essence, constitutes several "phase transitions", at which new endmembers arise from the existing ones. This phenomenon is due to the discrepancy in the affinity between the image pixels and the existing endmembers.

At each phase of the annealing process, where the temperature T is kept constant, the algorithm proceeds as two interleaved minimisation steps at each iteration so as to arrive at an equilibrium state. These two minimisation steps are performed alternately with respect to the material association probabilities $\mathcal{P}$ and the endmembers as captured by the pixel-material affinity function $d(u, M)$. For the recovery of the material association probabilities, we fix the endmember set and seek for the probability distribution which minimises the cost function $C_{Entropy}$ in Equation 3. This is achieved by setting the partial derivative $\frac{\partial C_{Entropy}}{\partial p(M|u)} = d(u,M) + T\log p(M|u) + T - \alpha(u)$ to zero. We obtain

$$p(M|u) = \exp\left(\frac{-d(u,M)}{T} + \frac{\alpha(u)}{T} - 1\right) \propto \exp\left(\frac{-d(u,M)}{T}\right) \forall M, u \qquad (10)$$

Since $\sum_{M \in \mathcal{M}} p(M|u) = 1$, it can be shown that the optimal material association probability distribution for a fixed endmember set $\mathcal{M}$ is given by the Gibbs distribution

$$p(M|u) = \frac{\exp\left(\frac{-d(u,M)}{T}\right)}{\sum_{M' \in \mathcal{M}} \exp\left(\frac{-d(u,M')}{T}\right)} \qquad (11)$$

3 Experiments

In this section, we provide validation results of our algorithm on real-world multispectral and trichromatic imagery. To perform the validation task, we used two datasets. The first of these comprises 321 Mondrian and specular images from the Simon Fraser University database [13]. In addition, we have acquired an image database of 51 human subjects, each captured under one of 10 light sources with varying directions and spectral power. The multispectral imagery has been acquired using a pair of benchtop hyperspectral cameras. Each of which is equipped with Liquid Crystal Tunable Filters which are capable of resolving up to $10nm$ in both the visible (430–$720nm$) and near infrared (650–$990nm$) wavelength ranges. The ground truth illuminant spectra have been measured using a white reference target, *i.e.* a Labsphere Spectralon.

On both the multispectral and trichromatic image databases, we configure the deterministic annealing process with the initial and terminal temperatures of $T_{max} = 0.02$ and $T_{min} = 0.00025$. We employ an exponential decay function as the cooling schedule with a decay rate of 0.8. The maximum number of endmembers in each image is set to 20.

First, we provide results on the illuminant recovered by our algorithm. Since our method is an unmixing one which delivers at output the illuminant power spectrum and endmember reflectance, the error on the recovered illuminant is an indirect measure of the efficacy of the algorithm. Here we present a quantitative comparison with Colour Constancy methods that can be applied to single images with no pre-processing steps and no prior statistics of image colours gathered from a large number of images. These alternatives include the Grey-World [14] and White-Patch hypotheses [15] which are special instances of the Shades of Grey method [16], and the Grey-Edge method [17]. The accuracy of the estimated multispectral illumination power and trichromatic illuminant colour is quantified as the Euclidean angle with respect to the ground truth.

In the fourth and fifth columns of Table 1, we present the trichromatic illuminant estimation results on the Simon Fraser University dataset [13]. These include the Mondrian and specular datasets with 8 and 16-bit dynamic ranges. Overall our method yields better results than the Grey-World method and is quite comparable to the White-Patch method. The other methods outperform ours but the difference in performance is less than two degrees. Recall that the illumination estimation method we employ is purely

Table 1. A comparison of the illumination recovery performance of our method with a number of alternatives. The angular error (in degrees) are shown for both multispectral and trichromatic image datasets.

Method	Multispectral Images		Trichromatic Images	
	Visible Range	Infrared Range	8-bit	16-bit
Our method	6.84 ± 3.92	4.19 ± 4.75	8.24 ± 8.73	7.99 ± 7.59
Grey-World [14]	8.44 ± 3.03	6.89 ± 2.23	9.75 ± 9.4	9.67 ± 9.25
White-Patch [15]	11.91 ± 8.02	8.53 ± 6.22	7.66 ± 6.92	7.70 ± 6.92
Shades of Grey (sixth-order norm) [16]	8.17 ± 4.61	5.16 ± 3.65	6.23 ± 6.50	6.27 ± 6.47
Grey-Edge (first-order norm) [17]	6.23 ± 2.06	2.21 ± 1.07	6.78 ± 4.37	6.82 ± 4.35
Grey-Edge (sixth-order norm) [17]	8.37 ± 7.77	6.45 ± 5.86	6.42 ± 5.82	6.45 ± 5.82

based on dichromatic patches [11], *i.e.* its stability and robustness improve with an increasing number of materials and a higher level of image specularity. Therefore, it is not surprising that the illumination estimates are severely affected by the large number of images in this trichromatic database that are either completely diffuse or consist of only a few materials.

In the second and third columns we show the accuracy of the illumination spectra recovered from the multispectral images. On the multispectral images, our method clearly outperforms all instances of the Shades of Grey method and the Grey-Edge method implemented with a sixth-order Minkowski norm, by a significant margin. Note that the Shades-of-Grey paradigm relies on the heuristics that the average scene colour is achromatic. Therefore the accuracy of these methods is somewhat limited by the degree of achromaticity of the average colours in the multispectral images. The Grey-Edge method with a first-order Minkowski norm performs better than our method mainly because of the abundance of edges and material boundaries in the multispectral images. However, it should be stressed that the former method does not recover endmembers and material composition.

Now we turn our attention to the performance of our algorithm for the spectral unmixing task on the multispectral image database. We quantify the accuracy of the endmember reflectance extracted by our method from the multispectral images as compared to the ground truth measurements. To acquire the ground-truth endmembers, we normalise each input radiance image by its ground-truth illumination spectrum and then apply a K-means algorithm on the reflectance image to produce 20 clusters of pixels, each made of the same material. The resulting cluster centroids are deemed to be the ground-truth endmember reflectance. As before, we have computed the Euclidean angles between the basis material reflectance recovered by our method and those recovered by K-means clustering. The mean angular differences are 8.56 degrees for the visible and 11.49 degrees for the infrared spectrum, which are comparable to those produced by the alternative Colour Constancy methods shown before.

Next, we compare the association probability maps recovered by our method and the Spectral Angle Mapper (SAM) [18]. As an input to this operation the SAM, the illumination spectra are assumed to be those recovered by the Grey-Edge method with the first-order Minkowski norm. Recall that this method is the only Colour Constancy

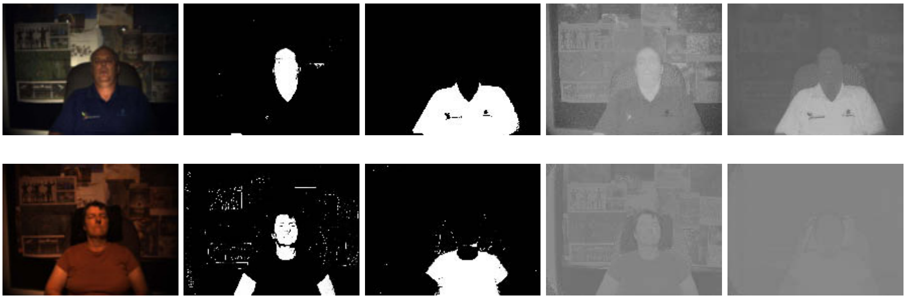


Fig. 1. Material maps estimated from a visible image (top row) and a near infrared image (bottom row). First column: the input image in pseudo colour. Second and third columns: the probability maps of skin and cloth produced by our method. Fourth and fifth columns: the probability maps of skin and cloth produced by the Spectral Angle Mapper(SAM).

method shown before that slightly outperforms our method in multispectral illumination spectrum recovery. For the SAM, the endmember reflectance spectra are those resulting from K-means clustering on the multispectral images.

In the second and third columns of Figure 1, we show the probability maps of skin and cloth materials recovered by our method. In the fourth and fifth columns, we show the probability maps recovered by the SAM. The two sample images shown have been captured under different illumination conditions, one of which in the visible and the other in the infrared spectrum. In the panels, the brightness of the probability maps is proportional to the association probability with the reference material. It is evident that our algorithm produces cleaner endmember maps than the SAM for both materials and spectral regions. In other words, our method correctly labels the skin and cloth regions as primarily composed of the respective ground truth endmember. We can attribute this to the ability of deterministic annealing in escaping from local minima. On the other hand, the SAM appears to assign a high proportion of non-primary materials to skin and cloth regions. In fact, the material maps in the fourth and fifth columns show a very weak distinction between the primary materials and the others in these regions. This symptom is not surprising since the SAM may not be able to determine the primary material in the case where a number of endmembers have nearly equal distances to the pixel reflectance spectrum.

4 Conclusions

We have introduced a probabilistic method for simultaneous spectral unmixing and illumination estimation provided no prior assumption on the lighting condition and the end-member spectra. We have formulated the unmixing task making use of the dichromatic reflection model [8] and the maximum entropy principle [9]. Moreover, we have used deterministic annealing as an optimisation method to improve convergence to the globally optimal solution. We have illustrated the utility of the method on hyperspectral and trichromatic imagery and compared our results against a number of alternatives.

References

1. Lennon, M., Mercier, G., Mouchot, M.C., Hubert-moy, L.: Spectral unmixing of hyperspectral images with the independent component analysis and wavelet packets. In: Proc. of the International Geoscience and Remote Sensing Symposium (2001)
2. Stainvas, I., Lowe, D.: A generative model for separating illumination and reflectance from images. Journal of Machine Learning Research 4(7-8), 1499–1519 (2004)
3. Klinker, G.J., Shafer, S.A., Kanade, T.: A physical approach to color image understanding. Int. J. Comput. Vision 4(1), 7–38 (1990)
4. Tajima, J.: Illumination chromaticity estimation based on dichromatic reflection model and imperfect segmentation. In: Trémeau, A., Schettini, R., Tominaga, S. (eds.) CCIW 2009. LNCS, vol. 5646, pp. 51–61. Springer, Heidelberg (2009)
5. Li, C., Li, F., Kao, C., Xu, C.: Image Segmentation with Simultaneous Illumination and Reflectance Estimation: An Energy Minimization Approach. In: ICCV 2009: Proceedings of the Twelfth IEEE International Conference on Computer Vision (2009)
6. Bergman, M.: Some unmixing problems and algorithms in spectroscopy and hyperspectral imaging. In: Proc. of the 35th Applied Imagery and Pattern Recognition Workshop (2006)
7. Fu, Z., Tan, R., Caelli, T.: Specular free spectral imaging using orthogonal subspace projection. In: Proc. Intl. Conf. Pattern Recognition, vol. 1, pp. 812–815 (2006)
8. Shafer, S.A.: Using color to separate reflection components. Color Research and Applications 10(4), 210–218 (1985)
9. Jaynes, E.: On the rationale of maximum-entropy methods. In: Proceedings of the IEEE, 70(9), 939–952 (1982)
10. Jaynes, E.T.: Information Theory and Statistical Mechanics. Phys. Rev. 106(4), 620–630 (1957)
11. Finlayson, G.D., Schaefer, G.: Convex and Non-convex Illuminant Constraints for Dichromatic Colour Constancy. In: CVPR, vol. 1, pp. 598–604 (2001)
12. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220, 671–680 (1983)
13. Barnard, K., Martin, L., Coath, A., Funt, B.V.: A comparison of computational color constancy Algorithms – Part II: Experiments with image data. IEEE Transactions on Image Processing 11(9), 985–996 (2002)
14. Buchsbaum, G.: A Spatial Processor Model for Object Color Perception. Journal of The Franklin Institute 310, 1–26 (1980)
15. McCann, J.J., Hall, J.A., Land, E.H.: Color mondrian experiments: the study of average spectral distributions. Journal of Optical Society America 67, 1380 (1977)
16. Finlayson, G.D., Trezzi, E.: Shades of gray and colour constancy. In: Color Imaging Conference, pp. 37–41 (2004)
17. van de Weijer, J., Gevers, T., Gijsenij, A.: Edge-Based Color Constancy. IEEE Transactions on Image Processing 16(9), 2207–2214 (2007)
18. Kruse, F.A., Lefkoff, A.B., Boardman, J.B., Heidebrecht, K.B., Shapiro, A.T., Barloon, P.J., Goetz, A.F.H.: The Spectral Image Processing System (SIPS) - Interactive Visualization and Analysis of Imaging Spectrometer Data. Remote Sensing of Environment, Special issue on AVIRIS 44(2), 145–163 (1993)

A Game-Theoretic Approach to the Enforcement of Global Consistency in Multi-view Feature Matching

Emanuele Rodolà, Andrea Albarelli, and Andrea Torsello

Dipartimento di Informatica – Università Ca' Foscari – Venice, Italy

Abstract. In this paper we introduce a robust matching technique that allows to operate a very accurate selection of corresponding feature points from multiple views. Robustness is achieved by enforcing global geometric consistency at an early stage of the matching process, without the need of ex-post verification through reprojection. Two forms of global consistency are proposed, but in both cases they are reduced to pairwise compatibilities making use of the size and orientation information provided by common feature descriptors. Then a game-theoretic approach is used to select a maximally consistent set of candidate matches, where highly compatible matches are enforced while incompatible correspondences are driven to extinction. The effectiveness of the approach in estimating camera parameters for bundle adjustment is assessed and compared with state-of-the-art techniques.

1 Introduction

The selection of 3D point correspondences from their 2D projections is arguably one of the most important steps in image based multi-view reconstruction, as errors in the initial correspondences can lead to sub-optimal parameter estimation. The selection of corresponding points is usually carried out by means of interest point detectors and feature descriptors. Salient points are localized with subpixel accuracy by general detectors, such as Harris Operator [2] and Difference of Gaussians [6], or by using techniques that are able to locate affine invariant regions, such as Maximally stable extremal regions (MSER) [7] and Hessian-Affine [8]. This latter affine invariance property is desirable since the change in appearance of a scene region after a small camera motion can be locally approximated with an affine transformation. Once salient and well-identifiable points are found on each image, correspondences between the features in the various views must be extracted and fed to the bundle adjustment algorithm. To this end, each point is associated a descriptor vector with tens to hundreds of dimensions, which usually include a scale and a rotation value. Arguably the most famous of such descriptors are the Scale-invariant feature transform (SIFT) [4], the Speeded Up Robust Features (SURF) [3], and the Gradient Location and Orientation Histogram (GLOH) [9], and more recently the Local Energy based Shape Histogram (LESH) [10]. Features are designed so that similar image regions subject to similarity transformation exhibit descriptor vectors with small

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 354–364, 2010.

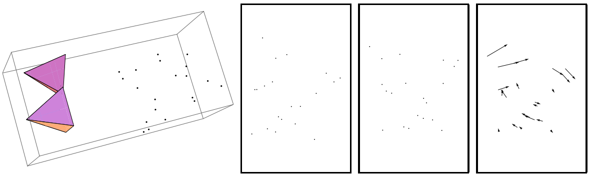


Fig. 1. Locally uniform 3D motion does not result in a locally uniform 2D motion. From left to right: 3D scene, left and right views, and motion estimation.

Euclidean distance. This property is used to match each point with a candidate with similar descriptor. However, if the descriptor is not distinctive enough this approach is prone to select many outliers since it only exploits local information. This limitation conflicts with the richness of information that is embedded in the scene structure. For instance, under the assumption of rigidity and small camera motion, features that are close in one view are expected to be close in the other one as well. In addition, if a pair of features exhibit a certain difference of angles or ratio of scales, this relation should be maintained among their respective matches. This prior information about scene structure can be accounted for by using a feature tracker [5,12] to extract correspondences, but this requires that the view positions be not far apart. Further, in the presence of strong parallax, a locally uniform 3D motion does not result in a locally uniform 2D motion, and for these reasons the geometric constraints can be enforced only locally (see Fig. 1 for an example). A common heuristic for the enforcement of global structure is to eliminate points that exhibit a large reprojection error after a first round of Bundle Adjustment [13]. Unfortunately this post-filtering technique requires good initial estimates to begin with.

In this paper we introduce a robust matching technique that allows to operate a very accurate inlier selection at an early stage of the process and without any need to rely on 3D reprojections. The approach selects feasible matches by enforcing global geometric consistency. Two geometric consistency models are presented. The first enforces that all pairs of correspondences between 2D views are consistent with a common 3D rigid transformation. Here, as is common in similar point-matching approaches, we assume that we have reasonable guesses for the intrinsic camera parameters and reduce the problem space to the search of a 3D rigid transformation from one image space to the other. This condition is in general underspecified, as a whole manifold of pairs of correspondences are consistent with a rigid 3D transformation. However, by accumulating mutual support through a large set of mutually compatible correspondences one can expect to reduce the ambiguity to a single 3D rigid transformation. In the proposed approach, high order consistency constraints are reduced to a second order compatibility where sets of 2D point correspondences that can be interpreted as projections of rigidly-transformed 3D points all have high mutual support. The reduction is obtained by making use of the scale and orientation information linked with each feature point in the SIFT descriptor [4] and a further reprojection that can be considered a continuous form of hypergraph clique expansion [15].

The second geometric consistency constraint assumes a weak perspective camera and matches together points whose maps are compatible with a common affine transformation. This allows us to extract small coherent clusters of points all laying at similar depths. The locally affine hypothesis could seem to be an unsound assumption for general camera motion, and in effect cannot account for point inversion due to parallax, but in the experimental section we will show that it holds well with the typical disparity found in standard data sets. Further, it should be noted that with large camera motion most, if not all, commonly used feature detectors fail, thus any inlier selection attempt becomes meaningless.

Once the geometric consistency contraints are specified, we can use them to drive the matching process. Following [14,1], we model the matching process in a game-theoretic framework, where two players extracted from a large population select a pair of matching points from two images. The player then receives a payoff from the other players proportional to how compatible his match is with respect to the other player's choice, where the compatibility derives from some utility function that rewards pair of matches that are consistent. Clearly, it is in each player's interest to pick matches that are compatible with those the other players are likely to choose. In general, as the game is repeated, players will adapt their behavior to prefer matchings that yield larger payoffs, driving all inconsistent hypotheses to extinction, and settling for an equilibrium where the pool of matches from which the players are still actively selecting their associations forms a cohesive set with high mutual support. Within this formulation, the solutions of the matching problem correspond to evolutionary stable states (ESS's), a robust population-based generalization of the notion of a Nash equilibrium. In a sense, this matching process can be seen as a contextual voting system, where each time the game is repeated the previous selections of the other players affect the future vote of each player in an attempt to reach consensus. This way, the evolving context brings global information into the selection process.

2 Pairwise Geometric Consistency

In what follows we will describe the two geometric constraints that will be used to drive the matching process. The first approach tries to impose that the points be consistent with a common 3D rigid transformation.

There are two fundamental hypotheses underlying the reduction to second order of this high-order 3D geometric consistency. First, we assume that the views have the same set of camera parameters, that we have reasonable guesses for the intrinsic parameters, and we can ignore lens distortion. Thus, the geometric consistency is reduced to the compatibility of the projected points with a single 3D rigid transformation related to the relative positions of the cameras. Second, we assume that the feature descriptor provides scale and orientation information and that this is related to actual local information in the 3D objects present in the scene. The effect of the first assumption is that the geometric consistency is reduced to a rigidity constraint that can be cast as a conservation along views of the distances between the unknown 3D position of the feature points, while the effect of the second assumption is that we can recover the missing depth information as a variation in scale between two views of the same point and that this variation is inversely proportional to variation in projected size of the local patch around the 3D point and, thus, to the projected size of the feature descriptor. More formally, assume that we have two points p_1 and p_2, which in one view have coordinates (u_1^1, v_1^1) and (u_2^1, v_2^1) respectively, while in a second image they have coordinates (u_1^2, v_1^2) and (u_2^2, v_2^2). These points, in the coordinate system of the first camera, have 3D coordinates $z_1^1(u_1^1, v_1^1, f)$ and $z_2^1(u_2^1, v_2^1, f)$ respectively, while in the reference frame of the second camera they have coordinates $z_1^2(u_1^2, v_1^2, f)$ and $z_2^2(u_2^2, v_2^2, f)$. Up to a change in units, these coordinates can be re-written as

$$p_1^1 = \frac{1}{s_1^1}\begin{pmatrix} u_1^1 \\ v_1^1 \\ f \end{pmatrix},\ p_2^1 = \frac{a}{s_2^1}\begin{pmatrix} u_2^1 \\ v_2^1 \\ f \end{pmatrix},\ p_1^2 = \frac{1}{s_1^2}\begin{pmatrix} u_1^2 \\ v_1^2 \\ f \end{pmatrix},\ p_2^2 = \frac{a}{s_2^2}\begin{pmatrix} u_2^2 \\ v_2^2 \\ f \end{pmatrix},$$

where f is the focal lenght and a is the ratio between the actual scales of the local 3D patches around points p_1 and p_2, whose projections on the two views give the perceived scales s_1^1 and s_1^2 for point p_1 and s_2^1 and s_2^2 for point p_2.

The assumption that both scale and orientation are linked with actual properties of the local patch around each 3D point is equivalent to having 2 points for each feature correspondence: the actual location of the feature, plus a virtual point located along the axis of orientation of the feature at a distance proportional to the actual scale of the patch. These pairs of 3D points must move rigidly going from the coordinate system of one camera to the other, so that given any two sets of correspondences with 3D points p_1 and p_2 and their corresponding virtual points q_1 and q_2, the distances between these four points must be preserved in the reference frames of every view (see Fig. 2).

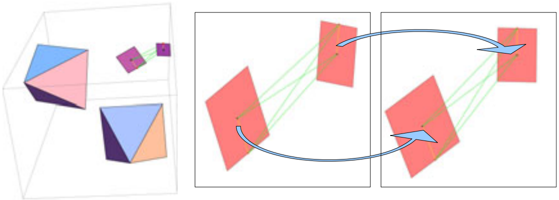


Fig. 2. Scale and orientation offer depth information and a second virtual point. The conservation of the distances in green enforces consistency with a 3D rigid transformation.

Under a frontal-planar assumption for each local patch, or, less stringently, under small variation in viewpoints, we can assign 3D coordinates to the virtual points in the reference frames of the two images:

$$q_1^1 = p_1^1 + \begin{pmatrix} \cos\theta_1^1 \\ \sin\theta_1^1 \\ 0 \end{pmatrix} \; q_2^1 = p_2^1 + a \begin{pmatrix} \cos\theta_2^1 \\ \sin\theta_2^1 \\ 0 \end{pmatrix}$$
$$q_1^2 = p_1^2 + \begin{pmatrix} \cos\theta_1^2 \\ \sin\theta_1^2 \\ 0 \end{pmatrix} \; q_2^2 = p_2^2 + a \begin{pmatrix} \cos\theta_2^2 \\ \sin\theta_2^2 \\ 0 \end{pmatrix},$$

where θ_i^j is the perceived orientation of feature i in image j. At this point, given two sets of correspondences between points in two images, namely the correspondence m_1 between a feature point in the first image with coordinates, scale and orientation $(u_1^1, v_1^1, s_1^1, \theta_1^1)$ with the feature point in the second image $(u_1^2, v_1^2, s_1^2, \theta_1^2)$, and the correspondence m_2 between the points $(u_2^1, v_2^1, s_2^1, \theta_2^1)$ and $(u_2^2, v_2^2, s_2^2, \theta_2^2)$ in the first and second image respectively, we can compute a distance from the manifold of feature descriptors compatible with a single 3D rigid transformation as

$$\begin{aligned} d(m_1, m_2, a) = (||p_1^1 - p_2^1||^2 - ||p_1^2 - p_2^2||^2)^2 + (||p_1^1 - q_2^1||^2 - ||p_1^2 - q_2^2||^2)^2 + \\ (||q_1^1 - p_2^1||^2 - ||q_1^2 - p_2^2||^2)^2 + (||q_1^1 - q_2^1||^2 - ||q_1^2 - q_2^2||^2)^2 \,. \end{aligned}$$

From this we define the compatibility between correspondences as $C(m_1, m_2) = \max_a e^{-\gamma d(m_1, m_2, a)}$, where a is maximized over a reasonable range of ratio of scales of local 3D patches. In our experiments a was optimized in the interval $[0.5; 2]$.

The second geometric consistency constraint assumes a weak perspective camera and matches together points whose maps are compatible with a common affine transformation. Specifically, we are able to associate to each matching strategy (a_1, a_2) one and only one similarity transformation, that we call $T(a_1, a_2)$. When this transformation is applied to a_1 it produces the point a_2, but when applied to the source point b_1 of the matching strategy (b_1, b_2) it does not need to produce b_2. In fact it will produce b_2 if and only if $T(a_1, a_2) = T(b_1, b_2)$, otherwise it will give a point b_2' that is as near to b_2 as the transformation $T(a_1, a_2)$ is similar $T(b_1, b_2)$. Given two matching strategies (a_1, a_2) and (b_1, b_2) and their respective associated similarities $T(a_1, a_2)$ and $T(b_1, b_2)$, we calculate their reciprocal reprojected points as:

$$\begin{aligned} a_2' &= T(b_1, b_2)a_1 \\ b_2' &= T(a_1, a_2)b_1 \end{aligned}$$

That is the virtual points obtained by applying to each source point the similarity transformation associated to the other match (see Fig 3). Given virtual points a_2' and b_2' we are finally able to calculate the payoff between (a_1, a_2) and (b_1, b_2) as:

$$\Pi((a_1, a_2), (b_1, b_2)) = e^{-\lambda \max(||a_2 - a_2'||, ||b_2 - b_2'||)} \tag{1}$$

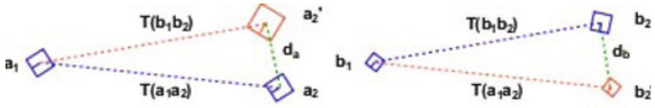


Fig. 3. The payoff between two matching strategies is inversely proportional to the maximum reprojection error obtained by applying the affine transformation estimated by a match to the other

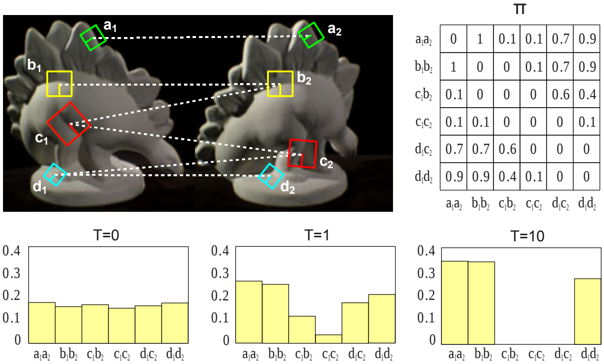
Where λ is a selectivity parameter that allows to operate a more or less strict inlier selection. If λ is small, then the payoff function (and thus the matching) is more tolerant, otherwise the evolutionary process becomes more selective as λ grows.

The rationale of the payoff function proposed in equation 1 is that, while by changing point of view the similarity relationship between features is not mantained (as the object is not planar and the transformation is projective), we can expect the transformation to be a similarity at least "locally". This means that we aim to extract clusters of feature matches that belong to the same region of the object and that tend to lie in the same level of depth.

Each matching process selects a group of matching strategies that are coherent with respect to a local similarity transformation. This means that if we want to cover a large portion of the subject we need to iterate many times and prune the previously selected matches at each new start. Obviously, after all the depth levels have been swept, small and not significative residual groups start to emerge from the evolution. To avoid the selection of this spurious matches we fixed a minimum cardinality for each valid group.

3 Game-Theoretic Feature Matching

We model the matching process in a game-theoretic framework [1], where two players extracted from a large population select a pair of matching points from two images. The player then receives a payoff from the other players proportional to how compatible his match is with respect to the other player's choice. Clearly, it is in each player's interest to pick matches that are compatible with those the other players are likely to choose. It is supposed that some selection process operates over time on the distribution of behaviors favoring players that receive larger payoffs and driving all inconsistent hypotheses to extinction, finally settling for an equilibrium where the pool of matches from which the players are still actively selecting their associations forms a cohesive set with high mutual support. More formally, let $O = \{1, \cdots, n\}$ be the set of available strategies (*pure strategies* in the language of game theory) and $C = (c_{ij})$ be a matrix specifying the payoff that an individual playing strategy i receives against someone playing strategy j. A *mixed strategy* is a probability distribution $\mathbf{x} = (x_1, \ldots, x_n)^T$ over the available strategies O, thus lying in the n-dimensional standard simplex $\Delta^n = \{\mathbf{x} \in \mathbb{R}^n : \forall i \in 1 \ldots n \; x_i \geq 0, \; \sum_{i=1}^{n} x_i = 1\}$. The expected payoff



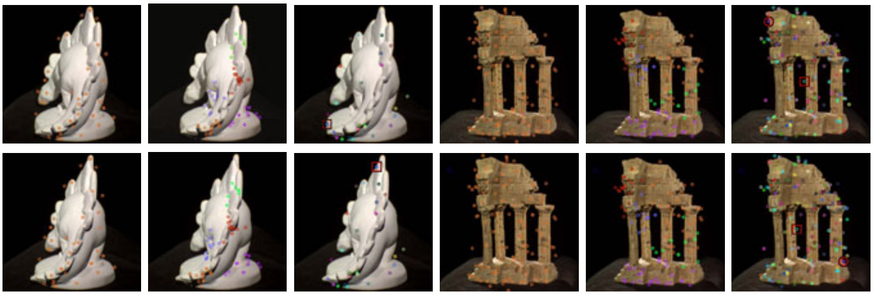
	a_1a_2	b_1b_2	c_1b_2	c_1c_2	d_1c_2	d_1d_2
a_1a_2	0	1	0.1	0.1	0.7	0.9
b_1b_2	1	0	0	0.1	0.7	0.9
c_1b_2	0.1	0	0	0	0.6	0.4
c_1c_2	0.1	0.1	0	0	0	0.1
d_1c_2	0.7	0.7	0.6	0	0	0
d_1d_2	0.9	0.9	0.4	0.1	0	0

Fig. 4. An example of the evolutionary process. Four feature points are extracted from two images and a total of six matching strategies are selected as initial hypotheses. The matrix Π shows the compatibilities between pairs of matching strategies according to a one-to-one similarity-enforcing payoff function. Each matching strategy got zero payoff with itself and with strategies that share the same source or destination point (i.e., $\Pi((b_1, b_2), (c_1, b_2)) = 0$). Strategies that are coherent with respect to a similarity transformation exhibit high payoff values (i.e., $\Pi((a_1, a_2), (b_1, b_2)) = 1$ and $\pi((a_1, a_2), (d_1, d_2)) = 0.9$)), while less compatible pairs get lower scores (i.e., $\pi((a_1, a_2), (c_1, c_2)) = 0.1$). Initially (at T=0) the population is set to the barycenter of the simplex and slightly perturbed. After just one iteration, (c_1, b_2) and (c_1, c_2) have lost a significant amount of support, while (d_1, c_2) and (d_1, d_2) are still played by a sizable amount of population. After ten iterations (T=10) (d_1, d_2) has finally prevailed over (d_1, c_2) (note that the two are mutually exclusive). Note that in the final population $((a_1, a_2), (b_1, b_2))$ have a larger support than (d_1, d_2) since they are a little more coherent with respect to similarity.

received by a player choosing element i when playing against a player adopting a mixed strategy $\mathbf{x}$ is $(C\mathbf{x})_i = \sum_j c_{ij} x_j$, hence the expected payoff received by adopting the mixed strategy $\mathbf{y}$ against $\mathbf{x}$ is $\mathbf{y}^T C\mathbf{x}$. A strategy $\mathbf{x}$ is said to be a *Nash equilibrium* if it is the best reply to itself, i.e., $\forall \mathbf{y} \in \Delta,\ \mathbf{x}^T C\mathbf{x} \geq \mathbf{y}^T C\mathbf{x}$. A strategy $\mathbf{x}$ is said to be an *evolutionary stable strategy* (ESS) if it is a Nash equilibrium and $\forall \mathbf{y} \in \Delta \quad \mathbf{x}^T C\mathbf{x} = \mathbf{y}^T C\mathbf{x} \Rightarrow \mathbf{x}^T C\mathbf{y} > \mathbf{y}^T C\mathbf{y}$. This condition guarantees that any deviation from the stable strategies does not pay. The search for a stable state is performed by simulating the evolution of a natural selection process. Under very loose conditions, any dynamics that respect the payoffs is guaranteed to converge to Nash equilibria and (hopefully) to ESS's; for this reason, the choice of an actual selection process is not crucial and can be driven mostly by considerations of efficiency and simplicity. We chose to use the replicator dynamics, a well-known formalization of the selection process governed by the recurrence $\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^t \frac{(C\mathbf{x}^t)_i}{\mathbf{x}^{t\,T} C\mathbf{x}^t}$, where $\mathbf{x}_i^t$ is the proportion of the population that plays the i-th strategy at time t. Once the population has reached a local maximum, all the non-extincted pure strategies can be considered selected by the game. One final note should be made about one-to-one matching. Since each source feature can correspond with at most one destination point, it is desirable to avoid any kind of multiple match. It is easy to show that a pair of strategies with mutual zero payoff cannot belong to the support of an ESS (see [1]), thus any payoff function can easily be adapted to enforce one-to-one matching by setting to 0 the payoff of mates that share either the source or the destination point.

4 Experimental Results

To evaluate the performance of our proposals, we compared the results with those obtained with the keymatcher included in the structure-from-motion suite Bundler [13]. For the first set of experiments we selected pairs of adjacent views from the "DinoRing" and "TempleRing" sequences from the Middlebury Multi-View Stereo dataset [11]; for these models, camera parameters are provided and used as a ground-truth. For all the sets of experiments we evaluated the differences in radians between the (calibrated) ground-truth and respectively the estimated rotation angle ($\Delta\alpha$) and rotation axis ($\Delta\gamma$). The "Dino" model is a difficult case in general, as it provides very few features; the upper part of Fig. 5 shows the correspondences produced by our game -theoretic matching approach



GT-3Drigid GT-2Daffine Bundler GT-3Drigid GT-2Daffine Bundler

Dino sequence

	GT-3Drigid	GT-2Daffine	Bundler
Matches	262.5 ± 61.4	271.1 ± 64.2	172.4 ± 79.5
$\Delta\alpha$	0.0668 ± 0.0777	0.0497 ± 0.0810	0.0767 ± 0.1172
$\Delta\gamma$	0.4393 ± 0.4963	0.3184 ± 0.3247	0.6912 ± 0.8793

Temple sequence

	GT-3Drigid	GT-2Daffine	Bundler
Mathces	535.7 ± 38.7	564.3 ± 37.2	349.3 ± 36.2
$\Delta\alpha$	0.1326 ± 0.0399	0.0989 ± 0.0224	0.1414 ± 0.0215
$\Delta\gamma$	0.0809 ± 0.0144	0.0792 ± 0.0091	0.0850 ± 0.0065

Fig. 5. Results obtained with the Dino and Temple data sets

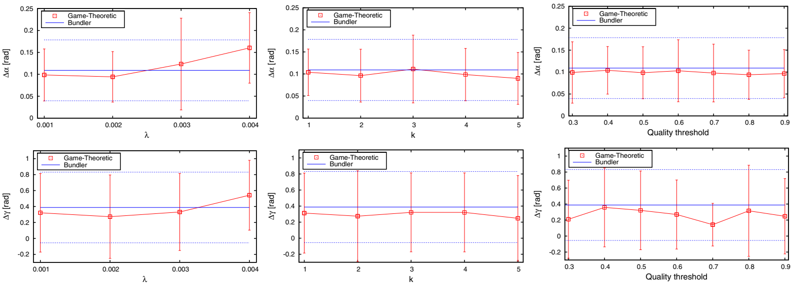


Fig. 6. Analysis of the performance of the approach with respect to variation of the parameters of the algorithm

with geometric constraints enforcing a 3D rigid transformation (GT-3Drigid), the approach with the weak perspective camera assumptions (GT-2Daffine), and the Bundler matcher (Bundler). The color of the points matched using GT-2Daffine relate to the extraction group, i.e., points with the same color have been matched at th same re-iteration of the game-theoretic matching process. The "Temple" model is richer in features and for visualization purposes we only show a subset of the detected matches for all three techniques. The Bundler matcher, while still achieving good results, provides some mismatches in both cases. This can be explained by the fact that the symmetric parts of the object, e.g. the pillars in the temple model, result in very similar features that are hard to disambiguate by a purely local matcher. Both our methods, on the other hand, by enforcing global consistency, can effectively disambiguate the matches. Looking at the results we can see that both our approaches extract around 50% more correspondences than Bundler. The first approach provides a slight increase in precision and reduction in variance of the estimates. Note, however, that the selected measures evaluate the quality of the underlying least square estimates of the motion parameters after a reprojection step, thus small variations are expected. The approach enforcing a global 2D affine transformation exhibits a larger increase in precision and reduction in variance. This can be explained by the fact that the adjacent views of the two sequences have very little parallax effects, thus the weak persective camera assumption holds quite well. In this context the stricter model is better specified and thus more discriminative.

Next, we analyzed the impact of the algorithm parameters over the quality of the results obtained. To this end, we investigated three parameters: the similarity decay λ, the number k of candidate mates per features, and the *quality threshold*, that is the minimum support for a correspondence to be considered non-extinct, divided by the maximum support in the population. Figure 6 reports the results of these experiments. The goal of these experiments was to show the sensitivity to the matcher's parameters, not to choose between constraints, so only the 3D geometric constraint was used. Overall, these experiments show that almost all reasonable values of the parameters give similar values for the match, thus those parameters have little influence over the quality of the result, with the

Game-Theoretic approach achieving better average results and smaller standard deviation than the Bundler matcher.

5 Conclusions

In this paper we introduced a robust matching technique for feature points from multiple views. Robustness is achieved by enforcing global geometric consistency in a pairwise setting. Two different geometric consistency models are proposed. The first enforces the compatibility with a single 3D rigid transformation of the points. This is achieved by using the scale and orientation information offered by SIFT features and projecting what is left of a high-order compatibility problem into a pairwise compatibility measure, by enforcing the conservation of distances between the unknown 3D positions of the points. The second model assumes a weak perspective camera model and enforces that points are subject to an affine transformation. This extracts only local groups at similar depths, but the matching process is repeated to cover the whole scene. In both cases, a game-theoretic approach is used to select a maximally consistent set of candidate matches, where highly compatible matches are enforced while incompatible correspondences are driven to extinction. Experimental comparisons with a widely used technique show the ability of our approach to obtain more accurate estimates of the scene parameters.

Acknowledgment

We acknowledge the financial support of the Future and Emerging Technology (FET) Programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open project SIMBAD grant no. 213250.

References

1. Albarelli, A., Bulò, S.R., Torsello, A., Pelillo, M.: Matching as a non-cooperative game. In: ICCV (2009)
2. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. Fourth Alvey Vision Conference, pp. 147–151 (1988)
3. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
4. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 20, 91–110 (2003)
5. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence, pp. 674–679 (1981)
6. Marr, D., Hildreth, E.: Theory of Edge Detection. Royal Soc. of London Proc. Series B 207, 187–217 (1980)

7. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing 22(10), 761–767 (2002) British Machine Vision Computing (2002)
8. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 128–142. Springer, Heidelberg (2002)
9. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(10), 1615–1630 (2005)
10. Sarfraz, M.S., Hellwich, O.: Head pose estimation in face recognition across pose scenarios. In: VISAPP (1), pp. 235–242 (2008)
11. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: CVPR 2006, pp. 519–528 (2006)
12. Shi, J., Tomasi, C.: Good features to track. In: CVPR, pp. 593–600 (1994)
13. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. Int. J. Comput. Vision 80(2), 189–210 (2008)
14. Torsello, A., Bulò, S.R., Pelillo, M.: Grouping with asymmetric affinities: A game-theoretic perspective. In: CVPR 2006, pp. 292–299 (2006)
15. Zien, J.Y., Schlag, M.D.F., Chan, P.K.: Multi-level spectral hypergraph partitioning with arbitrary vertex sizes. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 18, 1389–1399 (1999)

An Algorithm for Recovering Camouflage Errors on Moving People

D. Conte, P. Foggia, G. Percannella, F. Tufano, and M. Vento

Dipartimento di Ingegneria dell'Informazione ed Ingegneria Elettrica
Via Ponte Don Melillo, 84084 Fisciano (SA), Italy
{dconte,pfoggia,pergen,ftufano,mvento}@unisa.it

Abstract. In this paper we present a model-based algorithm working as a post-processing phase of any foreground object detector. The model is suited to recover camouflage errors producing the segmentation of an entity in small and unconnected parts. The model does not require training procedures, but only information about the estimated size of the person, obtainable when an inverse perspective mapping procedure is used.

A quantitative evaluation of the effectiveness of the method, used after four well known moving object detection algorithms has been carried out. Performance are given on a variety of publicly available databases, selected among those presenting highly camouflaged objects in real scenes referring to both indoor and outdoor environments.

1 Introduction

Most of video analysis applications require the extraction of the moving objects from the scene, so as to apply further processing aimed to classify them into different categories (persons, vehicles, animals, bags or luggage), or to characterize the trajectories traced in the environment. Both these elaborations are necessary processing steps toward the understanding of the events occurring in the scene. Of course, the semantic analysis is unavoidably affected by any error occurring in the detection phase, as the fragmentation of a single object into parts (for instance, a person into pieces of body), or the merging of close objects into a bigger one.

In this complex application framework, starting from the first approaches to the problem, *background subtraction* is being considered as a simple and powerful technique for detecting the moving objects as opposed to the static elements that are part of the observed scene. Although many related issues have been receiving further research attention, and some problems are still considered open, the background subtraction techniques are gaining popularity among the object detection algorithms.

Experimentation over the years, highlighted that the background subtraction technique is subject to a set of well known problems, categorized in [9]; consequently, most of the algorithms proposed up to now, have been devised so as to face the above cited problems. A noteworthy exception is constituted by

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 365–374, 2010.

camouflage: an intrinsic and hardly faceable problem occurring when the pixel characteristics of a foreground object are too similar to the background to be discerned, as happens when a person is wearing clothes having similar colors to the background. The effect is that the difference of these pixels from the background model is under the threshold, and consequently incorrectly considered as foreground pixels.

Camouflage has comparatively received less attention than the other problems, probably because most of detection methods operate at a pixel level, where there is not enough information to effectively tackle this problem. So most authors either ignore the issue, testing their detection algorithms in contexts where camouflage is unlikely, or assume that later processing phases will be able to correct the anomalies induced by camouflage.

Among the papers specifically devoted to the camouflage problem, Tankus and Yeshurum [8] propose the use of an operator to enhance areas whose shading corresponds to a convex object to separate such areas from a "flat" background with similar intensity and texture. However the method is not suitable for environments in which the background also contains convex objects, and does not work well for objects with dark colors.

The paper by Harville et al. [4] is representative of an approach to the problem that involves the use of depth information to detect camouflaged objects. The authors also evaluate other popular video analysis methods proposed in the literature, maintaining that among the considered systems, only the ones incorporating depth information are able to deal with camouflage. While the use of depth information can surely improve the detection performance of a video analysis system, it has a non negligible computational cost and, more important, it precludes the use of the legacy cameras often already installed for applications such as video-surveillance or traffic monitoring.

The paper by Boult et al. [1] is devoted to *intentional* camouflage, and uses background subtraction with two thresholds: a larger and a smaller one, used to respectively detect pixels that are certainly in the foreground, or either part of the background or a camouflaged part of the foreground. The regions detected using the two thresholds are then grouped using suitable conditions to form the so called "Quasi Connected Components" to recover the split of camouflaged objects.

The approach proposed by TrakulPong and Bowden [10] is instead based on a simple model of the shape, integrated in the tracking phase; it builds a statistical model of the shape of the tracked object, and when an abrupt shape change occurs, the algorithm assumes it is due to camouflage and tries to match the object image at the previous frame to restore the correct shape.

The paper by Guo et al. [3] proposes to address the camouflage problem by performing a temporal averaging of the frames before computing or updating the background model. The idea is that this way, the model will have a smaller variance and so a smaller detection threshold can be used. However, as the experiments performed by the authors show, the method has problems with slowly moving objects.

In this paper we propose a method for correcting the errors typically generated in the detection phase of a background subtraction procedure, in presence of camouflage; its generality allows the user to apply it as a post processing module operating after a generic object detection algorithm.

The errors, consisting in the fragmentation of the actual object in the scene, are detected and corrected by a grouping phase performed on the basis of a model of the shape to be recognized, that in our system are isolated people. Once the object have been detected, a set of merges of adjacent objects are performed in the case that, after fusion the obtained object is more likely to be a person. For the sake of notational simplicity, hereinafter we denote with blobs all those objects generated as output by the detection phase, independently of the fact that they are actually persons, fragments of them or unanimated objects. The algorithm has been devised so as to make it possible the recursive merging of blobs, so as to allow the possibility of recovering highly critical situations caused by camouflage, as the split of a single person in a plurality of small parts, otherwise considered as noise.

The algorithm has been widely experimented applying it on the output of four well known detection algorithms, over a wide video database publicly available, including indoor and outdoor scenes.

2 Proposed Method

As anticipated in the introduction, the camouflage problem causes that the foreground mask of a person is split into two or more foreground blobs so generating plenty of different configurations. The Fig. 1 shows the output of some different foreground detection algorithms on various video sequences. It is evident from the picture that, even recognizable by a human being, the obtained configurations of blobs are far to be considered as ideal. A great effort must be done

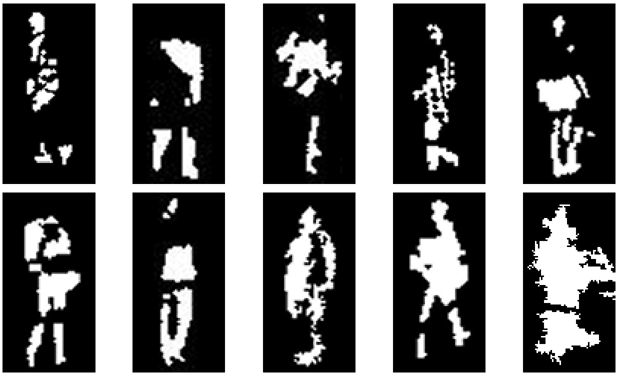


Fig. 1. Examples of errors generated in the detection phase of a background subtraction procedure in presence of camouflage. In all cases a person is detected as a set of separated blobs.

to process them so as to obtain, for each person a single blob. The problem of recovering such kind of errors is equivalent to the problem of finding how several foreground blobs can be grouped together in order to suitably represent the object of interest, in our case a person. By examining a single blob it is not possible to determine if it can be considered as a part of a larger object or if it is spuriously generated during the foreground detection process. So, to reconstruct the objects affected by camouflage errors we need to define a model of the desired object and a procedure for suitably grouping the obtained blobs, so as to adequately fit the model.

In this paper we focus our attention to the detection of moving people, but the underlying idea can be generalized to the detection of other kinds of objects (as cars, animals, etc.).

Obviously the model must be carefully defined: a too detailed model would result in many missed detections, while on the contrary a too general one would cause the spurious generation of a plenty of false positive errors (blobs due to detection errors grouped to form, erroneously, objects of interest).

The chosen model starts from the simple consideration, that, as shown in Fig. 1, however the parts are arranged, they fall into an ideal box representing a person. It is worth noting that the model cannot be defined on the basis of the size expressed in pixels. The perspective causes, in fact, that a same configuration of pixels represents an object of different actual size depending on the distance from the camera. Therefore, the model be must defined in terms of actual size, and a suitably defined Inverse Perspective Mapping procedure must be used to pass from measures in the pixel space to actual ones.

Algorithm 1. The pseudo-code of the grouping algorithm

```
$S \leftarrow$ all detected blobs
$C \leftarrow S \times S$
while $\exists (X, Y) \in C$ do
  comment: Perform and verify the conditions for grouping blobs $X$ and $Y$
  $R1 \leftarrow right_p(X) \geq left_p(Y) \land left_p(X) \leq right_p(Y)$
  $Z \leftarrow X \cup Y$
  <perform Inverse Perspective Mapping to calculate the actual size of Z>
  $R2 \leftarrow height_r(Z) \in [h_1, h_2]$
  $R3 \leftarrow width_r(Z) \in [b_1, b_2]$
  if $R1 \land R2 \land R3$ then
    comment: Perform the grouping and update the set of blobs
    <connect the two foreground blobs $X$ and $Y$ by joining their barycenter>
    $S \leftarrow S - \{X, Y\}$
    $S \leftarrow S \cup \{Z\}$
    $C \leftarrow S \times S$
  else
    $C \leftarrow C - \{(X, Y)\}$
  end if
end while
```

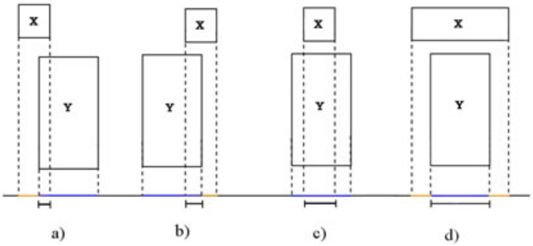


Fig. 2. The possible configurations of the overlaps between the projections of two boxes on the horizontal axis

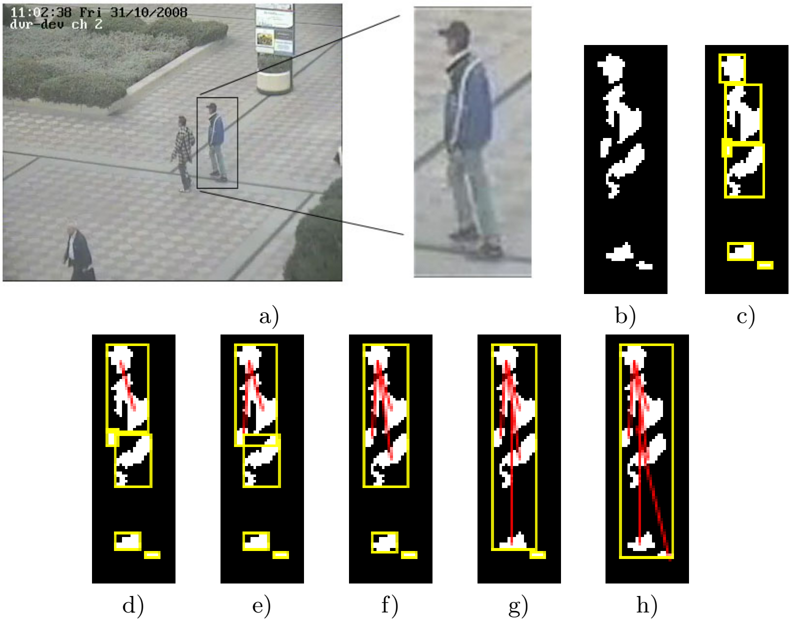


Fig. 3. An example of the algorithm's processing: a) original frame and the portion under analysis; b) the resulting foreground detection and the c) resulting bounding boxes; d), e), f), g), h) the steps of the algorithms on the considered portion of the frame

The adopted model represents a person as a box defined by four parameters h_1, h_2, b_1 and b_2 that, respectively, are the minimum and maximum actual height and the minimum and maximum actual width.

The pseudo-code of the algorithm is sketched in Algorithm 1. The aim of the algorithm is to group two or more blobs in order to form a unique blob representing a person (according to the defined model). Coherently with the adopted representation of a person, the algorithm represents each blob by its bounding box. The procedure operates by repeatedly merging couples of blobs into larger ones until the new blob best fits the defined model of person. Two blobs are grouped if and only if all the following conditions are verified:

R1. The projection on the horizontal axis of the bounding box of the considered blobs are overlapped (see Fig. 2). Note that the coordinates of the two boxes ($left_p(X)$, $right_p(X)$, $left_p(Y)$, $right_p(Y)$ in Algorithm 1) are expressed in pixels.
R2. The actual height, in meters, of the box grouping the two blobs ($height_r(Z)$ in Algorithm 1) is included between h_1 and h_2.
R3. The actual width, in meters, of the box grouping the two blobs ($width_r(Z)$ in Algorithm 1) is included between b_1 and b_2.

To verify the last two conditions, we firstly build the box grouping the considered blobs starting from the corresponding boxes by their pixels coordinates. Then, the Inverse Perspective Mapping is applied to the constructed box in order to determine its real size.

It is important to highlight that the proposed method is not computationally expensive because the number of detected boxes per frame is never greater than one or two dozens.

In Fig. 3 an example of the application of the algorithm is sketched.

3 Experimental Results

The experimental validation of the proposed method has been carried out by evaluating the performance improvements obtained when it is used as a post-processing on the output of four well known techniques of foreground detection; in particular:

- the *Mixture of Gaussians* (from now on called *MOG*), in the version proposed by Kaewtrakulpong and Bowden in [5];
- the *Enhanced Background Subtraction* (from now on *EBS*), proposed by Conte et al. in [2];
- the *Self-Organizing Background Subtraction* (from now on *SOBS*), proposed by Maddalena and Petrosino in [7];
- the *Statistical Background Algorithm* (from now on *SBA*), proposed by Li et al. in [6].

The performance is measured by using the *f-score* index, defined as the harmonic mean of *precision* and *recall*, according to the following formulas:

$$precision = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN} \tag{1}$$

In the previous formulas, the true positive (TP), false positive (FP) and false negative (FN) are given by:

$$TP = \sum_{g \in G} \sum_{d \in D} \frac{|g \cap d|}{|g \cup d|} \qquad FP = \sum_{d \in D} \frac{|d| - max_{g \in G} |d \cap g|}{|d|}$$

$$FN = \sum_{g \in G} \frac{|g| - max_{d \in D} |d \cap g|}{|g|}$$

where G is the set of objects of the ground truth and D is the set of objects really detected by the algorithm (each object is represented by its bounding box).

Tests have been done on a dataset of five real video sequences either indoor or outdoor. In Table 1 the main features of the considered videos are reported: the visual properties (number of frames of the sequence, frame rate expressed in fps, resolution), a short description of the content. The dataset has been also characterized in terms of the total number of foreground objects in each sequence and the number of isolated persons. This data allows, on one side, to evaluate the effectiveness of the proposed method in terms of people detected by grouping single pieces and, on the other side, to quantify the overall detection performance (i.e. when in the scene there are also objects the method was not designed to handle, as animals, bags, ...)

The NA1-NA3 videos were acquired by the authors on a large square in different lighting and weather conditions, with several persons walking. The PETS video belongs to the dataset published at the 2006 edition of the PETS workshop and contains a scene framed within a railway station. The MSA sequence, presented in [7], refers to an indoor scene.

Table 2 reports the performance of the considered four algorithms when the proposed method is adopted or not together with the relative improvements. The results are given in two cases: respectively, when all the objects in the dataset or only people objects are considered.

Table 1. Main features of the employed dataset: the properties (number of frames of the sequence, frame rate expressed in fps, resolution), a short description of the content and of the total number of objects and the number of isolated persons

Video ID	Properties	Description	# of objects	# of people objects
NA1	9'365, 25, 352x288	outdoor, sunny, very dark shadows	19'093	17'875 (93.6%)
NA2	4'575, 25, 352x288	outdoor, cloudy, very high camouflage, few shadows	9'333	7'651 (82.0%)
NA3	21'000, 25, 352x288	outdoor, late afternoon, high camouflage, very long shadows	20'568	18'303 (89.0%)
PETS	2'556, 25, 768x576	indoor, reflections	5'779	4'823 (83.5%)
MSA	528, 30, 352x288	indoor, vertical shadows	685	329 (48%)

Table 2. Object detection performance given in terms of *f-score* obtained considering when all the objects of the dataset (rows denoted with *all*), or only people objects (rows denoted with *people*). The columns *before* and *after* show the performance of the original algorithms without and with the proposed post-processing, respectively.

Video	**Object**	**EBS**			**MOG**			**SBA**			**SOBS**		
ID	**type**	bef.	aft.	imp.	bef.	aft.	imp.	bef.	aft.	imp.	bef.	aft.	imp.
NA1	*all*	.671	.673	0.3%	.657	.668	2.0%	.220	.304	38.5%	.530	.573	8.0%
	people	.732	.747	2.1%	.646	.645	0.1%	.212	.312	46,9%	.516	.591	14.6%
NA2	*all*	.775	.875	12.9%	.671	.676	0.7%	.256	.278	8.5%	.434	.510	17.4%
	people	.729	.873	19.8%	.648	.663	2.3%	.198	.269	35.9%	.416	.449	7.9%
NA3	*all*	.555	.693	24.7%	.644	.649	0.7%	.204	.227	10.9%	.287	.420	46.4%
	people	.554	.705	27.1%	.646	.656	1.5%	.202	.229	13.3%	.304	.430	41.1%
PETS	*all*	.773	.801	3.6%	.753	.651	-13.4%	.645	.635	-1.5%	.582	.606	4.0%
	people	.724	.818	13.0%	.623	.643	3.2%	.423	.424	0.1%	.544	.597	9.8%
MSA	*all*	.847	.904	6.6%	.520	.539	3.5%	.163	.206	26.4%	.850	.850	-
	people	.839	.921	9.7%	.542	.565	4.3%	.613	.622	1.5%	.816	.828	1.5%

If we consider the results reported in Table 2 it is possible to note that in the large majority of cases the use of the proposed method improves the object detection performance.

By looking at the data in Table 2 with respect to the video sequence it is possible to consider that according to the obtained performance improvements, the video sequences can be roughly divided in three groups. The first group, composed by the NA2 and NA3 videos, is the one on which almost all the algorithms reach the highest improvements, ranging from about 10% to over 45%, with the exception of MOG whose behavior will be deeper discussed in the following. The second group, containing the NA1 and the MSA sequences, presents moderate improvements, while, finally, the performance is generally low on PETS. It is worth pointing out that the behavior of the algorithms on the above defined groups can be related to the characteristics of the videos as described in the Table 1. The limitation of color gamut in the video sequences belonging to the first group, due to the poor scene illumination (cloudy in NA2 and late noon in NA3), favors the occurrence of the camouflage errors: in this case all the algorithms significantly benefits from the use of the proposed grouping procedure. Both the video sequences in the second group contain well illuminated scenes, so as that the camouflage problem occurs less frequently, making the improvements provided by our method less evident. A final consideration is about the efficiency of most of the considered object detectors which tends to worsen when the proposed method is applied on the PETS video. This behavior can be explained by considering that these video sequences were framed in a complex indoor environment with artificial lighting and reflective surfaces. These conditions cause that the original detection algorithms produce numerous false blobs, in some cases erroneously grouped by the grouping procedure.

Moreover, if we analyze the data in Table 2 with respect to the original foreground detection algorithm, it is evident that the adoption of the proposed

Table 3. Absolute number of objects in the dataset associated to persons, split by the considered detection algorithms; performance are given before and after the grouping algorithm

Video ID	**EBS**			**MOG**			**SBA**			**SOBS**		
	bef.	after	impr.	bef.	after	impr.	bef.	after	impr.	bef.	after	impr.
NA1	530	394	26.7%	338	333	1.5%	122	67	45.1%	362	273	24.6%
NA2	419	206	50.8%	151	125	17.2%	106	72	32.1%	216	171	20.8%
NA3	1043	206	80.2%	137	129	5.8%	147	81	44.9%	573	395	31.1%
PETS	215	147	31.6%	106	79	25.5%	91	76	16.5%	185	149	19.5%
MSA	17	5	70.6%	7	6	14.3%	21	19	9.5%	11	9	18.2%

grouping procedure after the EBS, the SBA and the SOBS object detection algorithms produces significant improvements with respect to the f-score that in many cases are above 10%. The only exception is represented by the MOG algorithm that does not benefit from it. This behavior can be justified by considering that the MOG technique (with the exception of the PETS video) tends to be less sensitive to camouflage problems than other typical background subtraction algorithms.

It is worth pointing out that the above considerations are generally valid either when all the objects in the dataset are considered or the tests are carried out with respect only to the people objects.

Table 3 reports an evaluation of the reduction of the problem of objects splitting when the proposed method is used or not. The tests were done by considering only the objects that are associated to persons. The results in Table 3 still confirm that the grouping procedure is effective in recovering the split errors due to camouflage: in all the experiments, the proposed method significantly reduces the number of person detected as separated in several fragments.

4 Conclusions

In this paper we present a model-based method for removing errors caused by camouflage in the detection of foreground isolated persons for video surveillance applications. The approach is designed to be used as a post-processing phase of a generic background subtraction algorithm.

A wide experimentation confirmed the effectiveness of the method able to significantly improve the performance in the detection of persons. The tests also highlighted that this improvement can sometimes be moderate, expecially when it is used on videos characterized by very complex environments that cause the detection of many false foreground blobs by the original background subtraction algorithm: in some cases, the false blobs may be merged so determining an erroneous detection of persons. This problem has a very poor impact, even if it can possibly be reduced by suitably refining the model. As future work, we are going to extend the approach to other application domains, as the traffic monitoring through the definition of suitable models for the objects of interest (i.e. cars, trucks, bus, ...).

References

1. Boult, T.E., Michaels, R.J., Gao, X., Eckmann, M.: Into the woods: visual surveillance of noncooperative and camouflaged targets in complex outdoor settings. Proceedings of IEEE 89(10), 1382–1402 (2001)
2. Conte, D., Foggia, P., Petretta, M., Tufano, F., Vento, M.: Meeting the Application Requirements of Intelligent Video Surveillance Systems in Moving Object Detection. In: Singh, S., Singh, M., Apte, C., Perner, P. (eds.) ICAPR 2005. LNCS, vol. 3687, pp. 653–662. Springer, Heidelberg (2005)
3. Guo, H., Dou, Y., Tian, T., Zhou, J., Yu, S.: A robust foreground segmentation method by temporal averaging multiple video frames. In: International Conference on Audio, Lenguage and Image Preocessing, pp. 878–882 (2008)
4. Harville, M., Gordon, G., Woodfill, J.: Foreground segmentation using adaptive mixture models in color and depth. In: IEEE Workshop on Detection and Recognition of Events in Video, pp. 3–11 (2001)
5. Kaewtrakulpong, P., Bowden, R.: An improved adaptive background mixture model for realtime tracking with shadow detection. In: Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, AVBS 2001, Video Based Surveillance Systems: Computer Vision and Distributed Processing (2001)
6. Li, L., Huang, W., Gu, I.Y.H., Tian, Q.: Foreground object detection from videos containing complex background. In: Proceedings of the eleventh ACM international conference on Multimedia (2003)
7. Maddalena, L., Petrosino, A.: A self-organizing approach to background subtraction for visual surveillance applications. IEEE Transactions on Image Processing 17(7), 1168–1177 (2008)
8. Tankus, A., Yeshurum, Y.: Convexity-based visual camouflage breaking. Computer Vision and Image Understanding 82, 208–237 (2001)
9. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and practice of background maintenance. In: Proc. IEEE Int. Conf. Computer Vision, vol. 1, pp. 255–261 (1999)
10. TrakulPong, P.K., Bowden, R.: A real time adaptative visual surveillance system for tracking low-resolution colour targhets in dynamically changing scenes. Image and Vision Computing 17, 913–929 (2003)

Semi-supervised Clustering Using Heterogeneous Dissimilarities

Manuel Martín-Merino

Universidad Pontificia de Salamanca
C/Compañía 5, 37002, Salamanca, Spain
mmartinmac@upsa.es

Abstract. The performance of many clustering algorithms such as k-means depends strongly on the dissimilarity considered to evaluate the sample proximities. The choice of a good dissimilarity is a difficult task because each dissimilarity reflects different features of the data. Therefore, different dissimilarities should be integrated in order to reflect more accurately which is similar for the user and the problem at hand.

In many applications, the user feedback or the a priory knowledge about the problem provide pairs of similar and dissimilar examples. This side-information may be used to learn a distance metric and to improve the clustering results. In this paper, we address the problem of learning a linear combination of dissimilarities using side information in the form of equivalence constraints. The minimization of the error function is based on a quadratic optimization algorithm. A smoothing term is included that penalizes the complexity of the family of distances and avoids overfitting.

The experimental results suggest that the method proposed outperforms a standard metric learning algorithm and improves the classical k-means clustering based on a single dissimilarity.

1 Introduction

Clustering algorithms such as k-means depend critically on the choice of a good dissimilarity [17]. A large variety of dissimilarities have been proposed in the literature [1]. However, in real applications no dissimilarity outperforms the others because each dissimilarity reflects often different features of the data [11]. So, instead of using a single dissimilarity it has been recommended in [10,11] to consider a linear combination of heterogeneous dissimilarities.

Several authors have proposed techniques to learn a linear combination of kernels (similarities) from the data [10,13,11,18]. These methods are designed for classification tasks and assume that the class labels are available for the training set. However, for certain applications such as Bioinformatics, domain experts provide only incomplete knowledge in the form of which pairs of proteins or genes are related [7]. This a priory information should be incorporated into semi-supervised clustering algorithms via equivalence constraints [5]. Thus, [17] proposed a distance metric learning algorithm that incorporates such similarity/dissimilarity information using a convex programming approach. The

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 375–384, 2010.

experimental results show a significant improvement in clustering results. However, the algorithm is based on an iterative procedure that is computationally intensive particularly, for high dimensional applications. To avoid this problem, [5,8,15] presented more efficient algorithms to learn a Mahalanobis metric. However, these algorithms are not able to incorporate heterogeneous dissimilarities and rely on the use of the Mahalanobis distance that may not be appropriate for certain kind of applications.

Our approach considers that the integration of dissimilarities that reflect different features of the data should help to improve the clustering results. To this aim, a linear combination of heterogeneous dissimilarities is learnt considering the relation between kernels and distances [12]. A learning algorithm is proposed to estimate the optimal weights considering the similarity/dissimilarity constraints available. The method proposed is based on a convex quadratic optimization algorithm and incorporates a smoothing term that penalizes de complexity of the family of distances avoiding overfitting.

The algorithm has been evaluated considering several benchmark UCI datasets and two human complex cancer problems using the gene expression profiles. The empirical results suggest that the method proposed improves the clustering results obtained considering a single dissimilarity and a widely used metric learning algorithm.

This paper is organized as follows: Section 2 introduces the idealized metric considered in this paper, section 3 presents the algorithm proposed to learn a combination of dissimilarities from equivalence constraints. Section 4 illustrates the performance of the algorithm using several benchmark datasets. Finally, Section 5 gets conclusions and outlines future research trends.

2 Idealized Dissimilarity: Impact on Clustering Results

Let $\{\boldsymbol{x}_i\}_{i=1}^n \in \mathbb{R}^d$ be the input patterns. We are given side-information in the form of pairs that are considered similar or dissimilar for the application at hand. Let $\mathcal{S}$ and $\mathcal{D}$ be the subset of pairs of patterns known to be similar/dissimilar defined as:

$$\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) : \boldsymbol{x}_i \text{ is similar to } \boldsymbol{x}_j\} \tag{1}$$

$$\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) : \boldsymbol{x}_i \text{ is dissimilar to } \boldsymbol{x}_j\} \tag{2}$$

Let $\{d_{ij}^l\}_{l=1}^M$ be the set of heterogeneous dissimilarities considered. Each dissimilarity can be embedded in a feature space via the empirical kernel map introduced in appendix A. Let K_{ij}^l be the kernel matrix that represents the dissimilarity matrix $(d_{ij}^l)_{i,j=1}^n$. The kernel function can be written as an inner product in feature space [14] $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$ and therefore, it can be considered a similarity measure [15].

The ideal similarity (kernel) should be defined such that it becomes large for similar patterns and small for dissimilar ones. Mathematically, the ideal kernel is defined as follows:

$$k^*_{ij} = \begin{cases} \max_l\{k^l_{ij}\} \text{ If } (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S} \\ \min_l\{k^l_{ij}\} \text{ If } (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{D} \end{cases} \tag{3}$$

The idealized kernel introduced in this paper is related to the one proposed by [2] for classification purposes: $k(x_i, x_j) = 1$ if $y_i = y_j$ and 0 otherwise, where y_i denotes the label of x_i. However, there are two differences that are worth to mention. First, the ideal kernel proposed by [2] doesn't take into account the topology and distribution of the data, missing relevant information for the identification of groups in a semi-supervised setting. Second, this kernel can be considered an extreme case of the idealized kernel defined earlier and thus, more prone to overfitting.

Considering the relation between distances and kernels [14], the idealized distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be written in terms of kernel evaluations as:

$$d^{2*}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}_j)\|^2 \tag{4}$$

$$= k^*(\boldsymbol{x}_i, \boldsymbol{x}_i) + k^*(\boldsymbol{x}_j, \boldsymbol{x}_j) - 2k^*(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{5}$$

The idealized dissimilarity, collects information from a set of heterogeneous measures adapting the metric to the problem at hand and improving the clustering results.

3 Learning a Combination of Dissimilarities from Equivalence Constraints

In this section, we present a learning algorithm to estimate the optimal weights of a linear combination of kernels from a set of similarity or dissimilarity constraints.

Let $\{k^l_{ij}\}_{l=1}^M$ be the set of kernels obtained from a set of heterogeneous dissimilarities via the empirical kernel map introduced in appendix A. If non-linear kernels with different parameter values are considered, we get a wider family of measures that includes non-linear transformations of the original dissimilarities. The kernel sought is defined as:

$$k_{ij} = \sum_{l=1}^{M} \beta_l k^l_{ij}\,, \tag{6}$$

where the coefficients are constrained to be $\beta_l \geq 0$. This non-negative constraint on the weights helps to interpret the results and guarantees that provided all the individual kernels are positive semi-definite the combination of kernels is convex and positive semi-definite [13].

The optimization problem in the primal may be formulated as follows:

$$\min_{\boldsymbol{\beta}, \boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C_S}{N_S} \sum_{(x_i, x_j) \in S} \xi_{ij} + \frac{C_D}{N_D} \sum_{(x_i, x_j) \in D} \xi_{ij} \tag{7}$$

$$\text{s. t. } \boldsymbol{\beta}^T \boldsymbol{K}_{ij} \geq K^*_{ij} - \xi_{ij} \quad \forall\, (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S} \tag{8}$$

$$\boldsymbol{\beta}^T \boldsymbol{K}_{ij} \leq K^*_{ij} + \xi_{ij} \quad \forall\, (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{D} \tag{9}$$

$$\beta_l \geq 0 \quad \xi_{ij} \geq 0 \quad \forall\, l = 1, \ldots, M \tag{10}$$

where the first term in equation (7) is a regularization term that penalizes the complexity of the family of distances, C_S and C_D are regularization parameters that give more relevance to the similarity or dissimilarity constraints. N_S, N_D are the number of pairs in $\mathcal{S}$ and $\mathcal{D}$, $\boldsymbol{K}_{ij} = [K^1_{ij}, \ldots, K^M_{ij}]^T$, K^*_{ij} is the idealized kernel matrix and ξ_{ij} are the slack variables that allows for errors in the constraints.

To solve this constrained optimization problem the method of Lagrange Multipliers is used. Then, the dual problem becomes:

$$\max_{\alpha_{ij}, \gamma} \quad -\tfrac{1}{2} \sum_{\substack{(x_i,x_j)\in\mathcal{S}\\(x_k,x_l)\in\mathcal{S}}} \alpha_{ij}\alpha_{kl}\mathbf{K}^T_{ij}\mathbf{K}_{kl} - \frac{1}{2} \sum_{\substack{(x_i,x_j)\in\mathcal{D}\\(x_k,x_l)\in\mathcal{D}}} \alpha_{ij}\alpha_{kl}\mathbf{K}^T_{ij}\mathbf{K}_{kl} \tag{11}$$

$$+ \sum_{\substack{(x_i,x_j)\in\mathcal{S},\\(x_k,x_l)\in\mathcal{D}}} \alpha_{ij}\alpha_{kl}\mathbf{K}^T_{ij}\mathbf{K}_{kl} - \sum_{(x_i,x_j)\in\mathcal{S}} \alpha_{ij}\gamma^T\mathbf{K}_{ij} - \frac{1}{2}\gamma^T\gamma \tag{12}$$

$$+ \sum_{(x_i,x_j)\in\mathcal{D}} \alpha_{ij}\gamma^T\mathbf{K}_{ij} + \sum_{(x_i,x_j)\in\mathcal{S}} \alpha_{ij}K^*_{ij} - \sum_{(x_i,x_j)\in\mathcal{D}} \alpha_{ij}K^*_{ij}\,, \tag{13}$$

subject to:

$$0 \leq \alpha_{ij} \leq \begin{cases} \frac{C_S}{N_S} & \text{for}\,(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S} \\ \frac{C_D}{N_D} & \text{for}\,(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{D} \end{cases} \tag{14}$$

$$\gamma_l \;\geq 0 \quad \forall\, l = 1, \ldots, M\,, \tag{15}$$

where α_{ij} and γ_l are the lagrange multipliers. This is a standard quadratic optimization problem similar to the one solved by the SVM. The computational burden does not depend on the dimensionality of the space and it avoids the problem of local minima.

Once the α_{ij} and γ_l are computed, the weights β_l can be obtained considering $\partial L/\partial \boldsymbol{\beta} = 0$:

$$\boldsymbol{\beta} = \sum_{(x_i,x_j)\in\mathcal{S}} \alpha_{ij}\mathbf{K}_{ij} - \sum_{(x_i,x_j)\in\mathcal{D}} \alpha_{ij}\mathbf{K}_{ij} + \boldsymbol{\gamma}\,. \tag{16}$$

The weights β_l can be substituted in equation (6) to get the optimal combination of heterogeneous kernels. Next, a kernel k-means clustering algorithm [3] is run. Notice that the learning algorithm proposed may be applied together with any clustering based on kernels or dissimilarities.

Several techniques are related to the one proposed here. In [17] it has been proposed an algorithm to learn a full or diagonal Mahalanobis metric from similarity information. The optimization algorithm is based on an iterative procedure that is more costly particularly for high dimensional problems. [5] and [8,15] have proposed more efficient algorithms to learn a Mahalanobis metric from equivalence constraints. The first one (Relevant Component Analysis), can only take into account similarity constraints. Both of them, rely solely on a Mahalanobis metric that may fail to reflect appropriately the sample proximities for certain kind of applications. Hence, they are not able to integrate heterogeneous measures that convey complementary information. Finally, [16] has proposed a modification of the maximum margin clustering that is able to learn a linear combination of kernels. However, this algorithm is unsupervised and can not incorporate a priory information in a semi-supervised way. Besides, it can not be extended to other clustering algorithms based on dissimilarities or kernels as the method proposed here.

3.1 Support Vectors and KKT Complementary Conditions

The Lagrange Multipliers determine how difficult is for the linear combination of kernels to satisfy the constraints (8)-(9). Next, we can obtain a relation between the constraints satisfaction and the value of the Lagrange Multipliers.

The Karush-Kuhn-Tucker (KKT) complementary conditions in the primal are the following:

$$\alpha_{ij}(\beta^T \mathbf{K}_{ij} - K^*_{ij} + \xi_{ij}) = 0, \qquad (x_i, x_j) \in \mathcal{S} \tag{17}$$

$$\alpha_{ij}(\beta^T \mathbf{K}_{ij} - K^*_{ij} - \xi_{ij}) = 0, \qquad (x_i, x_j) \in \mathcal{D} \tag{18}$$

$$\eta_{ij}\xi_{ij} = 0, \qquad (x_i, x_j) \in \mathcal{S},\, (x_i, x_j) \in \mathcal{D} \tag{19}$$

$$\gamma_l \beta_l = 0, \qquad \forall l = 1, \ldots, M\,. \tag{20}$$

Now it can be easily shown the following proposition:

Proposition 1. *For all* $(x_i, x_j) \in \mathcal{S}$

$$\beta^T \mathbf{K}_{ij} \begin{cases} = K^*_{ij} & 0 < \alpha_{ij} < \frac{C_S}{N_S} \\ \geq K^*_{ij} & \alpha_{ij} = 0 \\ < K^*_{ij} & \alpha_{ij} = \frac{C_S}{N_S} \end{cases}$$

For all $(x_i, x_j) \in \mathcal{D}$

$$\beta^T \mathbf{K}_{ij} \begin{cases} = K^*_{ij} & 0 < \alpha_{ij} < \frac{C_D}{N_D} \\ \leq K^*_{ij} & \alpha_{ij} = 0 \\ > K^*_{ij} & \alpha_{ij} = \frac{C_D}{N_D} \end{cases}$$

When the Lagrange Multipliers are zero, the constraints are met with a margin equal or greater than zero. The corresponding pairs will not appear in the solution. When the Lagrange Multipliers are larger than zero, the constraints may be exactly met or violated. They are the support vectors as in the SVM. This will allow to solve the optimization problem more efficiently.

4 Experimental Results

The algorithm proposed has been evaluated considering a wide range of practical problems. Table 1 shows the features of the different datasets. We have chosen problems with a broad range of signal to noise ratio (Var/Samp.), varying number of samples and classes. The first three problems correspond to benchmark datasets obtained from the UCI database `http://archive.ics.uci.edu/ml/datasets/`. The last ones aim to the identification of complex human cancer samples using the gene expression profiles. They are available from `bioinformatics2.pitt.edu`.

Table 1. Features of the different datasets considered

	Samples	Variables	Var./Samp.	Classes
Wine (UCI)	177	13	0.17	3
Ionosphere (UCI)	351	35	0.01	2
Breast Cancer (UCI)	569	32	0.056	2
Lymphoma	96	4026	41.9	2
Colon Cancer	62	2000	32	2

All the datasets have been standardised subtracting the median and dividing by the inter-quantile range.

For high dimensional problems such as gene expression datasets, dimension reduction helps to improve significantly the clustering results [9]. Therefore, for the algorithms based on a single dissimilarity we have considered different number of genes $280, 146, 101, 56$ and 34 obtained by feature selection [11]. Genes have been ranked according to the method proposed by [4]. Then, we have chosen the subset that gives rise to the smallest error. Considering a larger number of genes or even the whole set of genes does not help to improve the clustering performance. Regarding the algorithm proposed to integrate several dissimilarities, we have considered all the dissimilarities obtained for the whole set of dimensions.

The similarity/dissimilarity constraints are obtained as in [17]. $\mathcal{S}$ is generated by picking a random subset of all pairs of points sharing the same class label. The size is chosen such that the number of connected components is roughly 20% of the size of the original dataset. $\mathcal{D}$ is chosen in a similar way although the size in this case is less relevant.

Regarding the value of the parameters, the number of clusters is set up to the number of classes, C_S and C_D are regularization parameters and the optimal value is determined by cross-validation over the subset of labeled patterns. Finally, kernel k-means is restarted randomly 20 times and the errors reported are averages over 20 independent trials.

Clustering results have been evaluated considering two objective measures. The first one is the accuracy. It determines the probability that the clustering

Table 2. Accuracy for k-means clustering considering different dissimilarities. The results are averaged over twenty independent random subsets $\mathcal{S}$ and $\mathcal{D}$.

Technique	Kernel	Wine	Ionosphere	Breast	Colon	Lymphoma
k-means (Euclidean)	linear	0.92	0.72	0.88	0.87	0.90
	pol. 3	0.87	0.73	0.88	0.88	0.90
k-means (Best diss.)	linear	0.94	0.88	0.90	0.88	0.94
	pol. 3	0.94	0.88	0.90	0.88	0.93
		χ^2	Maha.	Manha.	Corr./euclid.	χ^2
Comb. dissimilarities	linear	0.94	0.90	0.92	0.89	0.95
	pol. 3	0.96	0.89	0.92	0.90	0.92
Metric learning (Xing)	linear	0.87	0.74	0.85	0.87	0.90
	pol. 3	0.51	0.74	0.86	0.88	0.90

Table 3. Adjusted RandIndex for k-means clustering considering different dissimilarities. The results are averaged over twenty independent random subsets $\mathcal{S}$ and $\mathcal{D}$.

Technique	Kernel	Wine	Ionosphere	Breast	Colon	Lymphoma
k-means (Euclidean)	linear	0.79	0.20	0.59	0.59	0.65
	pol. 3	0.67	0.21	0.60	0.59	0.65
k-means (Best diss.)	linear	0.82	0.58	0.66	0.59	0.77
	pol. 3	0.81	0.58	0.66	0.59	0.76
		χ^2	Maha.	Manha.	Corr./euclid.	χ^2
Comb. dissimilarities	linear	0.82	0.63	0.69	0.60	0.79
	pol. 3	0.85	0.60	0.69	0.63	0.73
Metric learning (Xing)	linear	0.68	0.23	0.50	0.54	0.66
	pol. 3	0.50	0.23	0.52	0.58	0.65

agrees with the “true” clustering in the sense that the pair of patterns belong to the same or different clusters. It has been defined as in [17]:

$$\text{accuracy} = \sum_{i>j} \frac{1\{1\{c_i = c_j\} = 1\{\hat{c}_i = \hat{c}_j\}\}}{0.5m(m-1)}\,, \tag{21}$$

where c_i is the true cluster label for pattern x_i, $\hat{c}_i$ is the corresponding label returned by the clustering algorithm and m is the number of patterns. One problem of the accuracy is that the expected value for two random partitions is not zero. Therefore, we have computed also the adjusted randindex defined in [6] that avoids this problem. This index is also normalized between zero and one and larger values suggest better clustering.

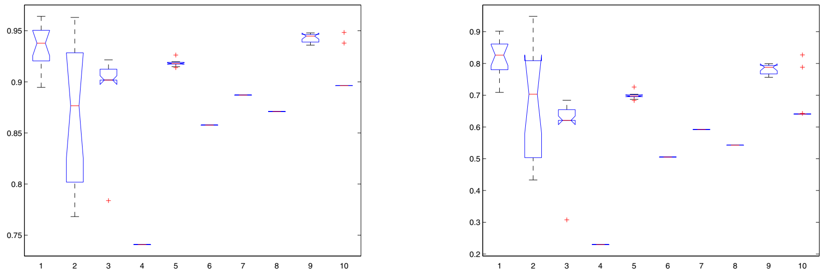
Tables 2 and 3 show the accuracy and the adjusted randindex for the clustering algorithms evaluated. We have compared with a standard metric learning strategy proposed by [17], k-means clustering based on the Euclidean distance and k-means considering the best dissimilarity out of ten different measures. Both tables indicates which is the best distance for each case.

From the analysis of tables 2 and 3, the following conclusions can be drawn:

- The combination of dissimilarities improves significantly a standard metric learning algorithm for all the datasets considered. Our method is robust to overfitting and outperforms the algorithm proposed by Xing [17] in high dimensional datasets such as Colon cancer and Lymphoma. These datasets exhibit a high level of noise. We can explain this because the algorithm based on a combination of dissimilarities allows to integrate distances computed for several dimensions discarding the noise and reducing the errors.
- The combination of dissimilarities improves usually kernel k-means based solely on the best dissimilarity. This suggests that the integration of several dissimilarities allows to extract complementary information that may help to improve the performance. Besides, the algorithm proposed always achieves at least the same performance that k-means based on the best dissimilarity. Only for lymphoma and polynomial kernel we get worst results, probably because the value assigned to the regularization parameters overfit the data. We remark that the algorithm proposed, helps to overcome the problem of choosing the best dissimilarity, the kernel and the optimal dimension. This a quite complex and time consuming task for certain applications such as Bioinformatics.

  Finally, the combination of dissimilarities improves always the standard k-means clustering based on the Euclidean measure.
- Tables 2 and 3 show that the best distance depends on the dataset considered. Moreover, we report that the performance of k-means depends strongly on the particular measure employed to evaluate the sample proximities.

Figure 1 shows a boxplot diagram for the accuracy and adjusted randindex coefficients. Odds numbers correspond to the combination of dissimilarities and the even ones to the metric learning algorithm proposed by Xing. We can see that the differences between the method proposed here and the one proposed



(a) (b)

Fig. 1. Boxplots that compare the combination of dissimilarities with the metric learning algorithm proposed by Xing according to (a) accuracy and (b) Adjusted RandIndex. All the boxplots consider linear kernels.

by Xing are statistically significant at 95% confidence level for all the datasets considered.

5 Conclusions

In this paper, we propose a semi-supervised algorithm to learn a combination of dissimilarities from equivalence constraints. The error function includes a penalty term that controls the complexity of the family of distances considered and the optimization is based on a robust quadratic programming approach that does not suffer from the problem of local minima.

The experimental results suggest that the combination of dissimilarities improves almost always the performance of clustering algorithms based solely on a single dissimilarity. Besides, the algorithm proposed improves significantly a standard metric learning algorithm for all the datasets considered in this paper and is robust to overfitting.

Future research trends will focus on the application of this formalism to the integration of heterogeneous data sources.

Appendix A

This appendix introduces shortly the Empirical Kernel Map that allow us to work with non-Euclidean dissimilarities considering kernel methods [12].

Let d: $\mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a dissimilarity and $R = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ a subset of representatives drawn from the training set. Define the mapping $\phi : \mathcal{F} \rightarrow \mathbb{R}^n$ as:

$$\phi(z) = D(z, R) = [d(z, \boldsymbol{x}_1), d(z, \boldsymbol{x}_2), \ldots, d(z, \boldsymbol{x}_n)] \tag{22}$$

This mapping defines a dissimilarity space where feature i is given by $d(., \boldsymbol{x}_i)$. The kernel of dissimilarities can be defined as the dot product of two dissimilarity vectors in feature space.

$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle = \sum_{i=1}^{n} d(\boldsymbol{x}, p_i) d(\boldsymbol{x}', p_i) \quad \forall \boldsymbol{x},\, \boldsymbol{x}' \in \mathcal{X}\,. \tag{23}$$

References

1. Cox, T.F., Cox, M.A.A.: Multidimensional scaling, 2nd edn. Chapman & Hall/CRC, USA (2001)
2. Cristianini, N., Kandola, J., Elisseeff, J., Shawe-Taylor, A.: On the kernel target alignment. Journal of Machine Learning Research 1, 1–31 (2002)
3. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
4. Dudoit, S., Fridlyand, J., Speed, T.P.: Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data. Journal of the American Statistical Association 97(457), 77–87 (2002)

5. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning a Mahalanobis Metric from Equivalence Constraints. Journal of Machine Learning Research 6, 937–965 (2005)
6. Hubert, L., Arabie, P.: Comparing Partitions. Journal of Classification, 193–218 (1985)
7. Huang, D., Pan, W.: Incorporating Biological Knowledge into Distance-Based Clustering Analysis of Microarray Gene Expression Data. Bioinformatics 22(10), 1259–1268 (2006)
8. Kwok, J.T., Tsang, I.W.: Learning with Idealized Kernels. In: Proceedings of the Twentieth International Conference on Machine Learning, Washington DC, pp. 400–407 (2003)
9. Jeffery, I.B., Higgins, D.G., Culhane, A.C.: Comparison and Evaluation Methods for Generating Differentially Expressed Gene List from Microarray Data. BMC Bioinformatics 7(359), 1–16 (2006)
10. Lanckriet, G., Cristianini, N., Barlett, P., El Ghaoui, L., Jordan, M.: Learning the kernel matrix with semidefinite programming. Journal of Machine Learning Research 3, 27–72 (2004)
11. Martín-Merino, M., Blanco, A., De Las Rivas, J.: Combining Dissimilarities in a Hyper Reproducing Kernel Hilbert Space for Complex Human Cancer Prediction. Journal of Biomedicine and Biotechnology, 1–9 (2009)
12. Pekalska, E., Paclick, P., Duin, R.: A generalized kernel approach to dissimilarity-based classification. Journal of Machine Learning Research 2, 175–211 (2001)
13. Soon Ong, C., Smola, A., Williamson, R.: Learning the kernel with hyperkernels. Journal of Machine Learning Research 6, 1043–1071 (2005)
14. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons, New York (1998)
15. Wu, G., Chang, E.Y., Panda, N.: Formulating distance functions via the kernel trick. In: ACM SIGKDD, Chicago, pp. 703–709 (2005)
16. Zhao, B., Kwok, J.T., Zhang, C.: Multiple Kernel Clustering. In: Proceedings of the Ninth SIAM International Conference on Data Mining, Nevada, pp. 638–649 (2009)
17. Xing, E., Ng, A., Jordan, M., Russell, S.: Distance Metric Learning, with Application to Clustering with Side-Information. In: Advances in Neural Information Processing Systems, vol. 15, pp. 505–512. MIT Press, Cambridge (2003)
18. Xiong, H., Chen, X.-W.: Kernel-Based Distance Metric Learning for Microarray Data Classification. BMC Bioinformatics 7(299), 1–11 (2006)

On Consensus Clustering Validation

João M.M. Duarte[1,2], Ana L.N. Fred[1], André Lourenço[1], and F. Jorge F. Duarte[2]

[1] Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa, Portugal
{jduarte,afred,arlourenco}@lx.it.pt
[2] GECAD - Knowledge Engineering and Decision Support Group,
Instituto Superior de Engenharia do Porto, Porto, Portugal
{jod,fjd}@isep.ipp.pt

Abstract. Work on clustering combination has shown that clustering combination methods typically outperform single runs of clustering algorithms. While there is much work reported in the literature on validating data partitions produced by the traditional clustering algorithms, little has been done in order to validate data partitions produced by clustering combination methods. We propose to assess the quality of a consensus partition using a pattern pairwise similarity induced from the set of data partitions that constitutes the clustering ensemble. A new validity index based on the likelihood of the data set given a data partition, and three modified versions of well-known clustering validity indices are proposed. The validity measures on the original, clustering ensemble, and similarity spaces are analysed and compared based on experimental results on several synthetic and real data sets.

1 Introduction

Clustering ensemble approaches have been proposed aiming to improve data clustering robustness and quality [1], reuse clustering solutions [2], and cluster data in a distributed way. Schematically, these methods can be split into two main phases: the construction of the clustering ensemble (CE); and the combination of information extracted from the CE into a consensus partition. The Evidence Accumulation Clustering method (EAC) [1] additionally produces, as an intermediate result, a learned pairwise similarity between patterns, summarized in a co-association matrix. In the literature on this topic, one can find many alternative ways of building the clustering ensemble, defining the combination strategy and extraction algorithm, and choosing the final number of clusters. All these lead to a myriad of alternative clustering solutions. Hence, we are faced with the following problem: "*for a given data set, which clustering solution should be selected?*".

While there is much work reported in the literature on validating data partitions produced by the traditional clustering algorithms [3], little has been done in order to validate data partitions produced by clustering combination methods. Most of the reported works use measures of consistency between consensus solutions and the clustering ensemble, such as Average Normalized Mutual Information [2] and Average Cluster Consistency [4]. The classical validity indices may also be used to assess the quality of the consensus partition. This requires the original data representation to be available,

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 385–394, 2010.

which may not always be possible. Also, not considering clustering ensemble information should be a drawback, since the clustering structure, used by the clustering combination methods to produce the consensus partitions, is not used.

In this paper we propose the validation of clustering combination results at three levels:

- *original data representation* – measure the consistency of clustering solutions with the structure of the data, perceived from the original representation (either feature-based or similarity-based);
- *clustering ensemble level* – measure the consistency of consensus partitions with the clustering ensemble;
- *learned pairwise similarity* – measure the coherence between clustering solutions and the co-association matrix induced by the clustering ensemble.

Additionally to the methodology of evaluation at these distinct levels, we propose a new criterion based on likelihood estimates, and adaptation of "classical" cluster validity measures to pairwise similarity representations.

The remaining of the paper is organized as follows. Section 2 formulates the clustering ensemble problem, and describes the EAC method, that will be used in our experiments. The methodology for the validation of consensus partitions is presented in section 3. In section 4, a new validity index based on pairwise similarities is proposed. Experiments comparing all the validation measures are presented in section 5. Finally, the conclusions appear in section 6.

2 Clustering Combination

Let $\mathcal{X} = \{x_1, \cdots, x_n\}$ be a data set with n data patterns. Different partitions of $\mathcal{X}$ can be obtained by using different clustering algorithms, changing parameters and/or initializations for the same clustering algorithm, using different subsets of data features or patterns, projecting $\mathcal{X}$ to subspaces, and combinations of these. A clustering ensemble, $\mathcal{P}$, is defined as a set of N data partitions of $\mathcal{X}$:

$$\mathcal{P} = \{P^1, \cdots, P^N\}, \;\; P^l = \{C^l_1, \cdots, C^l_{K^l}\}, \tag{1}$$

where C^l_k is the k^{th} cluster in data partition P^l, which contains K^l clusters. Different partitions capture different views of the structure of the data. Clustering ensemble methods use a consensus function f which maps a clustering ensemble $\mathcal{P}$ into a consensus partition $P^* = f(\mathcal{P})$.

The Evidence Accumulation Clustering method (EAC) [1] considers each data partition $P^l \in \mathcal{P}$ as an independent evidence of data organization. The underlying assumption of EAC is that two patterns belonging to the same "natural" cluster will be frequently grouped together. A vote is given to a pair of patterns every time they co-occur in the same cluster. Pairwise votes are stored in a $n \times n$ co-association matrix, $\mathbf{C}$, normalized by the total number of combined data partitions, i.e., $\mathbf{C}_{ij} = \frac{\sum_{l=1}^{N} vote^l_{ij}}{N}$ where $vote^l_{ij} = 1$ if x_i and x_j co-occur in a cluster of data partition P^l; otherwise $vote^l_{ij} = 0$. The consensus partition is obtained by applying some clustering algorithm over the co-association matrix, $\mathbf{C}$.

3 Consensus Partition Validation

We herein propose the assessment of the quality of a consensus partition, P^*, by measuring its consistency at three levels: the original representation space; the clustering ensemble; and the learned pairwise similarity.

3.1 Validity Measures on the Original Data Space

Validity measures on the original data space are the most common approaches to perform clustering validation. The basic idea consists of evaluating a data partition using a utility or cost function, and comparing it with other partitions of the same data set. The utitlity/cost function usually measures the intra-cluster compactness and inter-cluster separation of a given data partition. Many different validity measures on the original data representation space have been proposed in the literature [3]. In this paper we will focus on three of them: the Silhouette, Dunn's and Davies-Bouldin indices.

Let $\mathcal{X} = \{x_1, \cdots, x_n\}$ be the data set, $P = \{C_1, \cdots, C_K\}$ its partition into K clusters, and $|C_l|$ the number of elements in the l-th cluster. Let $d(x_i, x_j)$ be the dissimilarity (distance) between data patterns x_i and x_j.

The Silhouette index [5] is formally defined as follows. Let a_i denote the average distance between $x_i \in C_l$ and the other patterns in the same cluster, and b_i the minimum average distance between x_i and all patterns grouped in another cluster:

$$a_i = \frac{1}{|C_l| - 1} \sum_{\substack{x_j \in C_l \\ j \neq i}} d(x_i, x_j), \quad b_i = \min_{k \neq l} \frac{1}{|C_k|} \sum_{x_j \in C_k} d(x_i, x_j). \tag{2}$$

The silhouette width, s_i, for each x_i, produces a score in the range $[-1, 1]$ indicating how well x_i fits in its own cluster when compared to other clusters; the global Silhouette index, S, is given by the average silhouette width computed over all samples in the data set:

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}, \qquad S = \frac{1}{n} \sum_{i=1}^{n} s_i \tag{3}$$

Dunn's index, quantifying how well a set of clusters represent compact and separated clusters [6], is defined as:

$$D = \frac{\min\limits_{1 \leq q \leq K} \min\limits_{1 \leq r \leq K, r \neq q} \mathrm{dist}(C_q, C_r)}{\max\limits_{1 \leq p \leq K} \mathrm{diam}(C_p)} \tag{4}$$

where $\mathrm{dist}(C_q, C_r)$ represents the distance between clusters C_q and C_r, and $\mathrm{diam}(C_p)$ is the p^{th} cluster diameter:

$$\mathrm{dist}(C_q, C_r) = \min_{x_i \in C_q,\ x_j \in C_r} d(x_i, x_j), \quad \mathrm{diam}(C_p) = \max_{x_i,\ x_j \in C_p} d(x_i, x_j). \tag{5}$$

The best partition is the one that maximizes the index value, D.

Davies-Bouldin index [7], is defined as the ratio of the sum of within-cluster scatter and the value of between-cluster separation:

$$DB = \frac{1}{K}\sum_{k=1}^{K} \max_{m \neq k} \left\{ \frac{\Delta(C_k) + \Delta(C_m)}{d(\nu_k, \nu_m)} \right\}, \quad \Delta(C_k) = \frac{\sum_{x_i \in C_k} d(x_i, \nu_k)}{|C_k|} \tag{6}$$

where $\Delta(C_k)$ is the average distance between all patterns in C_k and their cluster center $\nu_k = \frac{\sum_{x_i \in C_k} x_i}{|C_k|}$. Small values of DB correspond to clusters that are compact, and whose centers are far away from each other. The data partition that minimizes DB is the optimal one.

3.2 Validity Measures on the Clustering Ensemble Space

These validity indices rely on the agreement between the consensus partition, P^*, and the partitions in the clustering ensemble $\mathcal{P} = \{P_1, \cdots, P_N\}$.

Let $H(P) = -\sum_{k=1}^{K} p(k) \log p(k)$ be the entropy of data partition P, with $p(k) = \frac{n_k}{n}$, and n_k the number of patterns in the k^{th} cluster of P. The mutual information between two data partitions, P^* and P^l, is defined as:

$$MI(P^*, P^l) = \sum_{i}^{K^*} \sum_{j}^{K^l} \frac{p(i,j)}{p(i)p(j)}, \tag{7}$$

with $p(i,j) = \frac{1}{n}|C_i^* \cap C_j^l|$, the fraction of shared samples in clusters C_i^* and C_j^l. Strehl and Ghosh [2] define the Average Normalized Mutual Information as:

$$ANMI(P^*, \mathcal{P}) = \frac{1}{N} \sum_{l=1}^{N} \frac{MI(P^*, P^l)}{\sqrt{H(P^*)H(P^l))}}. \tag{8}$$

Higher values of $ANMI(P^*, \mathcal{P})$ suggest better quality consensus partitions.

The Average Cluster Consistency [4] (ACC) is another validity measure based on the similarity between the partitions of the clustering ensemble and the consensus partition. The main idea consists of measuring how well the clusters C_m^l of the clustering ensemble fit in a cluster C_k^* of the consensus partition. If all patterns $x_i \in C_m^l$ belong to the same cluster C_k^*, for all clusters of the clustering ensemble, then the average cluster consistency between the consensus partition and the clustering ensemble is perfect. The ACC measures the similarity between two partitions, P^* and P^l, based on a weighting of shared samples in matching clusters:

$$\mathrm{sim}(P^*, P^l) = \frac{1}{n} \sum_{m=1}^{K^l} \max_{1 \leq k \leq K^*} |C_k^* \cap C_m^l| \left(1 - \frac{|C_k^*|}{n}\right), \tag{9}$$

where $K^l \geq K^*$. Note that cluster intersection, $|C_k^* \cap C_m^l|$, is weighted by $(1 - \frac{|C_k^*|}{n})$ in order to prevent high similarity values in situations where P^* has a few clusters with almost all the data patterns. The drawback is that consensus partitions with balanced cluster cardinality are preferred. The ACC is defined as the average similarity between each data partition in the clustering ensemble ($P^l \in \mathcal{P}$) and the consensus partition P^*:

$$ACC(P^*, \mathcal{P}) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{sim}(P^i, P^*). \tag{10}$$

From a set of possible choices, the *best* consensus partition is the one that achieves the highest $ACC(P^*, \mathcal{P})$ value.

3.3 Validity Measures on a Similarity Space

In the following, modifications of the validity indices presented in subsection 3.1 are proposed, aiming to accommodate the same principles to a pairwise similarity representation. Consider a pairwise similarity measure $s(x_i, x_j)$ between pairs of patterns (x_i, x_j). In this paper, we will define $s(x_i, x_j) = \mathbf{C}_{ij}$, the pairwise similarity induced from the clustering ensemble [1], summarized in matrix $\mathbf{C}$ (see section 2).

In order to compute a Silhouette-like validity index in a similarity space, we propose to measure the within-cluster compactness and the inter-cluster separability adapting the formulas defined in equation 2 as below:

$$a_{s_i} = \frac{1}{|C_l| - 1} \sum_{\substack{x_j \in C_l \\ j \neq i}} s(x_i, x_j), \quad b_{s_i} = \max_{k \neq l} \frac{1}{|C_k|} \sum_{x_j \in C_k} s(x_i, x_j). \tag{11}$$

While in equation 2 low values for a_i and high values for b_i corresponded to high cluster compactness and separation, in equation 11 it is the opposite since we are using similarities. In this case, high values for a_{s_i} and low values for b_{s_i} imply good data partitions. For this reason, the numerator of equation 3 (left) is changed for the computation of the silhouette width, being defined as:

$$s_{s_i} = \frac{a_{s_i} - b_{s_i}}{\max\{a_{s_i}, b_{s_i}\}}. \tag{12}$$

The average silhouette width using similarities is then computed as $S_s = \frac{1}{n} \sum_{i=1}^{n} s_{s_i}$.

For Dunn's index, the similarity between the q^{th} and the r^{th} clusters, and the diameter of C_p were redefined:

$$\mathrm{sim}(C_q, C_r) = \max_{x_i \in C_q, x_j \in C_r} s(x_i, x_j), \quad \mathrm{diam_s}(C_p) = \min_{x_i, x_j \in C_p} s(x_i, x_j). \tag{13}$$

By the fact that we are using similarities instead of distances, we take the inverse of equation 4 to define a Dunn-like validation index:

$$D_s = \frac{\min\limits_{1 \leq p \leq K} \mathrm{diam_s}(C_p)}{\max\limits_{1 \leq q \leq K} \max\limits_{1 \leq r \leq K, r \neq q} \mathrm{sim}(C_q, C_r) + 1}. \tag{14}$$

Since the information regarding the cluster centers $\{\nu_1, \cdots, \nu_K\}$ is not available in a similarity-based data representation, in our adaptation of the Davies and Bouldin's validity index, it was necessary to introduce a new concept of center of a cluster. In order to incorporate pairwise similarities instead of the original vectorial data representation, we estimate the central pattern ν_k of cluster C_k as the element with maximum mean similarity within each cluster (innermost pattern), as defined below.

$$\nu_k = \arg\max_{x_i \in C_k} \sum_{\substack{x_j \in C_k \\ j \neq i}} s(x_i, x_j), \tag{15}$$

Davies and Bouldin's validity index is redefined as

$$DB_s = \frac{1}{K} \sum_{k=1}^{K} \max_{m \neq k} \left\{ \frac{s(\nu_k, \nu_m)}{\varDelta_s(C_k) + \varDelta_s(C_m)} \right\}, \tag{16}$$

where $\varDelta_s(C_k)$ is the average similarity between all patterns in C_k.

4 Statistical Validity Index Based on Pairwise Similarity

We now propose a new validity index to assess the quality of P^* based on the likelihood of the data constrained to the data partition, $L(\mathcal{X}|\mathrm{P}^*)$, assessed from pairwise similarities, as per in the co-association matrix, **C**, defined in section 2.

Our work is inspired in the Parzen-window density estimation technique [8] with variable size window, also known as K-nearest neighbor density estimation. This technique estimates the probability density of pattern x, $p(x)$, within a region R with volume V_R. The volume R is defined as a function of the K_N nearest neighbors of x, i.e., V_R is the volume enclosed by the region that contains all the K_N nearest neighbors of x. The probability density $p(x)$ is estimated as $\hat{p}(x) = \frac{K_N}{nV_R}$.

The new validity measure based on the likelihood of the data $\mathcal{X}$ (assuming $x \in \mathcal{X}$ to be independent and identically-distributed random variables) given a partition P, is defined as:

$$L(\mathcal{X}|P) = \prod_{i=1}^{N} p(x_i|P), \quad p(x_i|P) = \sum_{k=1}^{K} p(x_i|C_k \in P) \cdot \Pr(C_k). \tag{17}$$

Following the idea behind the Parzen-window density estimation method, we define the probability density of x_i given cluster C_k as:

$$p(x_i|C_k) = \frac{K_N}{|C_k| \cdot V_k(x_i)} \tag{18}$$

where $V_k(x_i)$ represents the volume of a sufficiently small region that contains all the patterns of the neighborhood $KNN_k(x_i) \bigcup \{x_i\}$, and $KNN_k(x_i)$ is the set of the K_N most similar data patterns to x_i in cluster C_k. Since we rely only on pairwise similarities, as induced from the clustering ensemble, we approximate the intrinsic volume $V_k(x_i)$ by a quantity proportional to it, defined by:

$$V_k(x_i) \triangleq \mathrm{diam}_k(x_i), \qquad \mathrm{diam}_k(x_i) = 2\left(1 - \min_{x_j \in KNN_k(x_i)} \mathbf{C}_{ij}\right) \tag{19}$$

where $\mathrm{diam}_k(x_i)$ represents the "diameter" of the region centered at x_i that contains the neighborhood of x_i. Since the similarity matrix, $\mathbf{C}$, takes values in the interval $[0; 1]$, the above transformation $1 - \mathbf{C}_{ij}$ leads to a dissimilarity measure; the diameter thus corresponds to twice the dissimilarity of the K_N^{th} nearest neighbor of x_i.

Using equations 17-18 and estimating $\Pr(C_k)$ as $\frac{1}{n}|C_k|$, the likelihood of the data set $\mathcal{X}$ given a data partition P is defined as:

$$L(\mathcal{X}|P) = \prod_{i=1}^{N} \sum_{k=1}^{K} \frac{K_N}{n \cdot V_k(x_i)}. \tag{20}$$

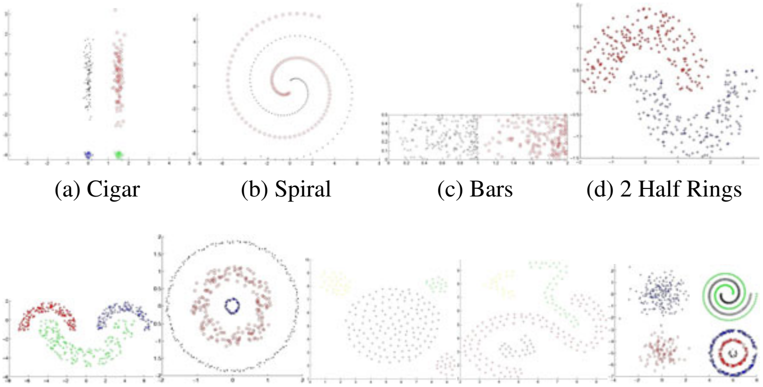
The underlying reasoning for using L as a validity index is the following.

Given a clustering ensemble, the co-association matrix, $\mathbf{C}$, corresponds to the maximum likelihood estimate of the probability of pairwise co-occurrence of patterns in a cluster. Taking this co-occurrence probability as the pattern pairwise similarity induced by the CE, the likelihood of the data set $\mathcal{X}$ given a combination partition P^* is estimated by $L(\mathcal{X}|P^*)$. The statistical validity index based on the pairwise similarity, L, thus corresponds to a goodness of fit of the combined partition, P^*, with the clustering ensemble and the pairwise information extracted from it. Best combination strategies should therefore lead to highest likelihood values, L, of the data.

In a similar way, we can compute the likelihood of the data given the combination partition using the original data representation space. In this case, the likelihood L corresponds to a goodness of fit of the combined partition, P^*, with the statistical properties of the data on the original representation. In the following we denote by L_O the likelihood computed from the original data representation, and by L_S the likelihood computed from the co-association matrix (induced similarity).

5 Experimental Results

Five real (available at the UCI repository http://archive.ics.uci.edu/ml) and nine synthetic data sets were used to assess the performance of the validity measures on a wide variety of situations, including data sets with arbitrary cluster shapes, different cardinality and dimensionality, well-separated and touching clusters, and distinct cluster densities. The Iris data set consists of 50 patterns from each of three species of iris flowers, characterized by four features. The Std Yeast is composed of 384 patterns (normalized to have 0 mean 0 and unit variance) characterized by 17 features, split into 5 clusters concerning 5 phases of the cell cycle. The Optdigits is a subset of Handwritten Digits data set containing only the first 100 patterns of each digit, from a total of 3823 data samples characterized by 64 attributes. The House Votes data set is composed of two clusters of votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the Congressional Quarterly Almanac. From a total of 435 (267 democrats and 168 republicans) only the patterns without missing values were considered, resulting in 232 patterns (125 democrats and 107 republicans). The Wine data set consists of the results of a chemical analysis of wines grown in the same region in Italy divided into three clusters with 59, 71 and 48 patterns described by 13 features. Both House Votes and Wine data sets were normalized to have unit variance. The synthetic data sets are shown in figure 1.



(a) Cigar (b) Spiral (c) Bars (d) 2 Half Rings

(e) 3 Half Rings (f) Concentric (g) D1 (h) D2 (i) Complex

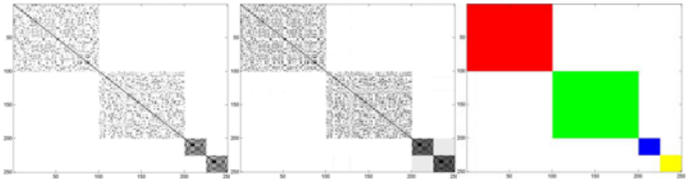
Fig. 1. Synthetic data sets

Table 1. $NMI(P^*, P^0)$ for the consensus partitions selected by each validity measure

	Clustering Ensemble Construction Method A											**Clustering Ensemble Construction Method B**										
Data Set	L_O	S_O	D_O	DB_O	L_S	S_S	D_S	DB_S	ANMI	ACC	Best	L_O	S_O	D_O	DB_O	L_S	S_S	D_S	DB_S	ANMI	ACC	Best
Iris	**0.81**	**0.81**	0.71	0.71	**0.81**	**0.81**	0.71	**0.81**	**0.81**	**0.81**	0.81	**0.81**	**0.81**	0.72	0.72	**0.81**	**0.81**	0.72	0.72	**0.81**	**0.81**	0.81
Std Yeast	0.49	0.49	0.08	**0.53**	0.49	**0.53**	0.24	0.08	0.49	0.49	0.53	0.48	0.48	0.37	0.32	0.48	**0.53**	0.23	0.48	0.48	0.48	0.53
Optdigits	**0.81**	**0.81**	0.71	0.63	**0.81**	**0.81**	0.63	**0.81**	**0.81**	**0.81**	0.81	0.81	**0.83**	**0.83**	0.72	0.81	0.81	0.72	**0.83**	0.81	**0.83**	0.83
House Votes	**0.50**	**0.50**	0.03	0.03	**0.50**	0.14	0.14	0.14	**0.50**	**0.50**	0.50	**0.49**	**0.49**	0.02	**0.49**	**0.49**	**0.49**	0.14	0.14	**0.49**	**0.49**	0.49
Wine	**0.77**	**0.77**	0.66	0.08	**0.77**	0.66	0.08	0.66	**0.77**	**0.77**	0.77	0.77	**0.80**	0.06	0.17	**0.80**	0.77	0.17	0.06	0.77	0.77	0.80
Cigar	**1.00**	**1.00**	**1.00**	0.23	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00	0.84	**1.00**	**1.00**	**1.00**	0.84	**1.00**	0.38	**1.00**	0.84	0.84	1.00
Spiral	**1.00**	0.00	0.05	0.05	**1.00**	0.00	**1.00**	**1.00**	0.00	0.00	1.00	0.01	0.01	0.08	0.08	0.01	0.01	**1.00**	**1.00**	0.01	0.01	1.00
Bars	**0.94**	**0.94**	0.06	0.06	**0.94**	**0.94**	0.06	**0.94**	**0.94**	**0.94**	0.94	**0.94**	**0.94**	0.21	0.21	**0.94**	**0.94**	0.21	0.21	**0.94**	**0.94**	0.94
2 Half Rings	**0.99**	**0.99**	0.17	0.17	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	0.99	0.87	**0.99**	0.21	0.21	0.87	0.87	**0.99**	**0.99**	0.87	0.87	0.99
3 Half Rings	**1.00**	**1.00**	0.08	0.08	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00
Concentric	**1.00**	**1.00**	0.09	0.09	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00	0.70	**1.00**	0.14	0.14	0.70	0.70	**1.00**	**1.00**	0.70	0.70	1.00
D1	**1.00**	**1.00**	**1.00**	0.05	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00	0.40	**1.00**	**1.00**	**1.00**	0.40	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00
D2	**1.00**	**1.00**	0.14	0.14	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00	0.57	0.71	0.34	0.34	0.57	0.57	**1.00**	**1.00**	0.71	0.71	1.00
Complex	**0.83**	0.44	**0.83**	0.44	**0.83**	**0.83**	0.82	0.82	**0.83**	**0.83**	0.83	**0.70**	**0.70**	**0.70**	0.63	**0.70**	0.63	0.56	**0.70**	**0.70**	**0.70**	0.87
#Best criterion	13	11	3	1	13	11	7	10	12	12		5	11	5	4	6	7	6	9	6	7	

For each data set, two different methods were used to build the clustering ensembles. In the first method (A), the K-means algorithm was used to produce $N = 150$ data partitions, each one with exactly $K = 20$ clusters for the Iris data set, $K = 50$ for the Concentric data sets, $K = 120$ for the Complex data set, and $K = 30$ for all the other data sets. In the second method (B), the K-means algorithm was also used to build clustering ensembles with the same size, but the number of clusters for each data partition was randomly chosen to be an integer in the interval $[10; 30]$. The clustering ensemble construction method A (leading to $\mathcal{P}^A$) is expected to be a "good" clustering ensemble, in the sense that its clusters have less probability of mixing patterns from different "natural" clusters than the clustering ensemble construction method B ($\mathcal{P}^B$), since $K^l, \forall P_l \in \mathcal{P}^A$ is always higher than $\min_{P^l \in \mathcal{P}^B} K^l$, with the exception of the Iris data set. The consensus partitions were obtained applying the EAC method using the Single-Link, Average-Link, Complete-Link, Centroid-Link and Ward-Link hierarchical clustering algorithms at the final step. K_N was defined as $\lceil \sqrt{n} \rceil$.

Table 1 shows the $NMI(P^*, P^0)$ values between the best data partition P^*, according to each validity measure, and the "real" (ground-truth) data partition P^0. The



(a) (b) (c)

Fig. 2. Co-association matrices for (a) CE construction method A, (b) CE construction method B and (c) "natural" partition of data, for Cigar data set

subscripts $_o$ and $_s$ point out that the validity measure was evaluated on the original space or the similarity space, respectively. The columns designated by "Best" indicate the value of NMI for the best obtained consensus partition. In order to use the criterion based on the likelihood estimates (L) on the original space, the diameter of a region was computed as $\text{diam}(x_i) = 2 \max_{x_j \in KNN(x_i)} d(x_i, x_j)$, using the Euclidean distance to measure dissimilarity, and $KNN(x_i)$ corresponds to the set of the K_N closest patterns to x_i. The results for the clustering ensemble construction method A show that the L validity measure had the best performance, both on the original and similarity spaces, selecting the best consensus partition in 13 out of 14 data sets, followed by $ANMI$ and ACC criteria with 12, and S_O and S_S with 11. While L_o and L_s selected the same partitions, S_o and S_s had different choices on several data sets. The performances of D and DB were better on the pairwise similarity space than on the original space, suggesting that the first should be preferred. For the clustering ensemble construction method B, S on the original space was the best validity measure, being the best criterion in 11 data sets. DB was the best on the similarity space by selecting in 9 data sets equal or better partitions than the other indices. L was the best criterion only 5 times on the original space and 6 on the similarity space. The poor performance of L is due to its sensibility to "bad" clustering ensembles. Figure 2 shows the co-association matrices for construction methods A and B and the "natural" partition for the Cigar data set. While in figure 2 (a) there are no co-associations between patterns belonging to different "natural" clusters, in figure 2 (b) it can be seen (especially on the lower right corner) that some patterns from distinct "natural" clusters have co-association different from 0. This explains why the L_s performed correctly on the clustering ensemble construction method A and not on B.

From the comparison involving the criteria on the original and similarity spaces, we conclude that L (on both spaces) is the best choice if the clustering ensemble is "good", S is robust on the original space, and D was the worst criterion (despite that the similarity space version presents better results than the original space version). We also conclude that the consensus partition evaluation may also be restricted to the co-association matrix. This has the advantages of exploring sparse similarities representations (particularly when using L_s) and complying with data privacy. Evaluating consensus partitions on the original space has also another disadvantage: *how to validate a consensus partition if the partitions belonging to the clustering ensemble were produced using different representations (e.g. distinct subset of feature, random projections, etc)?*

By comparing the criteria on the similarity spaces with the criteria based on the consistency between the clustering ensemble partitions and the consensus partition, L_s was better than ANMI and ACC in construction method A, and DB_s was better in construction method B; so we can discard both ANMI and ACC, and rely instead on the similarity-based criteria in order to assess the consensus partitions.

6 Conclusions

The validation of clustering solutions were proposed at three distinct levels: original data representation, learned pairwise similarity, and consistency with the clustering ensemble partitions. A new validity measure based on the likelihood estimation of pattern pairwise co-occurrence probabilities was introduced. Experimental results seem to indicate that: the new validity measure is a good choice for performing consensus clustering validation when the clusters belonging to the clustering ensemble are not likely to contain patterns of different "natural" clusters; the learned similarity-based criteria can be used, instead of the traditional clustering ensemble measures; and the similarity-based criteria are a good option when the original data representation is not available. More extensive evaluation of the validity indices is being conducted over a larger number of data sets and on the comparison of consensus results produced by different combination strategies.

Acknowledgments

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). This work was partially supported by the Portuguese Foundation for Science and Technology (FCT), Portuguese Ministry of Science and Technology, under grant PTDC/EIACCO/103230/2008.

References

1. Fred, A., Jain, A.: Combining multiple clustering using evidence accumulation. IEEE Trans. Pattern Analysis and Machine Intelligence 27(6), 835–850 (2005)
2. Strehl, A., Ghosh, J.: Cluster ensembles — a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. 3, 583–617 (2003)
3. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. Intelligent Information Systems Journal 17(2-3), 107–145 (2001)
4. Duarte, F.J., Duarte, J.M.M., Rodrigues, M.F.C., Fred, A.L.N.: Cluster ensemble selection using average cluster consistency. In: KDIR 2009: Proc. of Int. Conf. on Knowledge Discovery and Information Retrieval (October 2009)
5. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20, 53–65 (1987)
6. Dunn, J.C.: A fuzzy relative of the isodata process and its use in detecting compact, well separated clusters. Cybernetics and Systems 3(3), 32–57 (1974)
7. Davies, D., Bouldin, D.: A cluster separation measure. IEEE Transaction on Pattern Analysis and Machine Intelligence 1(2) (1979)
8. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, Hoboken (November 2000)

Pairwise Probabilistic Clustering Using Evidence Accumulation

Samuel Rota Bulò[1], André Lourenço[3], Ana Fred[2,3], and Marcello Pelillo[1]

[1] Dipartimento di Informatica - University of Venice - Italy
{srotabul,pelillo}@dsi.unive.it
[2] Instituto Superior Técnico - Lisbon - Portugal
[3] Instituto de Telecomunicações - Lisbon - Portugal
{arlourenco,afred}@lx.it.pt

Abstract. In this paper we propose a new approach for consensus clustering which is built upon the evidence accumulation framework. Our method takes the co-association matrix as the only input and produces a soft partition of the dataset, where each object is probabilistically assigned to a cluster, as output. Our method reduces the clustering problem to a polynomial optimization in probability domain, which is attacked by means of the Baum-Eagon inequality. Experiments on both synthetic and real benchmarks data, assess the effectiveness of our approach.

1 Introduction

There is a close connection between the concepts of pairwise similarity and probability in the context of unsupervised learning. It is a common assumption that, if two objects are similar, it is very likely that they are grouped together by some clustering algorithm, the higher the similarity, the higher the probability of co-occurrence in a cluster. Conversely, if two objects co-occur very often in the same cluster (high co-occurrence probability), then it is very likely that they are very similar. This duality and correspondence between pairwise similarity and pairwise probability within clusters forms the core idea of the clustering ensemble approach known as evidence accumulation clustering (EAC) [1].

Evidence accumulation clustering combines the results of multiple clusterings into a single data partition by viewing each clustering result as an independent evidence of pairwise data organization. Using a pairwise frequency count mechanism amongst a clustering committee, the method yields, as an intermediate result, a co-association matrix that summarizes the evidence taken from the several members in the clustering ensemble. This matrix corresponds to the maximum likelihood estimate of the probability of pairs of objects being in the same group, as assessed by the clustering committee. One of the main advantages of EAC is that it allows for a big diversification within the clustering committee. Indeed, no assumption is made about the algorithms used to produce the data partitions, it is robust to incomplete information, i.e., we may include partitions over sub-sampled versions of the original data set, and no restriction is made on the number of clusters of the partitions.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 395–404, 2010.

Once a co-association matrix is produced according to the EAC framework, a consensus clustering is obtained by applying a clustering algorithm, which typically induces a hard partition, to the co-association matrix. Although having crisp partitions as baseline for the accumulation of evidence of data organization is reasonable, this assumption is too restrictive in the phase of producing a consensus clustering. This is for instance the case for many important applications such as clustering micro-array gene expression data, text categorization, perceptual grouping, labeling of visual scenes and medical diagnosis. In fact, the importance of dealing with overlapping clusters has been recognized long ago [2] and recently, in the machine learning community, there has been a renewed interest around this problem [3,4]. Moreover, by inducing hard partitions we loose important information like the level of uncertainty of each label assignment. It is also worth considering that the underlying clustering criteria of ad hoc algorithms do not take advantage of the probabilistic interpretation of the computed similarities, which is an intrinsic part of the EAC framework.

In this paper we propose a new approach for consensus clustering which is built upon the evidence accumulation framework. Our idea was inspired by a recent work due to Zass and Sashua [5]. Our method takes the co-association matrix as the only input and produces a soft partition of the data set, where each object is probabilistically assigned to a cluster, as output. In order to find the unknown cluster assignments, we fully exploit the fact that each entry of the co-association matrix is an estimation of the probability of two objects to be in a same cluster, which is derived from the ensemble of clusterings. Indeed, it is easy to see that under reasonable assumptions, the probability that two objects i and j will occur in the same cluster is a function of the unknown cluster assignments of i and j. By minimizing the divergence between the estimation derived from the co-association matrix and this function of the unknowns, we obtain the result of the clustering procedure. More specifically, our method reduces the clustering problem to a polynomial optimization in the probability domain, which is attacked by means of the Baum-Eagon inequality [6]. This inequality, indeed, provides us with a class of nonlinear transformations that serve our purpose. In order to assess the effectiveness of our findings we conducted experiments on both synthetic and real benchmark data sets.

2 A Probabilistic Model for Clustering

Let $O = \{1, \dots, n\}$ be a set of data objects (or simply objects) to cluster into K classes and let $\mathcal{E} = \{cl_i\}_{i=1}^{N}$ be an ensemble of N clusterings of O obtained by running different algorithms with different parameterizations on (possibly) sub-sampled versions of the original data set O. Data sub-sampling is herein put forward as a most general framework for the following reasons: it favors the diversification of the clustering ensemble; it models situations of distributed clustering where local clusterers have only partial access to the data; by using this type of data perturbation, the co-association matrix has an additional interpretation of pairwise stability that can further be used for the purpose of cluster validation [7].

Each clustering in the ensemble $\mathcal{E}$ is a function $cl_i : O_i \to \{1, \dots, K_i\}$ from the set of objects $O_i \subseteq O$ to a class label. For the afore-mentioned reasons, O_i is a subset of the original data set O and, moreover, each clustering may assume a different number of classes K_i. We denote by Ω_{ij} the indices of the clusterings where i and j have been classified, which is given by

$$\Omega_{ij} = \{p = 1 \dots N \; : \; i, j \in O_p\} \; .$$

Consider also $N_{ij} = |\Omega_{ij}|$, where $| \cdot |$ provides the cardinality of the argument, which is the number of clusterings where i and j have been both classified.

The aim of our work is to learn, from the ensemble of clusterings $\mathcal{E}$, how to cluster the objects into K classes, without having, in principle, any other information about the objects we are going to cluster. To this end, we start from the assumption that objects can be softly assigned to clusters. Hence, the clustering problem consists in estimating, for each object $i \in O$, an unknown assignment $\mathbf{y}_i$, which is a probability distribution over the set of cluster labels $\{1, \dots, K\}$, or, in other words, an element of the *standard simplex* Δ_K given by

$$\Delta_K = \{\boldsymbol{x} \in \mathbb{R}_+^K \; : \; \|\mathbf{x}\|_1 = 1\} \, ,$$

where $\mathbb{R}_+$ is the set of nonnegative reals, and $\| \cdot \|_1$ is the ℓ^1-norm. The kth entry of $\mathbf{y}_i$ thus provides the probability of object i to be assigned to cluster k. Given the unknown cluster assignments $\mathbf{y}_i$ and $\mathbf{y}_j$ of objects i and j, respectively, and assuming independent cluster assignments, the probability of them to occur in a same cluster can be easily derived as $\mathbf{y}_i^\top \mathbf{y}_j$. Suppose now $Y = (\mathbf{y}_1, \dots, \mathbf{y}_n) \in \Delta_K^n$ to be the matrix formed by stacking the $\mathbf{y}_i$'s, which in turn form the columns of Y. Then, the $n \times n$ matrix $Y^\top Y$ provides the co-occurrence probability of any pair of objects in O.

For each pair of objects i and j, let X_{ij} be a Bernoulli distributed random variable (r.v.) indicating whether objects i and j occur in a same cluster. Note that, according to our model, the mean (and therefore the parameter) of X_{ij} is $\mathbf{y}_i^\top \mathbf{y}_j$, i.e., the probability of co-occurrence of i and j. For each pair of objects i and j, we collect from the clusterings ensemble N_{ij} independent realizations $x_{ij}^{(p)}$ of X_{ij}, which are given by:

$$x_{ij}^{(p)} = \begin{cases} 1 & \text{if } cl_p(i) = cl_p(j) \, , \\ 0 & \text{otherwise} \, . \end{cases}$$

for $p \in \Omega_{ij}$. By taking their mean, we obtain the empirical probability of co-occurrence c_{ij}, which is the fraction of times objects i and j have been assigned to a same cluster:

$$c_{ij} = \frac{1}{N_{ij}} \sum_{p \in \Omega_{ij}} x_{ij}^{(p)} \; .$$

The matrix $C = (c_{ij})$, derived from the empirical probabilities of co-occurrence of any pair of objects, is known as the *co-association matrix* within the evidence accumulation-based framework for clustering [8,1]. Since C is the maximum likelihood estimate of $Y^\top Y$ given the observations from the clustering ensemble $\mathcal{E}$, we will refer to the former as the *empirical co-association matrix*, and to the latter as the *true co-association matrix*.

At this point, by minimizing the divergence, in a least-square sense, of the true co-association matrix from the empirical one, with respect to Y, we find a solution Y^* of the clustering problem. This leads to the following optimization problem:

$$\begin{aligned} Y^* = \arg\min \quad & \|C - Y^\top Y\|_F^2 \\ \text{s.t.} \quad & Y \in \Delta_K^n \,. \end{aligned} \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm. Note that Y^* provides us with soft assignments of the objects to the K classes. Indeed, y^*_{ki} gives the probability of object i to be assigned to class k. If a hard partition is needed, this can be forced by assigning each object i to the highest probability class, which is given by: $\arg\max_{k=1\dots K}\{y^*_{ki}\}$. Moreover, by computing the entropy of each $\mathbf{y}_i$, we can obtain an indication of the uncertainty of the cluster assignment for object i.

3 Related Work

In [5] a similar approach is proposed for pairwise clustering. First of all, a preprocessing on the similarity matrix W looks for its closest doubly-stochastic matrix F under ℓ_1 norm, or Frobenius norm, or relative entropy [9]. The k-clustering problem is then tackled by finding a completely-positive factorization of $F = (f_{ij})$ in the least-square sense, i.e., by solving the following optimization problem:

$$\begin{aligned} G^* = \arg\min \quad & \|F - G^\top G\|_F^2 \\ \text{s.t.} \quad & G \in \mathbb{R}_+^{k\times n} \,. \end{aligned} \tag{2}$$

Note that this leads to an optimization program, which resembles (1), but is inherently different. The elements g_{ri} of the resulting matrix G provide an indication of object i to be assigned to class r. However, unlike our formulation, these quantities are not explicit probabilities and it may happen for instance that $g_{ri} = 0$ for all $r = 1\dots k$, i.e., some objects may remain in principle unclassified.

The approach proposed to find a local solution of (2) consists in iterating the following updating rule:

$$g_{ri} \leftarrow \frac{g_{ri}\sum_{j\neq i}^n g_{rj} f_{ij}}{\sum_{s=1}^k g_{si}\sum_{j\neq i}^n g_{sj} g_{rj}} \,.$$

The computational complexity for updating all entries in G once (complete iteration) is $O(kn^2)$, while we expect to find a solution in $O(\gamma k n^2)$, where γ is the average number of complete iterations required to converge. A disadvantage of this iterative scheme is that updates must be sequential, i.e., we cannot update all entries of G in parallel.

4 The Baum-Eagon Inequality

In the late 1960s, Baum and Eagon [6] introduced a class of nonlinear transformations in probability domain and proved a fundamental result which turns out to be very useful for the optimization task at hand. The next theorem introduces what is known as the Baum-Eagon inequality.

Theorem 1 (Baum-Eagon). *Let $X = (x_{ri}) \in \Delta_k^n$ and $Q(X)$ be a homogeneous polynomial in the variables x_{ri} with nonnegative coefficients. Define the mapping $Z = (z_{ri}) = \mathcal{M}(X)$ as follows:*

$$z_{ri} = x_{ri} \frac{\partial Q(X)}{\partial x_{ri}} \bigg/ \sum_{s=1}^{k} x_{si} \frac{\partial Q(X)}{\partial x_{si}}\,, \tag{3}$$

for all $i = 1 \dots n$ and $r = 1 \dots k$. Then $Q(\mathcal{M}(X)) > Q(X)$, unless $\mathcal{M}(X) = X$. In other words $\mathcal{M}$ is a growth transformation for the polynomial Q.

This result applies to homogeneous polynomials, however in a subsequent paper, Baum and Sell [10] proved that Theorem 1 still holds in the case of arbitrary polynomials with nonnegative coefficients, and further extended the result by proving that $\mathcal{M}$ increases Q homotopically, which means that for all $0 \leq \eta \leq 1$, $Q(\eta\mathcal{M}(X) + (1-\eta)X) \geq Q(X)$ with equality if and only if $\mathcal{M}(X) = X$.

The Baum-Eagon inequality provides an effective iterative means for maximizing polynomial functions in probability domains, and in fact it has served as the basis for various statistical estimation techniques developed within the theory of probabilistic functions of Markov chains [11]. It is indeed not difficult to show that, by starting from the interior of the simplex, the fixed points of the Baum-Eagon dynamics satisfy the first-order Karush-Kuhn-Tucker necessary conditions for local maxima and that we have a strict local solution in correspondence to asymptotically stable point.

5 The Algorithm

In order to use the Baum-Eagon theorem for optimizing (1) we need to meet the requirement of having a polynomial to maximize with nonnegative coefficients in the simplex-constrained variables. To this end, we consider the following optimization program, which is proved to be equivalent to (1):

$$\begin{aligned} \max \quad & 2Tr(CY^\top Y) + \|Y^\top E_K Y\|^2 - \|Y^\top Y\|^2 \\ \text{s.t.} \quad & Y \in \Delta_K^n\,, \end{aligned} \tag{4}$$

where E_K is the $K \times K$ matrix of all 1's, and $Tr(\cdot)$ is the matrix trace function.

Proposition 1. *The maximizers of (4) are minimizers of (1) and vice versa. Moreover, the objective function of (4) is a polynomial with nonnegative coefficients in the variables y_{ki}, which are elements of Y.*

Proof. Let $P(Y)$ and $Q(Y)$ be the objective functions of (1) and (4), respectively.

To prove the second part of the proposition note that trivially every term of the polynomial $\|Y^\top Y\|^2$ is also a term of $\|Y^\top E_K Y\|^2$. Hence, $Q(Y)$ is a polynomial with nonnegative coefficients in the variables y_{ki}.

As for the second part, by simple algebra, we can write $Q(Y)$ in terms of $P(Y)$ as follows:

$$\begin{aligned} Q(Y) &= \|C\|^2 - P(Y) + \|Y^\top E_K Y\|^2 \\ &= \|C\|^2 - P(Y) + 1\,, \end{aligned}$$

where we used the fact that $\|Y^\top E_K Y\| = 1$. Note that the removal of the constant terms from $Q(Y)$ leaves its maximizers over Δ_K^n unaffected. Therefore, maximizers of (4) are also maximizers of $-P(Y)$ over Δ_K^n and thus minimizers of (1). This concludes the proof.

By Proposition 1 we can use Theorem 1 to locally optimize (4). This allows us to find a solution of (1). Note that, in our case, the objective function is not a homogeneous polynomial but, as mentioned previously, this condition is not necessary [10]. By applying (3), we obtain the following updating rule for $Y = (y_{ki})$:

$$y_{ki}^{(t+1)} = y_{ki}^{(t)} \frac{n + [Y(C - Y^\top Y)]_{ki}}{n + \sum_k y_{ki}^{(t)} [Y(C - Y^\top Y)]_{ki}}\,, \tag{5}$$

where we abbreviated $Y^{(t)}$ with Y and any non-constant iteration of (5) strictly decreases the objective function of (1).

The computational complexity of the proposed dynamics is $O(\gamma k n^2)$, where γ is the average number of iterations required to converge (note that in our experiments we kept γ fixed). One remarkable advantage of this dynamics is that it can be easily parallelized in order to benefit from modern multi-core processors. Additionally, it can be easily implemented with few lines of Matlab code.

6 Experiments

We conducted experiments on different real data-sets from the UCI Machine Learning Repository: iris, house-votes, std-yeast-cell and breast-cancer. Additionally, we considered also the image-complex synthetic data-set, shown in figure 1. For each data-set, we produced the clustering ensemble $\mathcal{E}$ by running different clustering algorithms, with different parameters, on subsampled versions of the original data-set (the sampling rate was fixed to 0.9). The clustering algorithms used to produce the ensemble were the following [12]: Single Link (SL), Complete Link (CL), Average Link (AL) and K-means (KM).

Table 1 summarizes the experimental setting that has been considered. For each data-set, we report the optimal number of clusters K and the size n of the data-set, respectively. As for the ensemble, each algorithm was run several times in order to produce clusterings with different number of classes, K_i. For each clustering approach and each parametrization of the same we generated $N = 100$ different subsampled versions of the data-set.

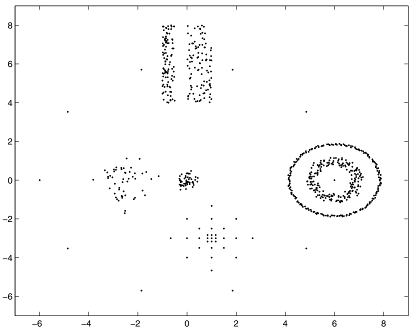


Fig. 1. Image Complex Synthetic data-set

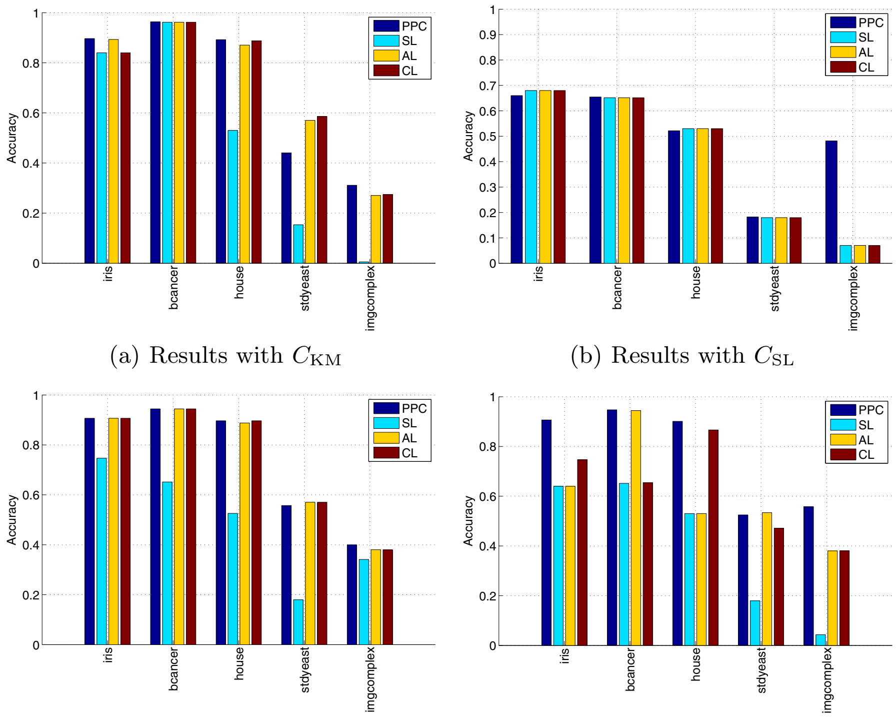
Table 1. Benchmark data-sets and parameter values used with different clustering algorithms (see text for description)

Data-Sets	K	n	Ensemble
			K_i
iris	3	150	3-10,15,20
house-votes	2	232	2-10,15,20
std-yeast-cell	5	384	5-10,15,20
breast-cancer	2	683	2-10,15,20
image-complex	8	1000	8-15,20,30, 37

Once all the clusterings have been generated, we grouped them by algorithm into several *base ensembles*, namely $\mathcal{E}_{\mathrm{SL}}$, $\mathcal{E}_{\mathrm{AL}}$, $\mathcal{E}_{\mathrm{CL}}$ and $\mathcal{E}_{\mathrm{KM}}$. Moreover, we created a large ensemble $\mathcal{E}_{\mathrm{All}}$ from the union of all of them. For each ensemble we created a corresponding co-association matrix, namely C_{SL}, C_{AL}, C_{CL}, C_{KM} and C_{All}. For each of these co-association matrices, we applied our Pairwise Probabilistic Clustering (PPC) approach, and compared it against the performances obtained with the same matrices by the agglomerative hierarchical algorithms SL, AL and CL. Each method was provided with the optimal number of classes as input parameter.

Figure 2 summarizes the results obtained over the benchmark data-sets. The performances are assessed in terms of accuracy, i.e., the percentage of correct labels. When we consider the base ensembles, i.e., $\mathcal{E}_{\mathrm{SL}}$, $\mathcal{E}_{\mathrm{AL}}$, $\mathcal{E}_{\mathrm{CL}}$ and $\mathcal{E}_{\mathrm{KM}}$, on average our approach achieves the best results, although other approaches, such as the AL, perform comparably well. Our algorithm, however, outperforms the competitors when we take the union $\mathcal{E}_{\mathrm{All}}$ of all the base ensembles into account. Interestingly, the results obtained by PPC on the combined ensemble are as good as the best one obtained in the base ensembles and, in some cases like the image-complex dataset, they are even better.

The different levels of performance obtained by the several algorithms over the different clustering ensembles, as shown in Figures 2(a) to 2(d), are illustrative of the distinctiveness between the underlying clustering ensembles, and the diversity of clustering solutions. It is then clear that the ensemble $\mathcal{E}_{\mathrm{All}}$ has



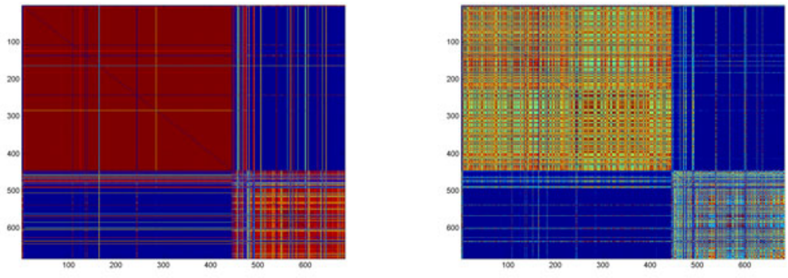
(a) Results with C_{KM}

(b) Results with C_{SL}

(c) Results with C_{AL}

(d) Results with C_{All}

Fig. 2. Experiments on benchmark data-sets

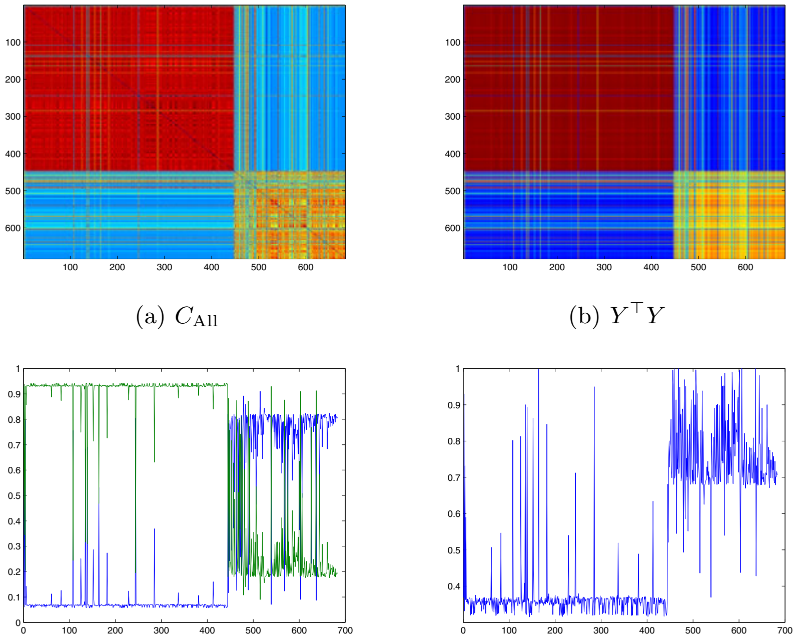


(a) C_{AL}

(b) C_{KM}

Fig. 3. Co-association matrices with ensembles $\mathcal{E}_{AL}$ and $\mathcal{E}_{KM}$

the largest diversity when compared to the individual ensembles; this is quantitatively confirmed when computing average pairwise consistency values between partitions in the individual CEs and the one resulting by the merging of these. This higher diversity causes the appearance of noisy-like structure in the



(a) C_{All}

(b) $Y^{\top}Y$

(c) Y

(d) Uncertainty

Fig. 4. Results on the breast-cancer data-set

co-association matrices. This is illustrated in Figures 3(a) and 3(b) corresponding to the co-association matrices C_{AL} and C_{KM}, respectively, when compared to the C_{All} in Figure 4(a). The better performance of the PPC algorithm on the latter CE, can be attributed to a leveraging effect over these local noisy estimates, thus better unveiling the underlying structure of the data. This is illustrated next.

Figures 4(a) and 4(b) show the empirical co-association matrix C_{All} and the true one, respectively, for the breast-cancer data-set. While the block structure of two clusters is apparent in both figures, we can see that the true co-association turns out to be less noisy than the empirical one. In Figure 4(c) we plot the soft cluster assignments, Y. Here, object indices are on the x-axis, and probabilities are on the y-axis, each curve representing the profile of a cluster. As one can see from the cluster memberships, the two clusters can be clearly evinced, although there is a higher level of uncertainty in the assignments of objects belonging to the smallest cluster. Indeed, this can also be seen in Figure 4(d), where we plot the uncertainty h_i in the cluster assignments, which is computed for each object i as the normalized entropy of $\mathbf{y}_i$, i.e.,

$$h_i = -\frac{\sum_{k=1}^{K} y_{ki} \log(y_{ki})}{\log(K)} .$$

7 Conclusion

In this paper we introduced a new approach for consensus clustering. Taking advantage of the probabilistic interpretation of the computed similarities of the the co-association matrix, derived from the ensemble of clusterings, using the Evidence Accumulation Clustering, we propose a principled soft clustering method. Our method reduces the clustering problem to a polynomial optimization in probability domain, which is attacked by means of the Baum-Eagon inequality. Experiments on both synthetic and real benchmarks assess the effectiveness of our approach.

Acknowledgement

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). This work was partially supported by the Portuguese Foundation for Science and Technology (FCT), Portuguese Ministry of Science and Technology, under grant PTDC/ EIACCO/ 103230/ 2008.

References

1. Fred, A., Jain, A.K.: Combining multiple clustering using evidence accumulation. IEEE Trans. Pattern Anal. Machine Intell. 27(6), 835–850 (2005)
2. Jardine, N., Sibson, R.: The construction of hierarchic and non-hierarchic classifications. Computer J. 11, 177–184 (1968)
3. Banerjee, A., Krumpelman, C., Basu, S., Mooney, R.J., Ghosh, J.: Model-based overlapping clustering. In: Int. Conf. on Knowledge Discovery and Data Mining, pp. 532–537 (2005)
4. Heller, K., Ghahramani, Z.: A nonparametric bayesian approach to modeling overlapping clusters. In: Int. Conf. AI and Statistics (2007)
5. Zass, R., Shashua, A.: A unifying approach to hard and probabilistic clustering. In: Int. Conf. Comp. Vision (ICCV), vol. 1, pp. 294–301 (2005)
6. Baum, L.E., Eagon, J.A.: An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. Bull. Amer. Math. Soc. 73, 360–363 (1967)
7. Fred, A., Jain, A.K.: Learning pairwise similarity for data clustering. In: Int. Conf. Patt. Recogn. (ICPR), pp. 925–928 (2006)
8. Fred, A., Jain, A.K.: Data clustering using evidence accumulation. In: Int. Conf. Patt. Recogn. (ICPR), pp. 276–280 (2002)
9. Zass, R., Shashua, A.: Doubly stochastic normalization for spectral clustering. In: Adv. in Neural Inform. Proces. Syst (NIPS), vol. 19, pp. 1569–1576 (2006)
10. Baum, L.E., Sell, G.R.: Growth transformations for functions on manifolds. Pacific J. Math. 27, 221–227 (1968)
11. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann. Math. Statistics 41, 164–171 (1970)
12. Jain, A.K., Dubes, R.C.: Algorithms for data clustering. Prentice-Hall, Englewood Cliffs (1988)

Exploring the Performance Limit of Cluster Ensemble Techniques

Xiaoyi Jiang and Daniel Abdala*

Department of Mathematics and Computer Science, University of Münster
Einsteinstrasse 62, D-48149 Münster, Germany
{xjiang,abdalad}@math.uni-muenster.de

Abstract. Cluster ensemble techniques are a means for boosting the clustering performance. However, many cluster ensemble methods are faced with high computational complexity. Indeed, the median partition methods are $\mathcal{NP}$-complete. While a variety of approximative approaches for suboptimal solutions have been proposed in the literature, the performance evaluation is typically done by means of ground truth. In contrast this work explores the question how well the cluster ensemble methods perform in an absolute sense *without ground truth*, i.e. how they compare to the (unknown) optimal solution. We present a study of applying and extending a lower bound as an attempt to answer the question. In particular, we demonstrate the tightness of the lower bound, which indicates that there exists no more room for further improvement (for the particular data set at hand). The lower bound can thus be considered as a means of exploring the performance limit of cluster ensemble techniques.

1 Introduction

Clustering, or finding partitions[1], of data is a fundamental task in multivariate data analysis. It receives increasingly importance due to the ever increasing amount of data. A large variety of clustering algorithms [20] have been proposed in the past. A recent development is constrained clustering [4], which accommodates additional information or domain knowledge. Cluster ensemble techniques provide another means for boosting the clustering performance.

Motivated by the success of multiple classifier systems, the idea of combining different clustering results emerged. Given a data set, a cluster ensemble technique consists of two principal steps: ensemble generation and consensus computation. In the first step, an ensemble (with sufficient diversity) is computed. For this purpose different clustering algorithms or the same algorithm with varying parameter settings can be applied. Other options include the use of different subsets of features and projection of the data into different subspaces. The main

* Daniel D. Abdala thanks the CNPq, Brazil-Brasilia for granting him a Ph.D. scholarship under the process number 290101-2006-9.

[1] Recently, efforts have been undertaken to go beyond the traditional understanding of clustering as partitions, i.e. [16]. This is, however, not the focus of this work.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 405–414, 2010.

challenge of cluster ensemble techniques lies in an appropriate way of computing a final clustering, which disagrees least overall with the input ensemble.

There exist two main approaches for consensus computation: co-occurrence based and median partition methods. The fundamental assumption of co-occurrence based methods is that patterns belonging to a "natural" cluster are very likely to be co-located in the same cluster in different data partitions. Therefore, a matrix with such co-location information can serve as plausibility values that two patterns should be clustered together. Typically, a subsequent step based on this matrix is designed to compute a final clustering; see for instance the evidence accumulation method [6]. Median partition methods are based on an optimization formulation of consensus computation; see for instance [17]. Since this optimization problem is typically $\mathcal{NP}$-complete [3], various suboptimal solutions have been proposed.

The focus of this work is performance assessment of cluster ensemble techniques *without using any ground truth information*. In the literature the experimental validation is typically done by means of ground truth. In contrast we explore the question how well the cluster ensemble methods perform in an absolute sense without ground truth, i.e. how they compare to the (unknown) optimal solution. This paper presents a study of applying and extending the lower bound presented in [11] as an attempt to answer the question.

2 Problem Statement

Given the data set $X = \{x_1, x_2, \ldots, x_n\}$ of n patterns x_i, a cluster ensemble is a set $P = \{P_1, P_2, \ldots, P_N\}$, where P_i is a clustering of X. We denote the set of all possible clusterings of X by $\mathcal{P}_X$ ($P \subset \mathcal{P}_X$). The goal of cluster ensemble techniques is to find a consensus clustering $P^* \in \mathcal{P}_X$, which optimally represents the ensemble P.

In median partition methods this optimality is formulated as:

$$P^* \;=\; \arg \min_{P \in \mathcal{P}_X} \sum_{i=1}^{N} d(P, P_i)$$

where $d()$ is a distance (dissimilarity) function between two clusterings. Note that this definition is a special instance of the so-called generalized median problem, which has been intensively investigated in structural pattern recognition, see [12,10] for the case of strings and graphs.

The median partition problem has been proven to be $\mathcal{NP}$-complete [3]. An exhaustive search in $\mathcal{P}_X$ is computationally intractable. In practice suboptimal approaches [14,17] are thus developed to solve the optimization problem.

Given a suboptimal solution $\tilde{P} \in \mathcal{P}_X$, however, the question of its accuracy arises. In [11] a lower bound is proposed to answer this question (for the general case of generalized median problems). For an approximate solution $\tilde{P}$ the following relationship holds:

$$\mathrm{SOD}(\tilde{P}) \;=\; \sum_{i=1}^{N} d(\tilde{P}, P_i) \;\geq\; \sum_{i=1}^{N} d(P^*, P_i) \;=\; \mathrm{SOD}(P^*)$$

where SOD stands for sum of distances. The quality of $\tilde{P}$ can be absolutely measured by the difference $\mathrm{SOD}(\tilde{P}) - \mathrm{SOD}(P^*)$. Since P^* and $\mathrm{SOD}(P^*)$ are unknown in general, we resort to a lower bound Γ with

$$0 \;\leq\; \Gamma \;\leq\; SOD(P^*) \;\leq\; SOD(\tilde{P})$$

and measure the quality of $\tilde{P}$ by $\mathrm{SOD}(\tilde{P}) - \Gamma$ instead. Obviously, the trivial lower bound $\Gamma = 0$ is useless. We require Γ to be as close to $\mathrm{SOD}(P^*)$ as possible.

In [11] a lower bound based on linear programming is proposed for metric spaces. Assuming a metric distance function $d()$, the lower bound for the median partition problem is specified by the solution Γ of the following linear program:

$$\begin{aligned}
&\text{minimize } x_1 + x_2 + \cdots + x_N \text{ subject to} \\
&\forall i, j \in \{1, 2, \ldots, N\},\ i \neq j,\ \begin{cases} x_i + x_j \;\geq\; d(P_i, P_j) \\ x_i + d(P_i, P_j) \;\geq\; x_j \\ x_j + d(P_i, P_j) \;\geq\; x_i \end{cases} \\
&\forall i \in \{1, 2, \ldots, N\},\ x_i \geq 0
\end{aligned}$$

Given a suboptimal solution $\tilde{P}$ and the computed lower bound, the deviation $\Delta = \mathrm{SOD}(\tilde{P}) - \Gamma$ can thus give a hint of the absolute accuracy of $\tilde{P}$. In particular, if $\Delta \approx 0$, then it can be safely claimed that there is hardly room for further improvement (for the particular data set at hand).

In this paper we present a study of the lower bound Γ using two cluster ensemble methods and eleven data sets. Among others it will be demonstrated that this lower bound can (almost) be reached by the computed solution. This tightness indicates the limited room for further improvement. Therefore, the lower bound Γ represents a means of exploring the performance limit of cluster ensemble techniques.

The remainder of this paper is organized as follows. Section 3 describes the experimental settings of our study. The experimental results are presented in Section 4. Later in Section 5 the study is extended to deal with weighted cluster ensemble techniques. Finally, some further discussions conclude this paper.

3 Experimental Settings

In this section we give the details of designing our study: Metric distance functions, cluster ensemble methods, and data sets used in the experiments and the test protocol.

3.1 Metric Distance Functions

Many distance functions have been suggested to measure the dissimilarity of two partitions of the same data set; see [13] for a detailed discussion. For our study the following three were selected, which are provably metric.

Variance of information: This metric is an information-theoretic one. Given two partitions P and Q of X, it is defined by

$$d_{vi}(P,Q) \ = \ H(P) + H(Q) - 2I(P,Q)$$

where $H(P)$ and $H(Q)$ are the entropy of P and Q, respectively, and $I(P,Q)$ represents the mutual information of P and Q; see [13] for a proof of the metric property.

van Dongen metric: Fundamental to this distance function [18] is a (non-optimal) matching of the two sets of clusters.

$$d_{vd}(P,Q) \ = \ 1 - \frac{1}{2n} \cdot \Big(\sum_{C_p \in P} \max_{C_q \in Q} |C_p \cap C_q| + \sum_{C_q \in Q} \max_{C_p \in P} |C_q \cap C_p| \Big)$$

Mirkin metric: Let a equal to the number of pairs of patterns co-clustered in P but not in Q and b equal to the number of pairs of patterns co-clustered in Q but not in P. Then, the Mirkin metric belongs to the class of distance functions based on counting pairs and is simply defined by $d_m(P,Q) = a + b$. A proof of the metric property can be found in [7].

3.2 Cluster Ensemble Methods

We used two cluster ensemble methods in our experiments. The first one is the evidence accumulation method [6]. It computes the co-occurrence matrix, which is interpreted as a new similarity measure between the patterns. The consensus partition is then obtained by using a hierarchical clustering algorithm. We report the results based on the average-linkage variant (`EAC-AL`) only, since it mostly outperforms the single-linkage variant. The second cluster ensemble method (`RW`) is based on the co-occurrence matrix as well. But it adapts a random walker segmentation algorithm to produce a final clustering [1].

3.3 Data Sets

For our experiments we used two data sources. Nine UCI data sets [2] as summarized in Table 1. Special remarks need to be made about the Mammographic Mass (Mammo) and the Optical Recognition of Handwritten Digits (Optic) data sets. For Mammo, all patterns with missing values were removed, reducing this way the number of patterns from 961 to 830. For the Optic data set we extracted a subset of the first 100 patterns of each digit, producing a subset of 1000 patterns.

Two artificial data sets from [17] were included into our experiments. The first data set (2D2K) contains 500 2D points from two Gaussian clusters and the second data set (8D5K) contains 1000 points from five multivariate Gaussian distributions (200 points each) in 8D space.

Table 1. Summary of test data sets

Data set	n	# attributes	# clusters
Iris	150	4	3
Wine	178	13	3
Breast	683	9	2
Optic	1000	64	10
Soy	47	35	2
Glass	218	9	7
Haberman	306	3	2
Mammo	830	5	2
Yeast	1484	8	10
2D2K	500	2	2
8D5K	1000	8	5

3.4 Test Protocol

Given an ensemble P, we compute a final clustering $\tilde{P}$ using either `EAC-AL` or `RW`. The following measures are used to characterize the performance: $\mathrm{SOD}(\tilde{P})$, the lower bound Γ (for the ensemble), and the deviation

$$\Delta' = (\mathrm{SOD}(\tilde{P}) - \Gamma)/\mathrm{SOD}(\tilde{P})$$

(in percentage). For each data set, this procedure is repeated ten times (i.e. ten different ensembles) and the average measures are reported.

4 Experimental Results and Discussions

For the two cluster ensemble methods `EAC-AL` and `RW` the performance measures are shown in Table 2. The deviation Δ' can be interpreted as the potential of further improvement. For three data sets (Haberman, Mammo, and 2D2K) $\mathrm{SOD}(\tilde{P})$ almost reaches the lower bound Γ for all three distance functions, indicating practically no room for improvement. To some extent the same applies to the data set Soy and 8D5K in conjunction with `EAC-AL`. In these cases the lower bound turns out to be extremely tight. On the other hand, if the deviation is large, we must be careful in making any claims. The large deviation may be caused by two reasons: The lower bound is not tight enough in that particular case or the computed solution $\tilde{P}$ is still far away from the (unknown) optimal solution P^*.

The second case is certainly more delicate. But we may interpret as of some, although uncertain, potential of further improvement. Given such an ensemble, we could generate more ensembles and compute additional candidates for consensus clustering. The measure SOD can then be used for selecting a final solution. This strategy has been suggested in [17] (although in a different context): "Our objective function has the added advantage that it allows one to add a stage that selects the best consensus function without any supervisory information,

Table 2. Deviation Δ'

Evidence accumulation method `EAC-AL`

	d_{vi}			d_{vd}			d_m		
dataset	SOD($\tilde{P}$)	Γ	Δ'(%)	SOD($\tilde{P}$)	Γ	Δ'(%)	SOD($\tilde{P}$)	Γ	Δ'(%)
Iris	8.22	7.24	12.0	2.26	2.16	4.3	27621	25113	9.1
Wine	2.01	1.86	7.7	0.35	0.33	5.1	7232	6777	6.3
Breast	1.16	1.08	7.3	0.16	0.15	3.8	71244	68392	4.0
Optic	7.50	6.37	15.0	2.06	1.85	10.0	378439	315016	16.8
Soy	3.90	3.79	2.9	1.65	1.62	1.9	1616	1591	1.6
Glass	5.20	4.66	10.4	1.37	1.24	9.4	39909	33939	15.8
Haberman	7.60	7.58	0.3	2.84	2.84	0.0	233417	232994	0.2
Mammo	1.77	1.77	0.0	0.38	0.38	0.0	248649	248649	0.0
Yeast	13.94	11.40	18.3	3.85	3.34	13.4	3512666	3010184	14.3
2D2K	4.86	4.69	3.0	1.18	1.15	3.0	1037580	978050	5.7
8D5K	4.97	4.91	1.8	1.69	1.66	2.0	585462	579262	1.1

Random walker based method `RW`

	d_{vi}			d_{vd}			d_m		
dataset	SOD($\tilde{P}$)	Γ	Δ'(%)	SOD($\tilde{P}$)	Γ	Δ'(%)	SOD($\tilde{P}$)	Γ	Δ'(%)
Iris	8.40	7.24	13.8	2.28	2.16	5.2	28067	25113	10.5
Wine	2.09	1.86	10.0	0.35	0.33	4.5	7242	6777	5.8
Breast	1.49	1.08	27.7	0.20	0.15	23.9	90032	68392	24.0
Optic	11.38	6.37	44.0	3.90	1.85	50.9	749459	315016	57.7
Soy	6.19	3.79	36.9	4.08	1.62	52.0	3433	1591	49.3
Glass	7.96	4.66	41.1	2.53	1.24	45.9	69186	33940	49.3
Haberman	7.70	7.58	1.5	2.86	2.84	0.7	234484	232995	0.6
Mammo	1.77	1.77	0.0	0.38	0.38	0.0	248650	248650	0.0
Yeast	18.60	11.40	38.2	10.51	3.34	67.5	6606869	3010185	53.4
2D2K	4.69	4.69	0.0	1.15	1.15	0.0	978050	978050	0.0
8D5K	5.24	4.91	5.9	2.43	1.66	15.0	721412	579262	11.3

by simply selecting the one with the highest ANMI" (ANMI is the particular SOD used in that work). In doing so, a tight lower bound may give us a hint to continue or terminate the procedure without any knowledge of ground truth.

There is also the issue of inconsistency among different distance functions. Sometimes it happens that the deviation values for two distance functions vary, partly substantially. This observation is not really surprising. Different distance functions may not share the same view of dissimilarity, thus the quality of a consensus clustering. It is up to the user to decide which distance function is more suitable for a particular data clustering task.

Finally, we want to point out that the two cluster ensembles methods used in our study do not belong the class of median partition techniques. But even in this case the lower bound still provides useful information about the optimality of the computed consensus clustering.

Table 3. Comparison of lower bounds Γ and Γ_m

dataset	Γ	Γ_m	$(\Gamma_m - \Gamma)/\Gamma(\%)$
Iris	25113	26377	5.0
Wine	6777	6820	0.6
Breast	68392	71196	4.1
Optic	315016	335678	6.6
Soy	1591	1599	0.5
Glass	33940	34513	1.7
Haberman	232995	233273	0.1
Mammo	248650	248650	0.0
Yeast	3010185	3224160	7.1
2D2K	978050	1168728	8.4
8D5K	579262	584848	1.0

Special case d_m: The cluster ensemble problem with Merkin distance d_m has been intensively investigated [7,8]. This is mainly due to the simplicity of d_m, which allows to obtain deep insight into this particular consensus clustering problem. In particular, several suboptimal algorithms have been proposed with known approximation factor. In addition, a lower bound specific to d_m only can be defined:

$$\Gamma_m \;=\; \sum_{i<j} \min \Big(\sum_{k=1}^{N} X_{ij}^{(k)},\; N - \sum_{k=1}^{N} X_{ij}^{(k)} \Big)$$

where $X_{ij}^{(k)}$ is the Bernoulli random variable as 1 if x_i and x_j are co-clustered in partition P_k and 0 otherwise. Γ_m takes the specific properties of d_m into account, whereas Γ is based on the general properties of a metric only. Γ_m is thus better informed and expected to be tighter than Γ. In Table 3 we compare the closedness of the two lower bounds. It is remarkable that without any knowledge of d_m and using the metric properties alone, the general lower bound Γ almost reaches Γ_m.

5 Extension to Weighted Cluster Ensemble Techniques

Cluster ensembles techniques can be extended by assigning a weight w_i to each involved partition P_i, which represents the estimated relative merit of the partitions. In [19], for instance, four weights are considered: inter-cluster distance, intra-cluster distance, mean size of clusters, and difference between the cluster sizes. Then, the weighted median partition problem can be stated as:

$$P^* \;=\; \arg \min_{P \in \mathcal{P}_X} \sum_{i=1}^{N} w_i \cdot d(P, P_i)$$

Here we assume that smaller weights mean favorable partitions. The extension of the linear program lower bound Γ to deal with the weighted cluster ensemble problem is straightforward, resulting in a lower bound Γ_w.

$$\begin{aligned}&\text{minimize } w_1 \cdot x_1 + w_2 \cdot x_2 + \cdots + w_N \cdot x_N \text{ subject to}\\ &\forall i,j \in \{1,2,\ldots,N\},\ i \neq j,\ \begin{cases} x_i + x_j \ \geq \ d(P_i,P_j) \\ x_i + d(P_i,P_j) \ \geq \ x_j \\ x_j + d(P_i,P_j) \ \geq \ x_i \end{cases}\\ &\forall i \in \{1,2,\ldots,N\},\ x_i \geq 0\end{aligned}$$

Many cluster ensembles methods can be easily extended to integrate such weights. In co-occurrence based techniques such as `EAC-AL` and `RW` this can be done when computing the co-occurrence matrix. In our case we have used the inter-cluster distance as weights only.

For these weighted algorithms the performance measures are shown in Table 4. Compared to the unweighted results in Table 2 the things have not changed

Table 4. Deviation Δ' (weighted versions)

Weighted evidence accumulation method `EAC-AL`

	d_m			d_{vd}			d_{vi}		
dataset	SOD($\tilde{P}$)	Γ_w	$\Delta'(\%)$	SOD($\tilde{P}$)	Γ_w	$\Delta'(\%)$	SOD($\tilde{P}$)	Γ_w	$\Delta'(\%)$
Iris	0.78	0.68	12.1	0.21	0.12	4.5	2599	2356	9.2
Wine	0.20	0.19	7.5	0.04	0.03	5.0	723	678	6.2
Breast	0.12	0.11	7.3	0.02	0.02	3.8	7119	6834	4.0
Optic	0.75	0.64	14.7	0.21	0.19	9.7	36742	31492	13.9
Soy	0.39	0.38	2.2	0.16	0.16	1.4	160	158	1.2
Glass	0.52	0.47	10.5	0.14	0.12	9.6	3996	3423	12.5
Haberman	0.77	0.76	1.5	0.29	0.29	0.8	23754	23303	1.9
Mammo	0.17	0.17	0.0	0.04	0.04	0.0	23794	23794	0.0
Yeast	1.40	1.14	18.4	0.38	0.33	13.2	353189	299571	15.0
2D2K	0.52	0.52	0.0	0.13	0.13	0.0	107322	107834	0.0
8D5K	0.49	0.48	1.3	0.16	0.16	1.6	56825	56218	1.0

Weighted random walker based method `RW`

	d_m			d_{vd}			d_{vi}		
dataset	SOD($\tilde{P}$)	Γ_w	$\Delta'(\%)$	SOD($\tilde{P}$)	Γ_w	$\Delta'(\%)$	SOD($\tilde{P}$)	Γ_w	$\Delta'(\%)$
Iris	0.81	0.68	16.0	0.22	0.20	9.1	2753	2356	14.4
Wine	0.64	0.19	70.8	0.11	0.03	70.8	2303	677	70.6
Breast	0.22	0.11	50.6	0.03	0.02	50.5	13819	6834	50.5
Optic	1.12	0.64	43.0	0.36	0.19	46.0	55409	31492	42.2
Soy	0.52	0.38	25.5	0.39	0.16	47.6	307	157	42.3
Glass	0.85	0.47	44.7	0.30	0.13	51.9	6436	3422	42.5
Haberman	0.80	0.76	4.3	0.29	0.29	1.4	24101	23303	3.3
Mammo	0.17	0.17	0.0	0.04	0.04	0.0	23794	23794	0.0
Yeast	1.85	1.14	38.8	1.02	0.33	66.7	511552	299571	40.8
2D2K	0.52	0.52	0.9	0.13	0.13	0.9	108495	107833	0.5
8D5K	0.52	0.48	5.8	0.24	0.16	15.0	70603	56218	11.3

much. For the three data sets Haberman, Mammo, and 2D2K, SOD($\tilde{P}$) again almost reaches the lower bound Γ_w for all three distance functions, indicating practically no room for further improvement. In conjunction with `EAC-AL` the same can be said for the data set 8D5K. In these cases the lower bound turns out to be extremely tight. On the other hand, if the deviation is larger, we must be careful in making any claims. Also here we can take the deviation as a hint for continuing optimization.

6 Discussions and Conclusion

In this paper we have presented a study of the lower bound Γ using eleven data sets. It could be shown:

- In some cases this lower bound can (almost) be reached by the computed solution. This tightness implies that there exists no more room for further improvement for this particular data set (with respect to the used distance function). Larger deviation may indicate some, although uncertain, potential of improvement and thus serves as a hint for continuing optimization.
- The same observation can be made also for weighted version of cluster ensemble methods.
- The tightness of Γ can be even demonstrated in case of Merkin distance d_m by comparing with another lower bound, which is derived from the special nature of d_m.

Based on these facts we consider the lower bound Γ (and Γ_m in case of d_m) a means of exploring the performance limit of cluster ensemble techniques.

The lower bound defined in [11] presumes a metric distance function $d()$. The triangle inequality of a metric excludes cases in which $d(P, R)$ and $d(R, Q)$ are both small, but $d(P, Q)$ is very large. In practice, however, there may exist distance functions which do not satisfy the triangle inequality. The work [5] extends the concept of metrics to a relaxed triangle inequality. Instead of the strict triangle inequality, the relation:

$$d(P, R) + d(R, Q) \;\geq\; \frac{d(P, Q)}{1 + \varepsilon}$$

is required, where ε is a small nonnegative constant. This is also called quasi-metric in mathematics [9]. As long as ε is not very large, the relaxed triangle inequality still retains the human intuition of similarity. Note that the strict triangle inequality is a special case with $\varepsilon = 0$. The lower bound Γ can be easily extended to quasi-metric distance functions by changing the inequalities in the linear program accordingly. This extended lower bound can be expected to be useful in working with cluster ensemble methods based on quasi-metrics.

References

1. Abdala, D.D., Wattuya, P., Jiang, X.: Ensemble clustering via random walker consensus strategy. In: Proc. of ICPR, Istanbul (2010)
2. Asuncion, A., Newman, D.J.: UCI machine learning repository (2010)
3. Barthelemy, J.P., Leclerc, B.: The median procedure for partition. In: Partitioning Data Sets. AMS DIMACS Series in Discrete Mathematics, pp. 3–34 (1995)
4. Basu, S., Davidson, I., Wagstaff, K.L. (eds.): Constrained Clustering: Advances in Algorithms, Theory, and Applications. CRC Press, Boca Raton (2009)
5. Fagin, R., Stockmeyer, L.: Relaxing the triangle inequality in pattern matching. Int. Journal on Computer Vision 28(3), 219–231 (1998)
6. Fred, A., Jain, A.K.: Combining multiple clusterings using evidence accumulation. IEEE Trans. on PAMI 27(6), 835–850 (2005)
7. Gionis, A., Mannila, H., Tsapara, P.: Clustering Aggregation. ACM Trans. on Knowledge Discovery from Data 1(1) (2007)
8. Goder, A., Filkov, V.: Consensus clustering algorithms: Comparison and refinement. In: Proc. of Workshop on Algorithm Engineering and Experiments, pp. 109–117 (2008)
9. Heinonen, J.: Lectures on Analysis on Metric Spaces. Springer, New York (2001)
10. Jiang, X., Münger, A., Bunke, H.: On median graphs: Properties, algorithms, and applications. IEEE Trans. on PAMI 23(10), 1144–1151 (2001)
11. Jiang, X., Bunke, H.: Optimal lower bound for generalized median problems in metric space. In: Caelli, T., Amin, A., Duin, R.P.W., Kamel, M., de Ridder, D. (eds.) SPR 2002 and SSPR 2002. LNCS, vol. 2396, pp. 143–151. Springer, Heidelberg (2002)
12. Lopresti, D., Zhou, J.: Using consensus sequence voting to correct OCR errors. Computer Vision and Image Understanding 67(1), 39–47 (1997)
13. Meila, M.: Comparing clusterings - an information based distance. Journal of Multivariate Analysis 98(5), 873–895 (2007)
14. Luo, H., Jing, F., Xie, X.: Combining multiple clusterings using information theory based genetic algorithm. In: Proc. of Int. Conf. on Computational Intelligence and Security, pp. 84–89 (2006)
15. Mirkin, B.G.: Mathematical Classification and Clustering. Kluwer Academic Press, Dordrecht (1996)
16. Pelillo, M.: What is a Cluster? Perspectives from Game Theory. In: NIPS Workshop on ”Clustering: Science of Art?” (2009)
17. Strehl, A., Ghosh, J.: Cluster ensembles – a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3, 583–617 (2002)
18. van Dongen, S.: Performance criteria for graph clustering and Markov cluster experiments. Technical Report INSR0012, Centrum voorWiskunde en Informatica (2000)
19. Vega-Pons, S., Correa-Morris, J., Ruiz-Shulcloper, J.: Weighted cluster ensemble using a kernel consensus function. In: Ruiz-Shulcloper, J., Kropatsch, W. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 195–202. Springer, Heidelberg (2008)
20. Xu, R., Wunsch II, D.: Survey of clustering algorithms. IEEE Trans. on Neural Networks 16(3), 645–678 (2005)

Contour Grouping by Clustering with Multi-feature Similarity Measure

Xue Bai, Siwei Luo, Qi Zou, and Yibiao Zhao

School of Computer and Information Technology
Beijing Jiaotong University
Beijing, China
bjtuxbai@gmail.com

Abstract. Contour grouping is a key task in computer vision domain. It extracts the meaningful objects information from low-level image features and provides the input for the further processing. There have been many techniques proposed over the decades. As a useful data analysis method in machine learning, clustering is a natural way for doing the grouping task. However, due to many complicated factors in natural images, such as noises and clutter in background, many clustering algorithms, which just use pairwise similarity measure, are not robust enough and always fail to generate grouping results that are consistent with the visual objects perceived by human. In this article, we present how the grouping performance is improved by utilizing multi-feature similarity under the information based clustering framework compared with other clustering methods using pairwise similarity.

1 Introduction

As an important task in computer vision, contour grouping takes the basic image features (e.g. edgels) as input and forms the object contours for further processing. This vision task can be seen as a clustering process with some predefined similarity measure if there is no prior information about the detected objects. According to the Gestalt Laws of perceptual organization [1], the similarity between any pair of edgels can be calculated, so most methods use pairwise similarity matrix as input for clustering procedure. However, in practice, due to the limitation of pairwise similarity on capturing global data structure, the unsupervised recognition process is very sensitive to the quality of feature description and is affected significantly by the noisy features in background. In [2], the multi-feature grouping cue is introduced. It can be defined over three or more data features and is considered to be more general and reliable.

The information-based clustering (Iclust) [3] provides more flexible descriptions on data relationships by utilizing collective rather than pairwise measures of similarity. For contour grouping, we propose to use the collective similarity measure, named multi-feature similarity, as the input information for clustering process, and we compare the grouping results with the one produced by pairwise

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 415–422, 2010.

similarity. The experiments show that the grouping quality is obviously improved by using multi-feature similarity.

In the rest of this paper, the Iclust algorithm and its framework for multi-feature similarity measure is introduced in Sect. 2. In Sect. 3, we describe the contour grouping process based on Iclust with multi-feature similarity. The experimental results is presented in Sect. 4, and the last part is our conclusion.

2 Information Based Clustering

The information based clustering method formulate clustering as a tradeoff between maximizing the mean similarity of elements within a cluster and minimizing the complexity of the description provided by cluster membership. The goal of the algorithm is, for each data element i, finding a probabilistic assignment to clusters $P(C|i)$ that maximize the object function

$$\mathcal{F} = \langle s \rangle - TI(C;i) \ , \tag{1}$$

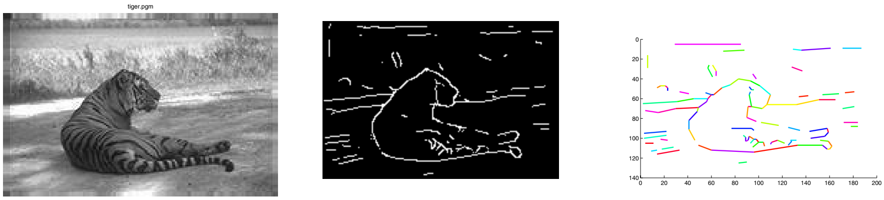
where $\langle s \rangle = \sum_{C=1}^{N_C} P(C)s(C)$ is the mean similarity of elements chosen at random out of each cluster, $I(C;i) = \frac{1}{N}\sum_{i=1}^{N}\sum_{C=1}^{N_C} P(C|i)\log\left[\frac{P(C|i)}{P(C)}\right]$ is the mutual information between the clusters variable C and elements variable i, and T is the Lagrange multiplier. Furthermore, $s(C)$ is defined as the average similarity among elements chosen out of a single cluster

$$s(C) = \sum_{i_1=1}^{N}\sum_{i_2=1}^{N}\cdots\sum_{i_r=1}^{N} P(i_1|C)P(i_2|C)\cdots P(i_r|C)s(i_1,i_2,\cdots,i_r) \ . \tag{2}$$

The similarity measure $s(i_1,i_2,\cdots,i_r)$ in above formulation is a collective measure of similarity among $r(r > 2)$ elements $i_1,i_2,\cdots,i_r$. It is very useful for contour grouping task when multi-feature grouping cues (e.g. cocircularity requires at least three data elements [2]) are involved. Thus, Iclust provides a good framework for describing more various data relations, not just limited to pairwise relation. We will demonstrate how the multi-feature similarity affects the grouping results through experiments in Sect. 4.

3 Contour Grouping

To do contour grouping, the first step is to obtain edges detected by an edge detector. Here, we use pb edge detector [4] to get the edge points and then use edgelink algorithm [5] to form small line segments (edgels). Figure 1 gives an example of edges and edgels of the same image. We now can take these edgels as basic features or elements for further grouping process.



(a) Original image (b) Binary edges (c) Edgels

Fig. 1. An example of edges and edgels generated from the same image. (a) original image; (b) edges detected by edge detector; (c) edgels formed by edgelink.

3.1 Multi-feature Grouping Cue

For each edgel, we compute the mean gray value of nearby area to represent that edgel. As an edgel reflects the change of gray value from one side to the other, we need to calculate the mean gray values of two areas on both sides of the edgel respectively. As shown in Fig. 2, the mid point of an edgel is firstly picked up, and we can obtain a certain size of square area (e.g. 6×6). So, taking the edgel as a borderline, the mean gray values on both sides of the edgel can be calculated by pixels within the area enclosed by the square and the borderline.

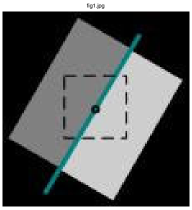


Fig. 2. Two regions beside an edgel

So far, we have two gray values for each edgel, one representing the area with small gray level and the other representing the area with higher gray level. To measure the similarity among multiple features or edgels, we first calculate the variance for each gray value among the edgels. It is inspired by the variance definition that "it is a measure of the dispersion of a sample". Then, the similarity value of $r(r > 2)$ edgels in terms of gray values is defined as following:

$$s(i_1, i_2, \cdots, i_r) = e^{-\frac{var_1^2(i_1,i_2,\cdots,i_r)}{\sigma_1^2}} \cdot e^{-\frac{var_2^2(i_1,i_2,\cdots,i_r)}{\sigma_2^2}}, \tag{3}$$

where $var_1(i_1, i_2, \cdots, i_r) = \frac{\sum_{i=1}^{r}(x_i - \overline{x})^2}{r-1}$, and $var_2(i_1, i_2, \cdots, i_r) = \frac{\sum_{i=1}^{r}(y_i - \overline{y})^2}{r-1}$. Variables x and y represent the two gray values respectively. The parameters σ_1

and σ_2 are the prior knowledge of the two variances. Here, we choose the average of all the values of var_1 for σ_1 and the average of all the values of var_2 for σ_2.

3.2 Clustering Process with Multi-feature Similarity Measure

As we mention in Sect. 2, the formulation of Iclust algorithm contains a collective measure of similarity which can describe the relation among multiple data elements. According to [3], if the derivative of object function (1) with respect to the variables $P(C|i)$ is equated to zero, we can obtain the optimal solution:

$$P(C|i) = \frac{P(C)}{Z(i;T)} \exp\left\{\frac{1}{T}[rs(C;i) - (r-1)s(C)]\right\} \quad , \tag{4}$$

where $Z(i;T)$ is a normalization constant and $s(C;i)$ is the expected similarity between i and $r-1$ elements in cluster C. Equation (4) can be turned into an iterative algorithm that finds an explicit numerical solution for $P(C|i)$ corresponding to a (perhaps local) maximum of (1). Algorithm 1 gives the detailed procedure of the clustering algorithm. We implemented the algorithm based on [6], and for the calculation of $s(C)$ and $s(C;i)$, we only consider the "pure" multi-feature similarity $s(i_1, i_2, \cdots, i_r)$ in which the r elements $i_1, i_2, \cdots, i_r$ are different from each other.

Algorithm 1. Information-based clustering algorithm with multi-feature similarity

```
Input:
    – parameter $T$ and convergence parameter $\epsilon$ (we set $T = 1/25$ and $\epsilon = 1 \times 10^{-10}$
      in our experiment)
    – number of clusters $N_C$
    – number of elements $r$ in similarity measure $s(i_1, i_2, \cdots, i_r)$
Output: "soft" partition of the $N$ elements into $N_C$ clusters.
Initialization:
    1. m=0
    2. For each element $i = 1, \cdots, N : P^{(m)}(C|i) \leftarrow$ random distribution.
While True
  For each element $i = 1, \cdots, N$ :
    1. update $P^{(m)}(C|i)$:
       $P^{(m+1)}(C|i) = \frac{P^{(m)}(C)}{Z(i;T)} \cdot \exp\left\{\frac{1}{T}[rs^{(m)}(C;i) - (r-1)s^{(m)}(C)]\right\}$
       where
         – $s(C;i) = \sum_{i_1=1}^{N} \cdots \sum_{i_{r-1}=1}^{N} P(i_1|C) \cdots P(i_{r-1}|C) s(i_1, \cdots, i_{r-1}, i)$
         – $s(C) = \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} \cdots \sum_{i_r=1}^{N} P(i_1|C)P(i_2|C) \cdots P(i_r|C) s(i_1, i_2, \cdots, i_r)$
         – $P(i|C) = P(C|i)P(i)/P(C)$, $P(C) = \sum_{i=1}^{N} P(C|i)P(i)$, $P(i) = 1/N$
         – $Z(i;T) = \sum_{C'=1}^{N_C} P(C'|i)$
    2. m=m+1
    3. if $\left|P^{(m+1)}(C|i) - P^{(m)}(C|i)\right| \leq \epsilon$, break.
```

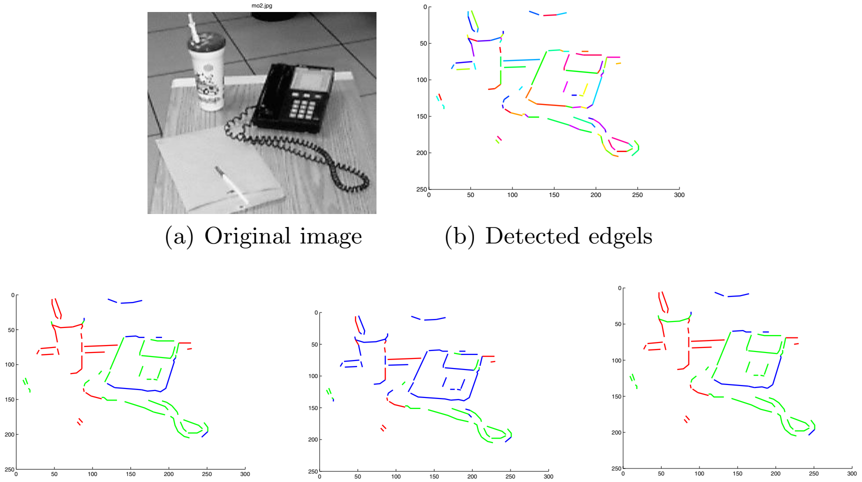
4 Experimentation and Analysis

In our experiment, we pick up gray images and only consider gray value information when calculate similarities described in Sect. 3.1. To investigate the influence of multi-feature similarity measure on grouping performance, we test two clustering algorithms using pairwise similarity. One is Ncut [7], which is considered an effective clustering method for perceptual organization of low-level image features by partitioning a graph representation [8]. The other is also based on Iclust in the case of parameter $r = 2$. We define the pairwise similarity measure based on mean gray values of areas beside an edgel as well:

$$s(i_1, i_2) = e^{-\frac{d_1^2(i_1,i_2)}{\sigma_1^2}} \cdot e^{-\frac{d_2^2(i_1,i_2)}{\sigma_2^2}} , \tag{5}$$

where $d_1(i_1, i_2) = |x_1 - x_2|$, and $d_2(i_1, i_2) = |y_1 - y_2|$. Variables x and y represent the two gray values respectively. The parameters σ_1 and σ_2 are the prior knowledge of two distance measures d_1 and d_2. Specifically, we choose the average of all the values of d_1 for σ_1 and the average of all the values of d_2 for σ_2. For grouping with multi-feature similarity measure, we test the case where $r = 3$. As mentioned in Sect. 3.2, only the "pure" 3-feature similarities $s(i_1, i_2, i_3)$ (i_1, i_2, i_3 are different edgel elements) are considered. Thus, the clustering process is totally based on multi-feature similarity.

We first test whether the clustering procedures can distinguish different object contours in the image including two salient objects. Figure 3 shows that Ncut and Iclust with 3-feature similarity have better performance than Iclust with pairwise similarity.



(a) Original image

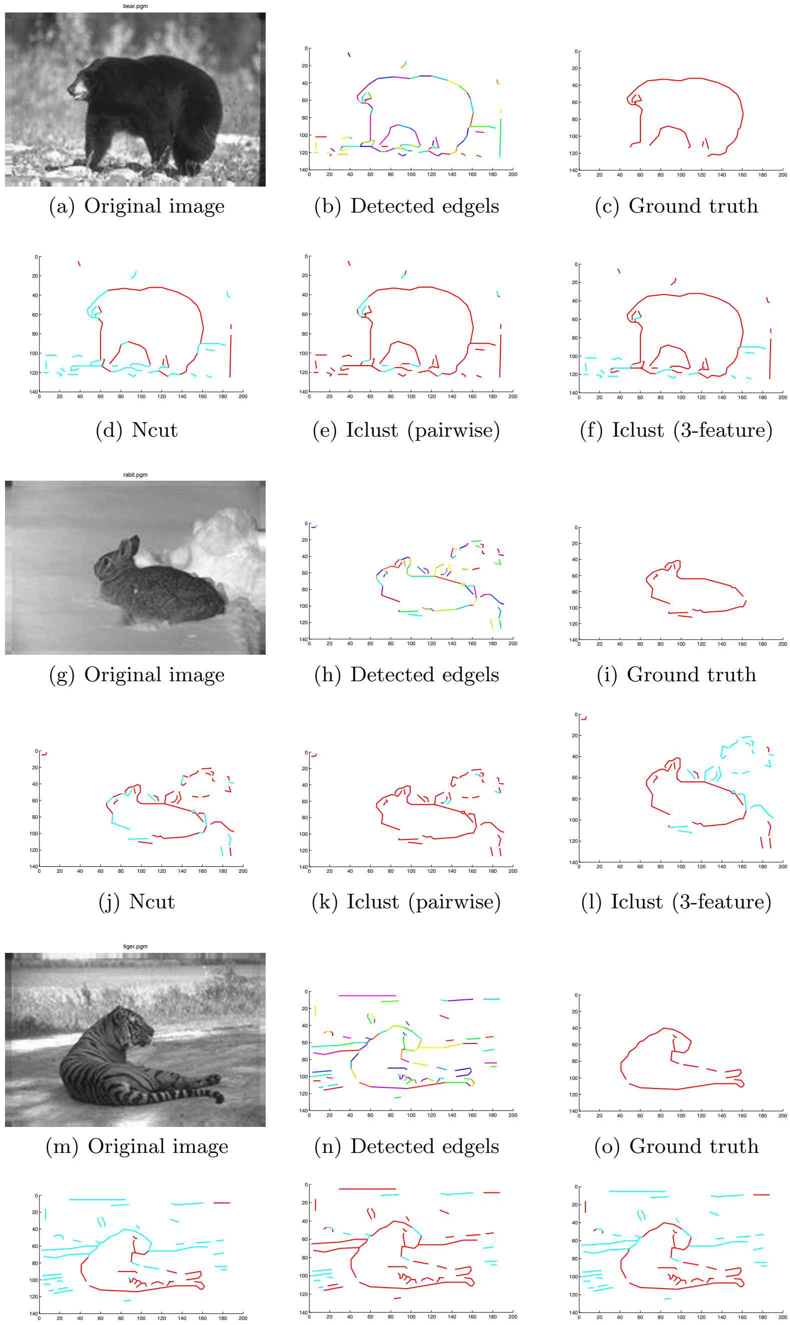
(b) Detected edgels

(c) Ncut

(d) Iclust (pairwise)

(e) Iclust (3-feature)

Fig. 3. Grouping results for the image with multiple objects



(a) Original image (b) Detected edgels (c) Ground truth

(d) Ncut (e) Iclust (pairwise) (f) Iclust (3-feature)

(g) Original image (h) Detected edgels (i) Ground truth

(j) Ncut (k) Iclust (pairwise) (l) Iclust (3-feature)

(m) Original image (n) Detected edgels (o) Ground truth

(p) Ncut (q) Iclust (pairwise) (r) Iclust (3-feature)

Fig. 4. Grouping results produced by Ncut, information-based clustering with pairwise similarity and information-based clustering with 3-feature similarity

Table 1. Grouping performance measure

Image Label	Performance Measure	Ncut	Iclust(pairwise)	Iclust(3-feature)
Fig.4 (a)	Precision	0.78	0.50	0.60
	Recall	0.54	0.85	0.89
	β	0.65	0.65	**0.73**
Fig.4 (g)	Precision	0.41	0.47	0.78
	Recall	0.70	1.00	0.95
	β	0.54	0.68	**0.86**
Fig.4 (m)	Precision	0.57	0.44	0.59
	Recall	0.64	0.92	0.85
	β	0.60	0.64	**0.71**

Figure 4 shows the grouping results for extracting one salient object contour from background. We evaluate the grouping quality by calculating precision and recall values, and the total performance is measured by $\beta = \sqrt{precision \cdot recall}$. Table 1 lists the performance measure for each clustering procedure. As we just consider gray information in calculating similarities, and a gray value is an average gray level within a certain small area, this feature description is relatively rough and could bring some inaccuracy to the measurement of similarity. We observe that 3-feature similarity measure has more stable and better grouping performance than the other two clustering procedures using pairwise similarity. It indicates that multi-feature similarity is more robust and not sensitive to the quality of feature description.

5 Conclusion

We present how the multi-feature similarity measure influence the grouping results under the information-based clustering framework for the computer vision task of contour grouping. We define this kind of similarity based on the variance of gray values over multiple edgels. Through the experiment, we find that multi-feature grouping cue is more reliable and robust compared with bi-feature cue. In the future work, more image data and various values of parameter r should be tested. And beside using variance for calculating multi-feature similarity, other grouping cues should be further investigated.

Acknowledgments

This work is supported by National High Technology Research and Development Program of China (2007AA01Z168), National Nature Science Foundation of China (60975078, 60902058, 60805041, 60872082, 60773016), Beijing Natural Science Foundation (4092033) and Doctoral Foundations of Ministry of Education of China (200800041049).

References

1. Elder, J.H., Goldberg, R.M.: Ecological Statistics for the Gestalt Laws of Perceptual Organization of Contours. Journal of Vision 2(4), 324–353 (2002)
2. Amir, A., Lindenbaum, M.: A Generic Grouping Algorithm and Its Quantitative Analysis. IEEE Trans. Pattern Analysis and Machine Intelligence 20(2), 168–185 (1998)
3. Slonim, N., Atwal, G.S., Tkačik, G., Bialek, W.: Information-based clustering. Proceedings of the National Academy of Sciences of the United States of America 102(51), 18297–18302 (2005)
4. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(5), 530–549 (2004)
5. Kovesi, P.D.: MATLAB and Octave Functions for Computer Vision and Image Processing. School of Computer Science & Software Engineering, The University of Western Australia (2000), http://www.csse.uwa.edu.au/~pk/research/matlabfns/
6. Slonim, N., Atwal, G.S., Tkačik, G., Bialek, W.: Iclust_1.0: Matlab Code for Information Based Clustering (2004), http://www.princeton.edu/~nslonim
7. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
8. Soundararajan, P., Sarkar, S.: An In-Depth Study of Graph Partitioning Measures for Perceptual Organization. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(6), 642–660 (2003)

A Psychophysical Evaluation of Texture Degradation Descriptors

Jiří Filip[1], Pavel Vácha[1], Michal Haindl[1], and Patrick R. Green[2]

[1] Institute of Information Theory and Automation of the ASCR, Czech Republic
[2] School of Life Sciences, Heriot-Watt University, Edinburgh, Scotland

Abstract. Delivering digitally a realistic appearance of materials is one of the most difficult tasks of computer vision. Accurate representation of surface texture can be obtained by means of view- and illumination-dependent textures. However, this kind of appearance representation produces massive datasets so their compression is inevitable. For optimal visual performance of compression methods, their parameters should be tuned to a specific material. We propose a set of statistical descriptors motivated by textural features, and psychophysically evaluate their performance on three subtle artificial degradations of textures appearance. We tested five types of descriptors on five different textures and combination of thirteen surface shapes and two illuminations. We found that descriptors based on a two-dimensional causal auto-regressive model, have the highest correlation with the psychophysical results, and so can be used for automatic detection of subtle changes in rendered textured surfaces in accordance with human vision.

Keywords: texture, degradation, statistical features, BTF, eye-tracking, visual psychophysics.

1 Introduction

Advanced graphics applications such as virtual interior design, cultural heritage ditization, etc. require considerable effort to render the appearance of real-world accurately. When it comes to photo-realistic appearance of materials there is no other way than to use view- and illumination-dependent measurements of real materials. Such measurements can be represented by means of *bidirectional texture functions* (BTF) [1]. Seven-dimensional BTFs represent challenging data due to theirs massive size and thus have high processing and rendering expenses. A number of approaches to BTF compression and modelling have been published in the past as shown in survey [2]. Although BTF generative statistical models exist that are capable to reach huge compression ratios themselves, they can profit from data measurement compression as well, as it can improve their learning and modelling efficiency.

The main disadvantage of most of the compression methods is that they have fixed parameters regardless of the type of sample being compressed. There have been attempts to use data on visual perception for improvement of texture data

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 423–433, 2010.

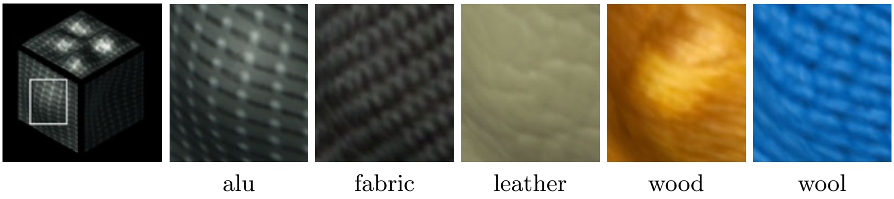
compression. Filip et al. [3] applied a psychophysical study to obtain perceptually important subset of view- and illumination-dependent images and thus consequently reduced the amount the data to be processed. On the other hand, Guthe et al. [4] used standard contrast sensitivity in cone response space together with a psychometric difference for improvement of the data compression. Interactions of human gaze fixation with different surface textures have also been analysed [5]. Although these approaches provide pioneering introductions of perceptual methods for improvement of texture compression, they are not suitable for evaluation of subtle visual compression effects.

Contribution of the paper: The main motivation of our research is to find a *computational texture descriptor having responses highly correlated with human vision*. Such a descriptor could then be used for comparison of rendered images resulting from original data and data parameterized by compression methods. Based on the responses from the descriptor the methods could iteratively adapt their parameters to automatically achieve an optimal visual performance. In this paper we test a set of descriptors motivated by standard texture features used in texture retrieval and recognition application. The descriptors we tested are based on a structure similarity, visual difference predictor, local binary patterns, Gabor features, and a causal auto-regressive wide-sense type of Markov random field model. The performance of the descriptors was evaluated by a psychophysical experiment on a group of twelve subjects.

Paper organization: The experimental data are introduced in Section 2, while the tested descriptors are explained in Section 3. Section 4 describes the experimental setup and discusses the results obtained, while Section 5 evaluates the performance of the descriptors with respect to the experimental data. Section 6 summarizes the paper.

2 Test Data Design

To test robustness of the descriptors we designed a set of testing images. Each image features a cube whose three visible faces were rendered using textured materials. We used five different samples (Fig. 1) of view and illumination-dependent textures represented by Bidirectional Texture Functions (BTF) [6] (each sample comprise 81 illumination × 81 view directions, i.e. 6561 texture images of resolution 256 × 256 pixels).



alu fabric leather wood wool

Fig. 1. Examples of five tested material samples shown on a region of one test image

The cube faces were modified in a way to feature different geometry on all three visible faces (top, left, right). To test a range of shapes that occur in the real-world we used different shapes for each cube face: **I**-wide indent, **R**-random bumps, **B**-wide bump, **F**-flat face, **H**-horizontal waves, **V**-vertical waves.

For illumination we used directional light from left and right directions parallel with the upper edge of the cubes. This configuration guaranteed the same illumination of the cubes in all stimuli and similar distribution of light across the top and left/right faces in single cubes. Not all combinations of test cube orientations were used in the experiment as this would result in too high a number of test images. We used only eleven different orientations selected in a way to allow us to compare the most interesting combinations of faces geometry. Additionally, not all orientations were illuminated from both directions as shown in Fig. 2. The figure also shows orientation number (first row) and shapes of left, right, top faces (third row). To simulate possible effects of texture compression we used three filters introducing artificial degradation to the original data modification:

A - illumination/view directions downsampling to 50%
B - spatial filtering (averaging by kernel 3×3)
C - spatial filtering (averaging by kernel 5×5).

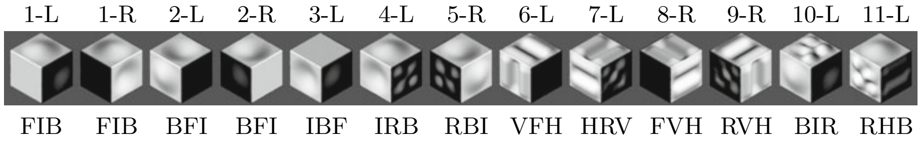


Fig. 2. Tested combinations of cube orientation and illumination direction

The proposed filters introduce only very subtle differences (Fig. 3) between the original and the modified data and force subjects to perform extensive visual search, which allows us to collect detailed gaze data. Finally, for 13 combinations of cube orientation & illuminations and 5 material samples, we obtained 65 test images for each degradation. These images were used for testing of texture descriptors proposed in the following section and also to generate stimuli in the validation experiment in Section 5.

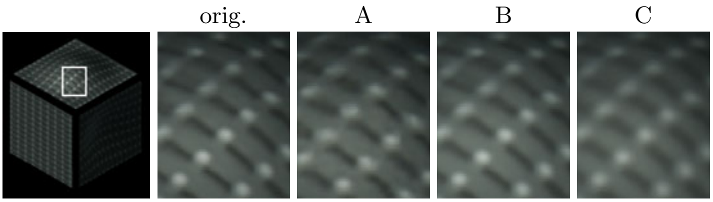


Fig. 3. Performance of the applied filters on sample *alu*

3 Texture Degradation Descriptors

The 65 images for each degradation were compared with their original counterparts. This means that we always compared images featuring the same sample, cube orientation and illumination direction. The only differences were faint degradation artifacts. Therefore, we do not require the texture descriptors to be view or illumination invariant. The descriptors tested in this paper can be principally divided into those which are translation invariant and those which are not.

3.1 Translation Non-invariant Features

These descriptors are based on perceptually motivated measures of image quality assessment measures computed in pixel-wise manner in a local neighborhood.

The first was ***visual difference predictor*** (VDP) [7], which simulates low level human perception for known viewing conditions (in our case: display size 37×30 cm, resolution 1280×1024 pixels, observer's distance 0.7 m) and thus is sufficient for our task of perceptually plausible detection of subtle texture degradation artifacts. The VDP provides percentage of pixels that differ with probability p>75% or p>95% from all pixels in the compared images. To ensure consistency with other descriptors, we set the VDP output to $(1-p)$, i.e. giving interval (0,1), where for an output 1 the images are the same.

The ***structure similarity index metric*** (SSIM) [8] is an empirical measure, which compares in power to VDP. SSIM measures the local structure similarity in a local neighborhood of an $R \times R$ window in an image (we used 11×11 pixels). The basic idea of SSIM is to separate the task of similarity measurement into comparisons of luminance, contrast, and structure. These independent components are then combined into one similarity function. The valid range of SSIM for a single pixel is $[-1, 1]$, with higher values indicating higher similarity. When the local neighborhood is evaluated for each pixel we obtain the SSIM difference of two images as a mean value of SSIM values across all pixels.

3.2 Translation Invariant Features

Markovian features are derived from the multiscale representation assuming a ***causal auto-regressive model*** (CAR) for each of the K factorisation pyramid levels. The spatial factorization is done using either the Gaussian (GP) or Gaussian-Laplacian (GLP) pyramid. Single model parameters are estimated and the texture features from all pyramid levels are concatenated into a common feature vector.

Let us assume that each multispectral texture is composed of $C = 3$ spectral planes. $Y_r = [Y_{r,1}, \ldots, Y_{r,C}]^T$ is multispectral pixel at location $r = [x, y]$. The spectral planes are either modelled by 3-dimensional (3D) CAR model or by means of a set of C 2-dimensional (2D) CAR models. The CAR representation assumes that the multispectral texture pixel Y_r can be modelled as linear combination of its neighbors:

$$Y_r = \gamma Z_r + \epsilon_r \ , \qquad Z_r = [Y_{r-s}^T : \forall s \in I_r]^T \tag{1}$$

where Z_r is the $C\eta \times 1$ data vector with multiindices r, s, t representing a causal or unilateral neighbourhood, $\gamma = [A_1, \ldots, A_\eta]$ is the $C \times C\,\eta$ unknown parameter matrix with square submatrices A_s. Some selected contextual index shift set is denoted I_r and $\eta = cardinality(I_r)$. The white noise vector ϵ_r has normal density with zero mean and unknown constant covariance matrix, same for each pixel. Given the known CAR process history, estimation of the parameters $\hat{\gamma}$ can be accomplished using fast, numerically robust and recursive statistics [9]. Five colour invariants were derived from CAR parameter estimates [10]. The texture features are these illumination invariants, which are easily evaluated during the process of estimating CAR parameters. Because the CAR models analyse a texture in some fixed movement direction, additional directions are employed to capture supplementary texture properties. The distance between two feature vectors was computed using L_1, $L_{0.2}$ norms, and by a fuzzy contrast FC_3 [11]. Although the CAR models theoretically assume texture homogenity, they can be still used as statistical descriptors of textured surfaces, and so we expect their ability to detect the degradation artifacts.

The ***Gabor features*** (GF) [12] are computed from responses of Gabor filters [13], which can be considered as orientation and scale tuneable edge and line detectors. A two dimensional Gabor function $g(r) : \mathbb{R}^2 \rightarrow \mathbb{C}$ can be specified as

$$g(r) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi i V x\right] \ , \tag{2}$$

where σ_x, σ_y, V are filter parameters. The convolution of the Gabor filter and a texture image extracts edges of given frequency and orientation range. The whole filter set was obtained by four dilatations and six rotations of the function $g(r)$, and the filter set is designed so that Fourier transformations of filters cover most of the image spectrum, see [12] for details. The Gabor features [12] are defined as the mean μ_j and the standard deviation σ_j of the magnitude of filter responses computed separately for each spectral plane and concatenated into the feature vector. These feature vectors are compared in the $L_{1\sigma}$ norm [12]. The other tested *Opponent Gabor features* (OGF) [14] are extension to colour textures, which analyses also relations between spectral channels. As our implementation involves FFT the Gabor features were computed only in the square cuts in each cube face.

The ***Local Binary Patterns*** (LBP$_{P,R}$) [15] are histograms of texture micro patterns, which are thresholded values sampled at each pixel neighbourhood. For each pixel, a circular neighbourhood around the pixel is sampled, P is the number of samples and R is the radius of the circle. Sampled points values are thresholded by a central pixel value and the pattern number is formed as follows:

$$LBP_{P,R} = \sum_{s=0}^{P-1} sgn\left(Y_s - Y_c\right) 2^s, \tag{3}$$

where *sgn* is the signum function, Y_s is a grey value of the sampled pixel, and Y_c is a grey value of the central pixel. Subsequently, the histogram of patterns is computed. Because of thresholding, the features are invariant to any monotonic grey scale change. The multiresolution analysis is done by growing the circular neighbourhood size. All LBP histograms were normalised to have a unit L_1 norm. The similarity between the histograms is computed using Kullback-Leibler divergence as authors suggested. We have tested combination of $LBP_{8,1}$ and $LBP_{8,3}$ features, and they were computed either on grey-scale images (grey) or on each spectral plane separately (RGB) and concatenated to form the feature vector.

All descriptors compute difference between sets of original images and images obtained for each degradation method, and their responses are averaged across different cube orientations, and illumination directions. It is important to note that the previous textural features are not invariant to texture deformation, which is cased by different shapes. Therefore, the features are always compared between the same surface shapes only.

4 Psychophysical Experiment

We performed a visual search experiment in order to investigate subjects' ability to identify individual introduced visual degradations. We also recorded their gaze fixations in order to analyse relations between their decisions and their fixations statistics.

Experimental Stimuli. For experimental stimuli we have used static images of size 1000×1000 pixels, featuring four cubes, described in Section 2, in individual quadrants (see Fig. 4-middle). We used this layout of stimuli to avoid the central bias in fixations reported in [16], i.e. observers have a tendency to fixate the central area of the screen. In each quadruple, three cubes showed the original data rendering and the remaining one showed a slightly modified rendering. The position of the modified cubes was random. Examples of stimuli are shown in Fig. 4. The edges of the cubes were set to black to mask potentially salient texture seams. The background and the remaining space on the screen were set to dark gray. Fig. 2 shows the 13 conditions of cube orientation and illumination direction that were used. Together with five BTF texture samples, and three different filters, the total number of stimuli was 195 (13×5×3).

Participants. Twelve paid observers (three females, nine males) participated in the experiments. All were students or university employees, were less than 35 years of age, and had normal or corrected to normal vision. All were naive with respect to the purpose and design of the experiment.

Experimental Procedure. The participants were shown the 195 stimuli in the same randomly generated order, and asked to identify which of the cubes had a surface texture slightly different from the remaining three cubes. A stimulus was shown until one of four response keys, identifying the different cube, was pressed. There was a pause of one second between stimulus presentations, and participants took on average around 90 minutes to perform the whole experiment, which was split into four sessions. All stimuli were presented on a calibrated 20.1" NEC2090UXi LCD display (60Hz, resolution 1600×1200, color

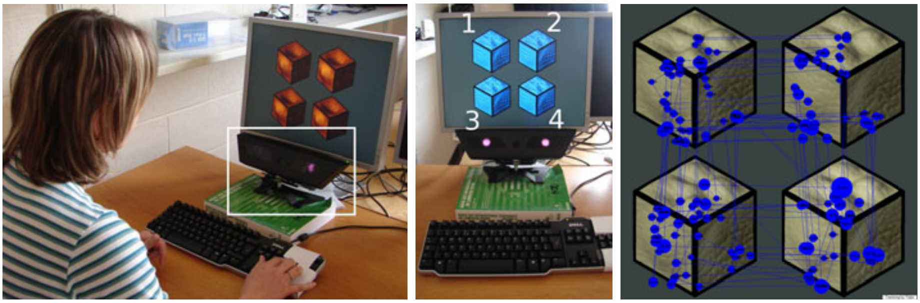


Fig. 4. Setup of the experiment with the eye-tracker highlighted, presentation of stimulus image from subject's view, and a typical gaze fixation pattern

temperature 6500K, gamma 2.2, luminance 120 cd/m^2). The experiment was performed in a dark room. Participants viewed the screen at a distance of 0.7m, so that each sphere in a pair subtended approximately 9^o of visual angle. Subjects' gaze data was recorded during the experiment using a Tobii x50 infrared-based binocular eye-tracking device as shown in Fig. 4. The device was calibrated for each subject individually and provided the locations and durations of fixations at a rate of 50 samples/s. The shortest fixation duration to be recorded was set to 100 ms.

Results – Responses accuracy. On average, the subjects were able to find the modified cube in 67% of the stimuli, which was surprisingly high in relation to the chance level 25%, given the subtle changes introduced by filters used (see Fig. 3). Informal interviews after the experiment revealed that the subjects were certain in less than 50% of stimuli and for the rest they believed that they were only guessing the right answer. The obtained rates suggest that in the difficult cases they often successfully relied on low level visual perception. The responses accuracy of individual filters is shown in Fig. 5-a and reveals that modifications introduced by the filter **A** are the hardest to spot while the smoothing by filter **C** is the most apparent; this was expected, since smoothing effect is uniform and generally more apparent that the slight illumination and view direction dependent changes in reflectance caused by reduction of directions reduction (filter **A**). While success rates across textures were quite similar for smoothing filters **B** and **C**, their values for filter **A** varied much more.

Results – Fixations. Twelve subjects performed 62 916 fixations longer than 100 ms. Average fixation duration was 242 ms. Each stimulus was on average fixated for 11 s by means of 26 fixations. Figures Fig. 5-b,c,d show subjects' gaze fixation statistics as (b) average number of fixations per stimuli, (c) average time spent fixating stimuli , and (d) average fixation time. The first two statistics are highly inversely correlated with subjects' response accuracies Fig. 5-a, with correlation coefficients $R_{(b)} = -0.904$ and $R_{(c)} = -0.930$, respectively. The figures also reveal apparent differences between the tested samples. For samples *leather* and *wood*, the subjects were less successful in identification of the modified cube,

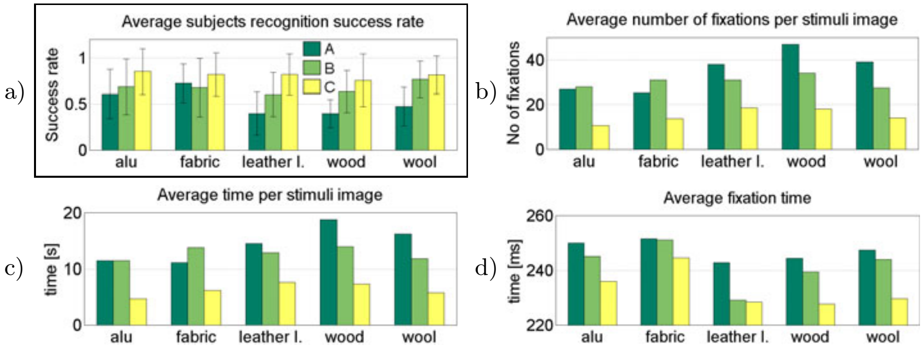


Fig. 5. Subjects' average (a) recognition success rate, (b) number of fixations per stimuli, (c) time spend on stimuli, (d) fixation duration for individual degradations and tested samples. Error-bars represent twice the standard error across subjects, different cube orientations and illuminations.

they fixated the stimuli for longer, and made significantly more fixations, which were shorter those on the other materials. We suspect that a lower local texture contrast in these samples makes detection of degradation artifacts more difficult.

5 Perceptual Evaluation and Discussion

In this section we evaluate performance of the proposed descriptors by comparison with subjects' responses obtained from the psychophysical experiment. The evaluation was based on computation of correlation coefficient $R_{X,Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$, where X, Y are compared data vectors, i.e. subjects responses and descriptor responses, and μ and σ are their means and variances.

The overall comparison of descriptors is shown in Tab. 1. From the results we observe low performance of SSIM and VDP descriptors. This can be caused 1) by their translation non-invariance, so that they give high responses even to a slight, perceptually insignificant, planar shift of texture caused by the filter **A** (this is most apparent for sample *alu* Fig. 3, and 2) by their lower sensitivity to the very subtle degradations that were tested. The Tab. 1 shows also the approximate speed of computation of differences between two textures, and the sizes of feature vectors ℓ_{FV}. We observe that although the CAR 3D has a slightly shorter feature vector than its 2D variant, it does not achieve the same performance. The table also shows for the CAR model comparison of different feature vector distances. While for 2D CAR the best performance was achieved for $L_{0.2}$ norm, for 3D CAR the best results were for L_1 norm. A high correlation with the psychophysical results was achieved by descriptors based on CAR model and LBP features. Fig. 6 shows performance of the best combination of parameters for each type of descriptor, i.e. (a) SSIM, (b) VDP (p>75%), (c) 2D CAR (GP 1), (d) 2D CAR (GLP 2), (e) Gabor (GF RGB), and (f) LBP (RGB). Generally, the best results were obtained for 2D CAR model without any pyramid (GL 1), where

Table 1. Correlation of the tested degradation descriptors with data obtained from the psychophysical experiment. The best variant of each descriptor type is highlighted.

SSIM (speed: ∼ 2 s)

R (11×11)
0.125

VDP (speed: ∼ 10 s)

R(p>75%)	R(p>95%)
0.107	0.097

CAR (speed: ∼ 4 s)

pyramid		model		R		
type	levels	dimens.	ℓ_{FV}	FC$_3$	L$_{0.2}$	L$_1$
GP	1	2D	195	0.777	$\boxed{\mathbf{0.787}}$	0.677
		3D	177	0.550	0.542	0.581
	2	2D	390	0.710	0.752	0.644
		3D	354	0.517	0.552	0.573
GLP	2	2D	390	0.654	**0.714**	0.638
		3D	354	0.360	0.362	0.573
	3	2D	585	0.648	0.677	0.620
		3D	531	0.422	0.439	0.475

LBP (speed: ∼ 1 s)

data	ℓ_{FV}	R
grey	512	0.610
RGB	1536	**0.712**

Gabor (speed: ∼ 8 s)

data	method	ℓ_{FV}	R
grey	GF	48	0.569
RGB	GF	144	**0.578**
	OGF	252	0.322

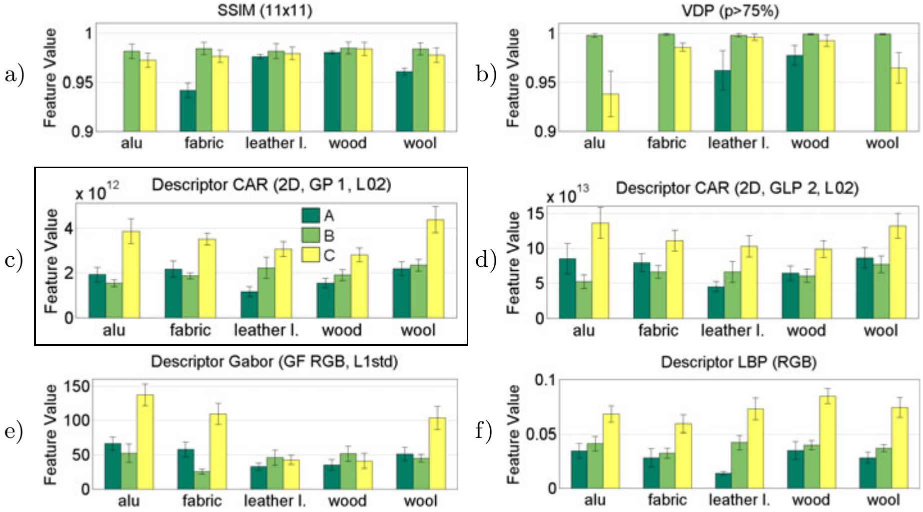


Fig. 6. Best performance of the tested descriptors (a) SSIM, (b) VDP (p>75%), (c) 2D CAR (GP 1), (d) 2D CAR (GLP 2), (e) Gabor (GF RGB), (f) LBP (RGB)

the difference of the feature vectors was evaluated using $L_{0.2}$ norm. Additionally, the CAR model enable to adjust pyramid type and size with regards to the type and intensity of degradation. Although the LBP features (f) are fast and have also quite high correlation with the human judgments, their responses clearly do not follow the trend of values across the samples present in Fig. 5-a. We tested also other variants of LBP features such as $LBP^{riu2}_{24,3+8,1}$ and $LBP^{u2}_{16,2}$, however their descriptive abilities were clearly worse than of those shown in Tab. 1.

6 Conclusions

Our results show that the statistical texture descriptors based on the causal auto-regressive model have the best performance in detection of subtle texture differences with respect to human judgments obtained in a psychophysical study. We conclude that these descriptors are the best, out of the tested features, for the automatic prediction of subtle perceptual differences in rendered view- and illumination-dependent surface textures in accordance with human perception. This highly demanded property can be used as automatic feedback for optimization the visual performance of texture compression and rendering methods.

Acknowledgements. We would like to thank Bonn University for providing the measured BTF samples, and the participants for help with the experiment. This research was supported by grants EC Marie Curie ERG 239294, GAČR 102/08/0593, and MŠMT 1M0572.

References

1. Dana, K., van Ginneken, B., Nayar, S., Koenderink, J.: Reflectance and texture of real-world surfaces. ACM Transactions on Graphics 18(1), 1–34 (1999)
2. Filip, J., Haindl, M.: Bidirectional texture function modeling: A state of the art survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(11), 1921–1940 (2009)
3. Filip, J., Chantler, M., Green, P., Haindl, M.: A psychophysically validated metric for bidirectional texture data reduction. ACM Transactions on Graphics 27(5), 138 (2008)
4. Guthe, M., Müller, G., Schneider, M., Klein, R.: BTF-CIELab: A perceptual difference measure for quality assessment and compression of BTFs. Comput. Graph. Forum 28(1), 101–113 (2009)
5. Filip, J., Chantler, M., Haindl, M.: On uniform resampling and gaze analysis of bidirectional texture functions. ACM Transactions on Applied Perception 6(3), 15 (2009)
6. Database BTF, Bonn (2003), http://btf.cs.uni-bonn.de
7. Mantiuk, R., Myszkowski, K., Seidel, H.P.: Visible difference predictor for high dynamic range images. In: IEEE International Conference on Systems, Man and Cybernetics, October 2004, vol. 3, pp. 2763–2769. IEEE, Los Alamitos (2004)
8. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004)
9. Haindl, M., Šimberová, S.: A Multispectral Image Line Reconstruction Method. In: Theory & Applications of Image Analysis, pp. 306–315. World Scientific Publishing Co., Singapore (1992)
10. Vacha, P., Haindl, M.: Image retrieval measures based on illumination invariant textural MRF features. In: CIVR, pp. 448–454. ACM, New York (2007)
11. Santini, S., Jain, R.: Similarity measures. IEEE Trans. Pattern Anal. Mach. Intell. 21(9), 871–883 (1999)
12. Ma, W.Y., Manjunath, B.S.: Texture features and learning similarity, pp. 425–430. IEEE, Los Alamitos (1996)

13. Bovik, A.: Analysis of multichannel narrow-band filters for image texture segmentation. IEEE Trans. on Signal Processing 39(9), 2025–2043 (1991)
14. Jain, A., Healey, G.: A multiscale representation including opponent colour features for texture recognition. IEEE Transactions on Image Processing 7(1), 125–128 (1998)
15. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24(7), 971–987 (2002)
16. Tatler, B.W.: The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. Journal of Vision 7(14), 1–17 (2007)

Content-Based Tile Retrieval System

Pavel Vácha and Michal Haindl*

Institute of Information Theory and Automation of the ASCR,
182 08 Prague, Czech Republic
{vacha,haindl}@utia.cas.cz

Abstract. A content-based tile retrieval system based on the underlying multispectral Markov random field representation is introduced. Single tiles are represented by our approved textural features derived from especially efficient Markovian statistics and supplemented with Local Binary Patterns (LBP) features representing occasional tile inhomogeneities. Markovian features are on top of that also invariant to illumination colour and robust to illumination direction variations, therefore an arbitrary illuminated tiles do not negatively influence the retrieval result. The presented computer-aided tile consulting system retrieves tiles from recent tile production digital catalogues, so that the retrieved tiles have as similar pattern and/or colours to a query tile as possible. The system is verified on a large commercial tile database in a psychovisual experiment.

Keywords: content based image retrieval, textural features, colour, tile classification.

1 Introduction

Ceramic tile is a decoration material, which is widely used in the construction industry. Tiled lining is relatively long-lived and labour intensive, hence a common problem to face is how to replace damaged tiles long after they are out of production. Obvious alternative to costly and laborious complete wall retiling is finding of the tile replacement from recent production which is as similar to the target tiles as possible. Tiles can differ in size, colours or patterns. We are interested in automatic retrieval of tiles as the alternative to usual slow manual browsing through digital tile catalogues and the subsequent subjective sampling. Manual browsing suffers from tiredness and lack of concentration problems, leading to errors in grading tiles. Additionally, gradual changes and changing shades due to variable light conditions are difficult to detect for humans. The presented computer-aided tile consulting system retrieves tiles from a tile digital database so that the retrieved tiles are maximally visually similar to the query tile. A user can demand either similar patterns, colours or a combination of both. Although the paper is concerned with the problem of automatic computer-aided content-based retrieval of ceramic tiles, the modification for defect detection or product quality control is straightforward.

* Corresponding author.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 434–443, 2010.

Textures are important clues to specify surface materials as well as design patterns. Thus their accurate descriptive representation can beneficial for sorting and retrieval of ceramic tiles. Without textural description the recognition is limited to different modifications of colour histograms only and it produces unacceptably poor retrieval results. Image retrieval systems (e.g.[4,13]) benefit from combination of various textural and colour features. Frequented features are colour invariant SIFT [3], Local Binary Patterns (LBP) [10], Gabor features [8], etc. A tile classifier [6] uses veins, spots, and swirls resulting from the Gabor filtering to classify marble tiles. The verification is done using manual measurement from a group of human experts. The method neglects spectral information and assumes oversimplified normalized and controlled illumination in a scanner. Similar features were used for tile defect detection [9]. A promising method for object/image recognition based on textural features was recently introduced [12].

Unfortunately, the appearance of natural materials is highly illumination and view angle dependent. As a consequence, most texture based classification or segmentation applications require multiple training images [18] captured under all available illumination and viewing conditions for each material class. Such learning is obviously clumsy and very often even impossible if the required measurements are not available. Popular illumination invariant features include LBP variants [10], however, they are very noise sensitive. This vulnerability was addressed [7], but used patterns are specifically selected according to the training set. Recently proposed LBP-HF [1] additionally studies relations between rotated patterns. Finally, the MR8 texton representation [18] was extended to be colour and illumination invariant [2].

We introduce a tile retrieval system, which takes advantage of a separate representation of colours and texture. The texture is represented by efficient colour invariant features based on Markov Random Fields (MRF), which are additionally robust to illumination direction and Gaussian noise degradation [16]. The performance is evaluated in a psychovisual experiment.

The paper is organised as follows: the tile analysis algorithm is introduced in Section 2, Section 3 describes a psychovisual evaluation and discusses its results. Section 4 summarises the paper.

2 Tile Analysis

The tile image analysis is separated into two independent parts: colour analysis and texture analysis. Advantage of this separation is ability to search for tiles with similar colours, texture, or both — according to user preference. Colours are represented by histograms, which discard any spatial relations. On the other hand, the texture analysis is based on spatial relation modelling by means of MRF type of model, which is followed by computation of colour invariants. Colour invariants are employed instead of texture analysis of grey-scale images, because colour invariants are able to distinguish among structures with same luminance.

The texture representation with MRF colour invariants was chosen, because this representation is invariant to changes of illumination colour and brightness

[15], robust to variation of illumination direction [16] and combinations of previous conditions [17]. Moreover the MRF colour invariants are robust to degradation with an additive Gaussian noise [15] and they outperformed alternative textural features such as Gabor features or LBP in texture recognition experiments [15,16,17], especially, with variations of illumination conditions. Such illumination variations are inevitable, unless all images are acquired in a strictly controlled environment.

2.1 Colour Histograms

Colour information is represented by means of cumulative histograms [14], which we compute for each spectral plane separately. The cumulative histogram is defined as the distribution function of the image histogram, the i-th bin H_i is computed as

$$H_i = \sum_{\ell \leq i} h_\ell \ , \tag{1}$$

where h_ℓ is the ℓ-th ordinary histogram bin. The distance between two cumulative histograms is computed in L_1 metric.

2.2 CAR Textural Features

The texture analysis is based on the underlying MRF type of representation, we use efficient Causal Autoregressive Random (CAR) model. The model parameters are estimated and subsequently transformed into colour invariants, which characterize the texture.

Let us assume that multispectral texture image is composed of C spectral planes (usually $C = 3$). $Y_r = [Y_{r,1}, \ldots, Y_{r,C}]^T$ is the multispectral pixel at location r, which is a multiindex $r = [r_1, r_2]$ composed of r_1 row and r_2 column index, respectively. The spectral planes are mutually decorrelated by the Karhunen-Loeve transformation (Principal Component Analysis) and subsequently modelled using a set of C 2-dimensional CAR models.

The CAR representation assumes that the multispectral texture pixel Y_r can be modelled as linear combination of its neighbours:

$$Y_r = \gamma Z_r + \epsilon_r \ , \qquad Z_r = [Y_{r-s}^T : \forall s \in I_r]^T \tag{2}$$

where Z_r is the $C\eta \times 1$ data vector with multiindices r, s, t, $\gamma = [A_1, \ldots, A_\eta]$ is the $C \times C\,\eta$ unknown parameter matrix with square submatrices A_s. In our case, C 2D CAR models are stacked into the model equation (2) and the parameter matrices A_s are therefore diagonal. Some selected contextual causal or unilateral neighbour index shift set is denoted I_r and $\eta = cardinality(I_r)$. The white noise vector ϵ_r has normal density with zero mean and unknown diagonal covariance matrix, same for each pixel.

The texture is analysed in a chosen direction, where multiindex t changes according to the movement on the image lattice. Given the known history of CAR process $Y^{(t-1)} = \{Y_{t-1}, Y_{t-2}, \ldots, Y_1, Z_t, Z_{t-1}, \ldots, Z_1\}$ the parameter estimation $\hat{\gamma}$ can be accomplished using fast and numerically robust statistics [5]:

$$\hat{\gamma}_{t-1}^T = V_{zz(t-1)}^{-1} V_{zy(t-1)} \ ,$$

$$V_{t-1} = \begin{pmatrix} \sum_{u=1}^{t-1} Y_u Y_u{}^T & \sum_{u=1}^{t-1} Y_u Z_u{}^T \\ \sum_{u=1}^{t-1} Z_u Y_u{}^T & \sum_{u=1}^{t-1} Z_u Z_u{}^T \end{pmatrix} + V_0 = \begin{pmatrix} V_{yy(t-1)} & V_{zy(t-1)}^T \\ V_{zy(t-1)} & V_{zz(t-1)} \end{pmatrix} \ ,$$

$$\lambda_{t-1} = V_{yy(t-1)} - V_{zy(t-1)}^T V_{zz(t-1)}^{-1} V_{zy(t-1)} \ ,$$

where the positive definite matrix V_0 represents prior knowledge.

Colour invariants are computed from the CAR parameter estimates to make them independent on colours. The following colour invariants were derived [15]:

1. trace: $\operatorname{tr} A_s, \ \forall s \in I_r$,
2. diagonal: $\nu_s = \operatorname{diag}(A_s), \ \forall s \in I_r$,
3. α_1: $1 + Z_r^T V_{zz}^{-1} Z_r$,
4. α_2: $\sqrt{\sum_r \left(Y_r - \hat{\gamma} Z_r\right)^T \lambda^{-1} \left(Y_r - \hat{\gamma} Z_r\right)}$,
5. α_3: $\sqrt{\sum_r \left(Y_r - \mu\right)^T \lambda^{-1} \left(Y_r - \mu\right)}$, μ is the mean value of vector Y_r ,

Feature vectors are formed from these illumination invariants, which are easily evaluated during the CAR parameters estimation process. The invariants $\alpha_1 - \alpha_3$ are computed for each spectral plane separately.

2.3 CAR-Based Tile Analysis

At the beginning, a tile image is factorised into K levels of the Gaussian-downsampled pyramid and subsequently each pyramid level is modelled by the previously described CAR model. The pyramid is used, because it enables models to captures larger spatial relations. Moreover, the CAR models analyse a texture in some fixed movement direction, therefore additional directions are employed to capture supplementary texture properties. More precisely, we used $K = 4$ levels of Gaussian-downsampled pyramid and the CAR models with the 6-th order hierarchical neighbourhood (cardinality $\eta = 14$). The texture was analysed in three orthogonal directions: row-wise, column-wise top-down and column-wise bottom-up. Finally, the estimated parameters for all pyramid levels and directions are transformed into colour invariants and concatenated into a common feature vector.

The dissimilarity between two feature vectors of two tiles T, S is computed using fuzzy contrast [11] in its symmetrical form FC_3:

$$FC_p\left(T,S\right) = M - \left\{ \sum_{i=1}^{M} \min\left\{ \tau(f_i^{(T)}), \tau(f_i^{(S)}) \right\} - p \sum_{i=1}^{M} \left| \tau(f_i^{(T)}) - \tau(f_i^{(S)}) \right| \right\} \ ,$$

$$\tau(f_i) = \left(1 + \exp\left(- \frac{f_i - \mu(f_i)}{\sigma(f_i)} \right) \right)^{-1} \ ,$$

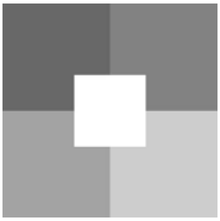


Fig. 1. Partition of tile image into five regions. The texture is analysed in the whole image and separately in these regions.

where M is the feature vector size and $\mu(f_i)$ and $\sigma(f_i)$ are average and standard deviation of the feature f_i computed over all database, respectively. The sigmoid function τ models the truth value of fuzzy predicate.

The textural representation is based on the homogeneity assumption, which is an inherent property of all textures. Unfortunately, some tiles contain insets or other violations of the homogeneity assumption. Therefore the CAR models are additionally estimated on each of five tile regions depicted in Fig. 1. The dissimilarities of corresponding image regions and whole images are combined to finally produce the dissimilarity of tiles $D(T,S)$:

$$D(T,S) = \mathrm{Norm}\left(\sum_{\ell=1}^{5} FC_3\left(T_\ell, S_\ell\right)\right) + \mathrm{Norm}\left(FC_3\left(T,S\right)\right) \ , \tag{3}$$

$$\mathrm{Norm}(FC_3\left(T,S\right)) = \frac{FC_3\left(T,S\right) - \mu(FC_3)}{\sigma(FC_3)} \ , \tag{4}$$

where T_ℓ, S_ℓ are the ℓ-th regions of images T, S, respectively. *Norm* is dissimilarity normalisation, where $\mu(FC_3)$ and $\sigma(FC_3)$ are mean and standard deviation of distances of all images. In practice, $\mu(FC_3)$ and $\sigma(FC_3)$ could be estimated on a subset of dataset, since the precise estimation is not necessary. This textural tile representation is denoted as “2D CAR 3x” in the results.

2.4 Local Binary Patterns

Local Binary Patterns (LBP) [10] are histograms of texture micro patterns. For each pixel, a circular neighbourhood around the pixel is sampled, P is the number of samples and R is the radius of circle. The sampled points values are thresholded by the central pixel value and the pattern number is formed:

$$LBP_{P,R} = \sum_{s=0}^{P-1} \mathrm{sgn}\left(Y_s - Y_c\right) 2^s , \tag{5}$$

where *sgn* is the sign function, Y_s is the grey value of the sampled pixel, and Y_c is the grey value of the central pixel. Subsequently, the histogram of patterns is computed. Because of thresholding, the features are invariant to any monotonic grey-scale change. The multiresolution analysis is done by growing of the circular neighbourhood size. All LBP histograms were normalised to have unit L_1 norm. The similarity between LBP feature vectors is measured by means of Kullback-Leibler divergence as the authors suggested. We have tested features $\mathrm{LBP}_{8,1+8,3}$, which are combination of features with radii 1 and 3 and which were computed on grey-scale images.

3 Experiments

Performance of two alternative textural retrieval methods (CAR, LBP) was evaluated in a psychovisual experiment, where the quality of retrieved images was evaluated by volunteers. The experiment was conducted on the dataset of 3301 tile images downloaded from an internet tile shop.[1] All images were resampled to the common size 300 × 400 pixels, the aspect ratio of rectangular images were maintained and the bigger side was resized to match the size. Thirty-four volunteers (26 males, 8 females) participated in our test. Age of participants ranged from nineteen to sixty, but majority was below forty. About one half of participants were specialist in the field of image processing. The test was administered over the Internet using a web application, so that each participant used its own computer in their environment. This setup is plausible, because we focused on significant, first glance differences, which are unlikely to be influenced by test conditions.

The test was composed of subsequent steps, where each step consisted of a query image and four test images. These four test images composed of two images retrieved by CAR method and two retrieved by LBP as the most similar to the query image, they were presented in a random order. Participants were instructed to evaluate quality of the retrieved images according to structural/textural similarity with the query image, regardless of colours. There were four ranks available: similar = 3, quite similar = 2, little similar = 1, dissimilar = 0. Subjects were also instructed that they should spend no more than one or two seconds per one test image. Because our system is intended to be a real-life application, we did not provide any examples of similar or dissimilar images, but we let people to judge the similarity in their own subjective opinion. The query images were once randomly selected and remained same for all participant in one run. They were presented in a fixed order so that the results were not influenced by different knowledge of previous images. Moreover, the first three query images were selected manually and were not counted in the results. The reason was to allow subjects to adjust and stabilise their evaluation scale.

The test was performed in two runs, where a single run consisted of the the same query and test images evaluated with different subjects. The first run consisted of 66 valid steps evaluated with 23 subjects, while the second one contained 67 valid steps ranked by 11 subjects. The evaluation of one subject was removed due to significant inconsistency with the others (correlation coefficient

[1] http://sanita.cz

Table 1. Subject evaluated quality of texture retrieval methods. The table contains average ranks (0 = dissimilar – 3 = similar) and corresponding standard deviations.

	2D CAR 3x	LBP$_{8,1+8,3}$
run 1	2.21 ± 0.64	2.22 ± 0.65
run 2	2.23 ± 0.62	2.21 ± 0.57

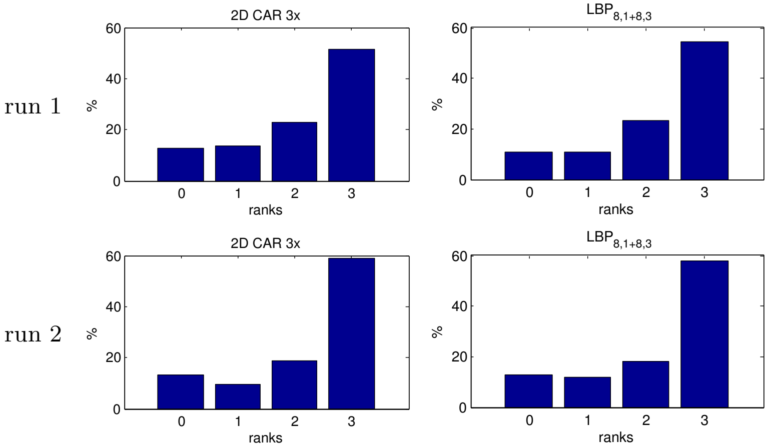


Fig. 2. Histogram of ranks (0 = dissimilar – 3 = similar) given by subjects. The first row shows histograms for the first test run, while the second row for the second run.

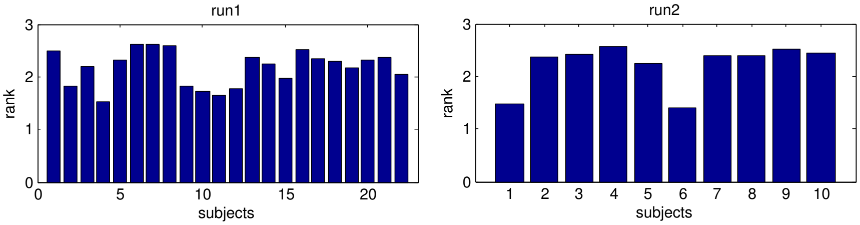


Fig. 3. Distribution of average ranks given by participants in the first and the second test run

= 0.4). Average correlation coefficients of subject evaluation were 0.64 and 0.73 for the first and the second run, respectively, which implies certain consistency in subject similarity judgements.

3.1 Discussion

The experimental results are presented in Tab. 1, which shows average ranks and standard deviations of retrieved images for CAR and LBP methods. The distribution of given ranks is displayed in Fig. 2. It can be seen that the performance

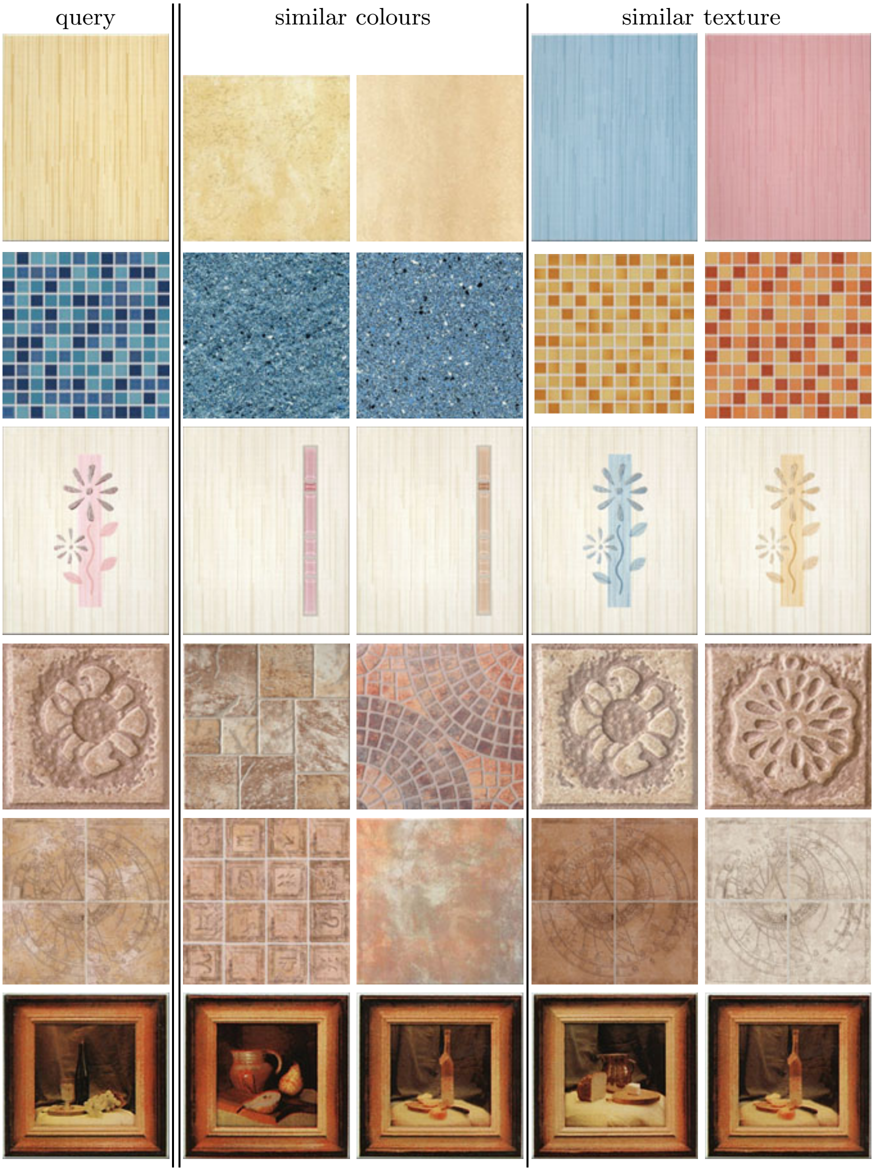


Fig. 4. Examples of similar tile retrieved by our system, which is available online at `http://cbir.utia.cas.cz/tiles/`. Query image, on the left, is followed by two images with similar colours and texture (CAR features). Images are from the internet tile shop `http://sanita.cz`

of both methods is comparable and successful. About 76% of retrieved images were considered to be similar or quite similar and only 12% were marked as dissimilar. More than two thirds of the participants ranked the retrieved tiles as quite similar or better in average, as can be seen in Fig. 3, which shows average ranks of participants. Different subject means in Fig. 3 show that the level of perceived similarity is subjective and a personal adaptation would be beneficial. Unfortunately, such an adaptation is not always possible since it requires user feedback.

As expected, the experiment revealed that LBP and CAR methods prefer different aspects of structural similarity. The LBP method is better with regular images that contain several distinct orientations of edges, while the CAR model excels in modelling of stochastic patterns. Moreover, LBP describes any texture irregularities in contrast to CAR model, which enforces homogeneity and small irregularities are ignored as errors or noise. Both approaches are plausible and it depends on a subjective view, which approach should be preferred. Moreover, according to previous experiments, the CAR features are more robust to changes of illumination direction [16] and noise degradation [15].

Based on these experiments, we decided to benefit from both these textural representations and include them into our retrieval system. The final retrieval result is consequently composed of images with colour similarity, texture similarity according to CAR, and texture according to LBP.

4 Conclusions

We designed and implemented a tile retrieval system based on two orthogonal components of visual similarity: colours and texture. The performance of the textural component was successfully evaluated in a psychovisual experiment. Example results from our interactive demonstration are shown in Fig. 4.

Our retrieval system is not limited to tile images, it can be used with other kinds of images, where the structure is important property, e.g. textiles/cloths and wallpapers.

Acknowledgements. This research was supported by grants GAČR 102/08/0593 and partially by the MŠMT grants 1M0572 DAR, 2C06019.

References

1. Ahonen, T., Matas, J., He, C., Pietikäinen, M.: Rotation invariant image description with local binary pattern histogram fourier features. In: Salberg, A.B., Hardeberg, J.Y., Jenssen, R. (eds.) SCIA 2009. LNCS, vol. 5575, pp. 61–70. Springer, Heidelberg (2009)
2. Burghouts, G.J., Geusebroek, J.M.: Material-specific adaptation of color invariant features. Pattern Recognition Letters 30, 306–313 (2009)
3. Burghouts, G.J., Geusebroek, J.M.: Performance evaluation of local colour invariants. Comput. Vision and Image Understanding 113(1), 48–62 (2009)

4. Chen, Y., Wang, J.Z., Krovetz, R.: Clue: Cluster-based retrieval of images by unsupervised learning. IEEE Trans. Image Process. 14(8), 1187–1201 (2005)
5. Haindl, M., Šimberová, S.: A Multispectral Image Line Reconstruction Method. In: Theory & Applications of Image Analysis, pp. 306–315. World Scientific Publishing Co., Singapore (1992)
6. Li, W., Wang, C., Wang, Q., Chen, G.: A generic system for the classification of marble tiles using gabor filters. In: ISCIS 2008, pp. 1–6 (2008)
7. Liao, S., Law, M.W.K., Chung, A.C.S.: Dominant local binary patterns for texture classification. IEEE Trans. Image Process. 18(5), 1107–1118 (2009)
8. Ma, W.Y., Manjunath, B.S.: Texture features and learning similarity, pp. 425–430. IEEE, Los Alamitos (1996)
9. Monadjemi, A.: Towards efficient texture classification and abnormality detection. Ph.D. thesis, University of Bristol (2004)
10. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24(7), 971–987 (2002)
11. Santini, S., Jain, R.: Similarity measures. IEEE Trans. Pattern Anal. Mach. Intell. 21(9), 871–883 (1999)
12. Shotton, J.D.J., Winn, J., Rother, C., Criminisi, A.: Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. Int. J. Comput. Vision 81(1), 2–23 (2009)
13. Snoek, C.G.M., van de Sande, K.E.A., de Rooij, O., Huurnink, B., van Gemert, J., Uijlings, J.R.R., He, J., Li, X., Everts, I., Nedovic, V., van Liempt, M., van Balen, R., de Rijke, M., Geusebroek, J.M., Gevers, T., Worring, M., Smeulders, A.W.M., Koelma, D., Yan, F., Tahir, M.A., Mikolajczyk, K., Kittler, J.: The mediamill TRECVID 2008 semantic video search engine. In: Over, P., Awad, G., Rose, R.T., Fiscus, J.G., Kraaij, W., Smeaton, A.F. (eds.) TRECVID. National Institute of Standards and Technology, NIST (2008)
14. Stricker, M., Orengo, M.: Similarity of color images. In: Storage and retrieval for Image and Video Databases III, Ferbruary 1995. SPIE Proceeding Series, vol. 2420, pp. 381–392. SPIE, Bellingham (1995)
15. Vacha, P., Haindl, M.: Image retrieval measures based on illumination invariant textural MRF features. In: Sebe, N., Worring, M. (eds.) CIVR, pp. 448–454. ACM, New York (2007)
16. Vacha, P., Haindl, M.: Illumination invariants based on markov random fields. In: Proc. of the 19th International Conference on Pattern Recognition (2008)
17. Vacha, P., Haindl, M.: Natural material recognition with illumination invariant textural features. In: Proc. of the 20th International Conference on Pattern Recognition (2010) (accepted)
18. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. Int. J. Comput. Vision 62(1-2), 61–81 (2005)

Performance Improvement in Multiple-Model Speech Recognizer under Noisy Environments

Jang-Hyuk Yoon and Yong-Joo Chung

Department of Electronics, Keimyung University
Daegu, S. Korea

Abstract. Multiple-model speech recognizer has been shown to be quite successful in noisy speech recognition. However, its performance has usually been tested using the general speech front-ends which do not incorporate any noise adaptive algorithms. For the accurate evaluation of the effectiveness of the multiple-model frame in noisy speech recognition, we used the state-of-the-art front-ends and compared its performance with the well-known multi-style training method. In addition, we improved the multiple-model speech recognizer by employing N-best reference HMMs for interpolation and using multiple SNR levels for training each of the reference HMM.

Keywords: speech recognition, multiple-model frame, noise robustness, MTR, DSR, Aurora database.

1 Introduction

Various research efforts have been done for the noise-robust speech recognition like speech feature extraction, speech enhancement and model parameter compensation [1][2][3]. These approaches are used independently or combined with each other to improve the performance of the speech recognizer under noisy environments.

As a different approach to those conventional methods, the multiple-model based speech recognizer has been proposed recently and shown quite successful results [4]. In the method, multiple acoustic models corresponding to various noise types and SNR levels are obtained during the training and the trained acoustic models are used altogether in the testing. This approach is contrary to the conventional methods where a single acoustic model corresponding to clean speech is used.

The real situation where the speech recognizer operates include various noisy environments and the distributed speech recognition (DSR) is thought to be one of the most representative noisy conditions. European Telecommunications Standards Institute (ETSI) has developed two standards for the DSR front-ends. The first standard is called FE. It a basic version and specifies a feature extraction scheme based on the widely used mel frequency cepstral coefficients (MFCC) [5]. As the FE standard did not show successful results in noisy environments, the ETSI has proposed the second standard called AFE which include some noise adaptive algorithms [6].

In the previous research [4], the multiple-model based speech recognizer has shown superior performance compared with the popular the MTR (Multi-style TRaining)

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 444–452, 2010.

approach. However, the evaluation was done using the FE front-end instead of the more noise-robust front end, AFE. In this paper, we will evaluate the effectiveness of the multiple-model framework using the AFE front-end and compare its performance with the MTR method. We also propose methods to improve the performance of the multiple-model based speech recognizer. In the previous work, only one acoustic model which is most similar to the input noisy speech is selected for recognition but there are always some errors in this process due to the inaccurate SNR estimation and even the most similar acoustic model will not exactly match to the input noisy speech due to the noise signal variability. To overcome this problem with the multiple-model based recognizer, we propose to select N most similar acoustic models and use them all together in recognition. Also, the SNR range for each acoustic model is extended to generate more robust acoustic models during training.

2 Multiple-Model Based Speech Recognizer

2.1 Improved Multiple-Model Based Speech Recognizer

In the multiple-model based speech recognizer, multiple reference HMMs are trained using noisy speech corresponding to various noise types and SNR levels and one reference HMM which is most similar to the testing noisy speech is chosen as the acoustic model for recognition. This approach is advantageous over the conventional method using a single reference HMM because it can improve robustness against various noise characteristics.

In this paper, we modified the structure of the multiple-model based speech recognizer and its architecture is shown in Fig.1. First, the noise signal extracted from the testing noisy speech is used to measure the similarity of the testing noisy speech to the reference HMMs and the most similar N reference HMMs are selected and they are interpolated for improved recognition performance. The interpolation can compensate for the errors in the selection process and the robustness of the recognizer is generally improved by using multiple acoustic models. When the probability density functions (PDFs) of the N most similar reference HMMs are given by $f_i(O), i=1,\cdots,N$, the interpolated PDF $f_{iter}(O)$ is defined as follows.

$$f_{iter}(O)=\sum_{i=1}^{N}\alpha_i f_i(O) \tag{1}$$

where O is the observation and α_i , $i=1,\cdots,N$ are the interpolation weights.

In this paper, $\alpha_i=\frac{1}{N}$, $i=1,\cdots,N$, are used to equally weight all the PDFs of the N reference HMMs. We experimented with assigning a distinct weight to each reference HMM but no significant performance improvement was observed.

Single mode Gaussian models (SGMs) are estimated for each noise type and SNR level during the training. The estimated SGMs are used in selecting the N most similar reference HMMs. The SGM for the D-dimensional noise vector **n** with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ is given as follows.

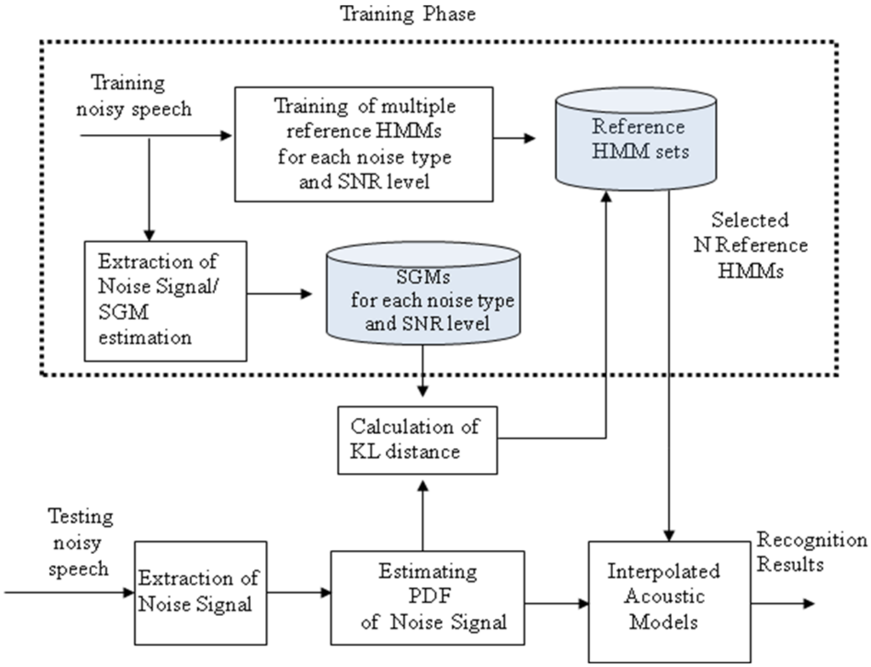

Training Phase
Training noisy speech
Training of multiple reference HMMs for each noise type and SNR level
Reference HMM sets
Extraction of Noise Signal/ SGM estimation
SGMs for each noise type and SNR level
Selected N Reference HMMs
Calculation of KL distance
Testing noisy speech
Extraction of Noise Signal
Estimating PDF of Noise Signal
Interpolated Acoustic Models
Recognition Results

Fig. 1. The architecture of the modified multiple-model based speech recognizer

$$p(\mathbf{n}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\mathbf{\Sigma}|^{D}} \exp\left\{ -\frac{1}{2} (\mathbf{n}-\boldsymbol{\mu})' \mathbf{\Sigma}^{-1} (\mathbf{n}-\boldsymbol{\mu}) \right\} \tag{2}$$

Given the noise vectors, we can estimate the mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$ by the expectation-maximization (EM) algorithm.

In recognition, the Kullback-Leibler (KL) distances between the Gaussian PDF of the testing noise signal and the SGMs are calculated and those N SGMs with the smallest KL distances are determined and their corresponding N reference HMMs are chosen as the acoustic models for recognition in the multiple-model based speech recognizer.

The KL distance (KLD) between two Gaussian PDFs $N_1(\boldsymbol{\mu}_1, \mathbf{\Sigma}_1)$, $N_2(\boldsymbol{\mu}_2, \mathbf{\Sigma}_2)$ is defined as follows [8].

$$KLD = \frac{1}{2} \sum_{i=1}^{D} \left[\log\left(\frac{\Sigma_{1,ii}}{\Sigma_{2,ii}} \right) + \frac{(\mu_{2,i} - \mu_{1,i})^2}{\Sigma_{1,ii}} + \left(\frac{\Sigma_{2,ii}}{\Sigma_{1,ii}} - 1 \right) \right] \tag{3}$$

where $\Sigma_{1,ii}$ and $\Sigma_{2,ii}$ are the i-th diagonal components of the covariance matrices and $\mu_{1,i}$ and $\mu_{2,i}$ are the i-th components of the mean vectors.

As a second approach for the performance improvement of the multiple-model based speech recognizer, we used multiple SNR levels for training each of the reference HMM. Although a single SNR level is usually assigned to each reference HMM for more discriminative acoustic models, but we improved robustness against the selection errors and noise variability by employing multiple SNR levels in the training.

2.2 Standards for the DSR Front-Ends

ETSI proposed two standard front-ends for the DSR speech recognition. The first standard ES 201 108 which was published in 2000 consists of two separate parts, feature extraction and encoding [5]. The widely used MFCC is generated in the feature extraction part while channel encoding for transmission is done in the encoding part. In this paper, we implemented only the feature extraction part as our concern is on the noise robustness of the front-ends. We call the first standard as FE and its block diagram is shown in Fig. 2.

The feature extraction part includes the compensation of the constant level offset, the pre-emphasis of high frequency components, the calculation of the spectrum magnitude, the bank of mel-scale filters, the logarithmic transform and finally the calculation of the discrete cosine transform. For every frame, a 14 dimensional feature vector consisting of 13 cepstral coefficients and a log energy is generated.

The FE front-end is known to perform inadequately under noisy conditions. Thus, a noise robust version of the front-end was proposed in 2002 [6]. This version called Advanced Front-End (AFE) is known to provide a 53(%) reduction in error rates on the connected digits recognition task compared to the FE standard [7].

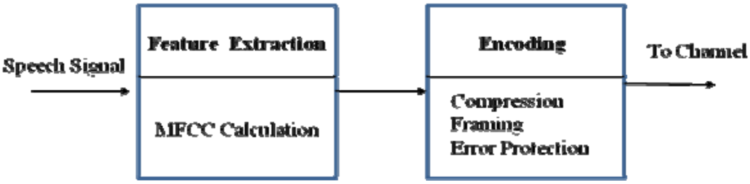


Fig. 2. Block diagram of the FE

Fig. 3 shows a block diagram of the AFE front-end. Wiener filter based noise reduction, voice activity detection (VAD), waveform processing improving the overall SNR and blind equalization for compensating the convolutional distortion are added in order to improve the recognition rates.

The multiple-model based speech recognizer has shown improved results compared with the previous noise-robust methods like the MTR when they use the FE. However, for the accurate comparison, it is necessary to compare the recognition rates when they use the AFE as the basic front-end because the AFE generally performs better than the FE in noisy conditions. Thus, in this paper, we evaluated the performance of the multiple-model speech recognizer using the AFE and proposed some methods to improve the recognition rates of the multiple-model based speech recognizer.

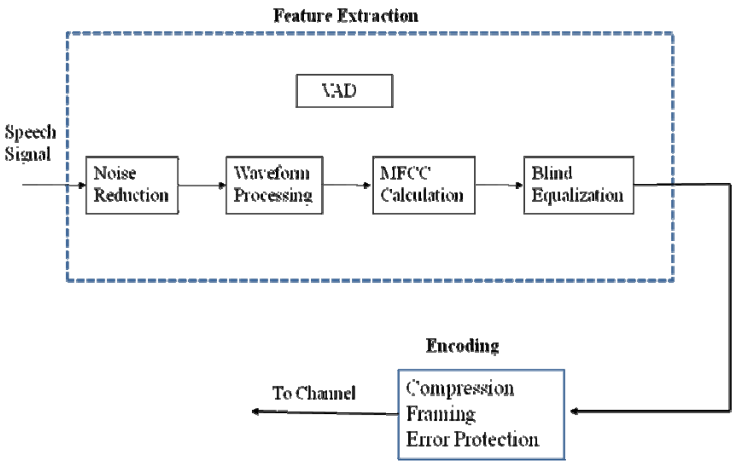


Fig. 3. Block diagram of the AFE

3 Experiments and Results

3.1 Databases and Recognition System

We used the Aurora 2 database for the experiments. There are two kinds of training approaches for the Aurora 2 database. The first one called CLEAN uses only clean speech not contaminated with any kinds of noises to train the HMM models. The second training method called MTR uses both clean and noisy speech signals which are contaminated by various kinds (subway, car, exhibition, babble) of noises at several SNR levels. The recognition experiments were conducted for Set A (including 4 known types of additive noise: subway, car, exhibition, babble), set B (including 4 unknown types of additive noise: restaurant, street, airport, train) and set C (including one known and one unknown type of noises with convolutional noises).

The AFE was used for the feature extraction. 13-th order feature vectors which consist of 12-th order MFCCs without 0-th cepstral component and the log energy are used as the basic feature vectors and their delta and acceleration coefficients are added to construct a 39-dimensional feature vector for each frame.

The HMM for each digit consists of 16 states with 3 Gaussian mixtures for each state. Silence is also modeled by a 3 state HMM with 6 Gaussian mixtures in each state. The approximate Baum-Welch algorithm was used to obtain the acoustic models.

3.2 Results

To compare the performance of the FE and AFE in noisy speech recognition, we show the word error rates (WERs) when the acoustic models are trained by CLEAN and MTR method.

Table 1. Performance comparison between the AFE and FE (WER(%))

Front-end Training		FE	AFE
CLEAN	Set A	37.43	13.67
	Set B	42.94	14.58
	Set C	33.08	15.36
	Average	**38.78**	**14.37**
MTR	Set A	12.55	8.51
	Set B	13.71	8.94
	Set C	17.03	9.83
	Average	**13.91**	**8.95**

As we can see in Table 1, the average word error rate (WER) with the FE was 38.78(%) in CLEAN training mode while the WER with the AFE was 14.37(%), which means that the AFE reduces the WER by 63(%) in CLEAN training mode. For the case of MTR training, we can also see that the AFE reduces the WER by about 35(%) compared with the FE. From these results, we can conclude that the AFE performs much better both in the CLEAN and MTR training mode on the Aurora 2 database. This also means that the previous research which demonstrated the superiority of the multiple-model based recognizer using the FE should be re-evaluated using the AFE.

In Table 2, we show the WERs of the multiple-model based recognizer using the AFE as the number of interpolated PDFs in (1) varies. The conventional multiple-model based recognizer corresponds to the case of N=1. As we increase the number of interpolated PDFs, some performance improvement is observed. We could obtain the best performance when N=4 with the WER of 10.71(%) reducing the WER of the conventional method by about 3(%).The decrease in the WER mainly comes from Set C where a 10(%) error rate reduction is achieved. The improvement may have come from reducing the negative effect of errors in finding the most similar reference HMM using the KL distance. Also, the variability of the noise signal in the testing noisy speech may have been more efficiently compensated by using multiple acoustic models in recognition.

In addition to the interpolation approach, we also tried to improve the performance of the multiple-model based speech recognizer by using multiple SNR levels for training each of the reference HMM. In Table 3, we show the two cases of merging SNR levels called SNRMERG, SNRMER2.

Table 2. The performance of the multiple-model based recognizer using the AFE (WER(%))

The number of Interpolated HMMs	Set A	Set B	Set C	Average
N=1	9.28	13.24	9.95	11.00
N=2	9.16	13.21	9.49	10.85
N=4	9.17	13.15	8.92	10.71
N=6	9.18	13.32	8.8	10.76

Table 3. The SNR levels for each noise type and the resulting number of reference HMMs for each noise type

	Conventional Method	SNRMERG	SNRMERG2
SNR Levels (dB)	{0},{5},{10}, {15},{20},{25}, {30}	{0,5},{10,15}, {20,25},{30}	{0,5},{5,10}, {10,15},{15,20}, {20,25},{25,30}, {30}
Number of reference HMMs	7	4	7

Table 4. Performance comparison of the SNRMERG and SNRMER2 method (WER(%))

	Number of HMMs	Set A	Set B	Set C	Average
SNRMERG	N=1	9.01	13.12	9.75	10.80
	N=2	8.60	13.04	9.02	10.46
	N=4	8.94	13.01	8.49	10.48
	N=6	9.17	13.07	8.49	10.59
SNRMERG2	N=1	8.8	12.72	9.66	10.54
	N=2	8.63	13.02	9.38	10.54
	N=4	8.70	13.17	9.10	10.57
	N=6	8.93	13.28	8.66	10.62

In the conventional method, the reference HMM was constructed for each SNR level (0, 5, 10, 15, 20, 25, 30 dB) independently while the SNRMERG method merged 0 and 5, 10 and 15, 20 and 25 to construct the reference HMMs reducing the number of reference HMMs for each noise type from 7 to 4. While SNRMERG2 method is similar to SNRMERG, it allows overlap in SNR levels among different reference HMMs.

In Table 4, we compared the performance of the proposed SNRMERG and SNRMERG2 method.

As we can see in Table 4, the overall recognition rates of the SNRMERG and SNRMERG2 are better than the conventional method. In Table2, the conventional method had the WER of 11.0(%) when N=1 while the SNRMERG and SNMERG2 had the WERs of 10.80(%) and 10.54(%) respectively. Also, the recognition rates of the SNRMERG and SNRMERG2 improve by increasing the number of interpolated HMMs as we have seen in Table 2. Although the difference in lowest WERs between the SNRMERG and SNRMERG2 is small, the SNRMERG2 has a merit that it does not need the interpolation to obtain the lowest WER.

We compared the improved multiple-model based speech recognizer with the MTR method which is a very popular approach in noisy speech recognition and the comparison results are shown in Table 5.

In Table 5, SNRMERG(N=4) and SNRMERG2(N=1) showed lower WERs than the conventional multiple-model based speech recognizer but they were worse than the MTR. This is contrary to the previous research result where the multiple-model based recognizer outperformed the MTR when the FE was used as the basic front-end [4]. The noise reduction algorithm in the AFE may have diminished the relative merit of noise robustness of the multiple-model based speech recognizer.

To increase the recognition rates of the proposed multiple-model based recognizer, we interpolated the PDF of the SNRMER2(N=1) with that of the MTR to take the advantage of the MTR. Although the average recognition rate of the interpolated acoustic model still falls behind that of the MTR, it shows better recognition rates for Set A and C. The quite inferior results for Set B contributed to the overall performance degradation. As the Set B consists of noisy speech with unknown noise types, the recognition rates for Set B may be increased by applying model parameter compensation approaches for the multiple-model based speech recognizer, which is the topic of our further study.

Table 5. Performance comparison of the multiple-model based speech recognizer with the MTR method (WER(%))

	Set A	Set B	Set C	Average
Conventional Method	9.28	13.24	9.95	11.00
SNRMERG(N=4)	8.94	13.01	8.49	10.48
SNRMERG2(N=1)	8.80	12.72	9.66	10.54
MTR	8.51	8.94	9.83	8.95
SNRMERG2(N=1)+ MTR	8.21	10.66	8.46	9.24

4 Conclusions

As compared to the conventional method where one single reference HMM is chosen as the acoustic model for recognition, we improved the performance of the multiple-model based speech recognizer by selecting N most similar reference HMMs based on the KL distance between the SGM of the training noise signal and the PDF of the noise in the testing noisy speech. We could also increase the recognition rates of the multiple-model based recognizer by using multiple SNR levels for training each of the reference HMM. To further improve the performance of the multiple-model based recognizer, the PDFs of the reference HMMs are interpolated with that of the MTR. The interpolated acoustic model performed better than MTR for the Set A and Set C in the Aurora 2 database. We think that the performance of the multiple-model based recognizer could be further improved by applying model parameter compensation approaches.

Acknowledgments. This work was supported by the Korea Research Foundation(KRF) grant funded by the Korea government(MEST) (No. 2009-0075480).

References

1. Gales, M.J.F.: Model Based Techniques for Noise-Robust Speech Recognition, Ph.D. Dissertation, University of Cambridge (1995)
2. Moreno, P.J.: Speech Recognition in Noisy Environments, Ph.D. Dissertation, Carnegie Mellon University (1996)
3. Ball, S.F.: Suppression of acoustic noise in speech using spectral subtraction. IEEE Trans. Acoust., Speech, Signal Process. 27, 113–120 (1979)
4. Xu, H., Tan, Z.-H., Dalsgaard, P., Lindberg, B.: Robust Speech Recognition on Noise and SNR Classification – a Multiple-Model Framework. In: Proc. Interspeech (2005)
5. ETSI draft standard doc. Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithm, ETSI Standard ES 202 108 (2000)
6. ETSI draft standard doc. Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Advanced Front-end feature extraction algorithm; Compression algorithm, ETSI Standard ES 202 050 (2002)
7. Macho, D., Mauuary, L., Noe, B., Cheng, Y., Eahey, D., Jouvet, D., Kelleher, H., Pearce, D., Saadoun, F.: Evaluation of a noise-robust DSR front-end on Aurora databases. In: Proc. ICSLP, pp. 17–20 (2002)
8. Juang, B.H., Rabiner, L.R.: A Probabilistic Distance Measure for Hidden Markov Models. AT&T Technology Journal, 391–408 (1984)

On Feature Combination for Music Classification

Zhouyu Fu, Guojun Lu, Kai-Ming Ting, and Dengsheng Zhang

Gippsland School of IT, Monash University, Churchill, VIC 3842, Australia
{zhouyu.fu,guojun.lu,kaiming.ting,dengsheng.zhang}@infotech.monash.edu.au

Abstract. We address the problem of combining different types of audio features for music classification. Several feature-level and decision-level combination methods have been studied, including kernel methods based on multiple kernel learning, decision level fusion rules and stacked generalization. Eight widely used audio features were examined in the experiments on multi-feature based music classification. Results on benchmark data set have demonstrated the effectiveness of using multiple types of features for music classification and identified the most effective combination method for improving classification performance.

1 Introduction

Combining multiple features from diverse sources is an effective way to enhance the performance of real-world classification systems. Image classification is one such example that benefited much from feature combination techniques. In recent years, substantial performance gains on challenging benchmark datasets have been reported in the literature [1] by combining multiple features based on different aspects like shape, appearance and texture.

In this paper, we address the problem of using multiple types of features for music classification, which has not yet been adequately investigated in previous studies. Specifically, we have studied a number of candidate schemes for using multiple features, including feature level combination methods such as Multiple Kernel Learning (MKL) for the Support Vector Machine (SVM) classifier [2], and the more general decision level fusion rules such as majority voting, the sum rule [3] and a principled approach to decision fusion called stacked generalization [4,5]. We have adopted the SVM classifier [6] for both individual feature learning and stacked generalization due to its good classification performance for music classification [7]. Moreover, SVM underlies the inherent formulation of MKL, the feature-level combination method discussed in this paper. Hence, it is best to use SVM for all classification tasks involved to make fair comparison of different combination schemes.

The purpose of this paper is to answer the following three questions regarding feature combination for music classification. Firstly, we are interested in the performance of common individual features for music classification. More importantly, we want to know whether combining multiple features is an effective way to enhance the performance of current music classification systems. Finally, we

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 453–462, 2010.

want to identify what is the best feature combination method for the application we study. The answers will be revealed through the empirical evaluation of various feature combination approaches on a benchmark data set.

2 Audio Features for Music Classification

Music classification is an emerging area in multimedia and information retrieval. A key problem in music classification is how to efficiently and effectively extract audio features for high level classification. Many types of features have been used in previous study on music classification [8,9,7,10,11], including low-level features such as timbre and temporal features, and mid-level features such as beats and chords.

Low-level features are normally extracted via spectral analysis, and have been used predominantly in music classification systems due to the simple procedures to obtain them and their good performance. There are two types of local features - timbre and temporal features. Both of them are obtained based on spectral analysis of the audio signal. The basic procedures for timbre and temporal feature extraction are quite similar. First, a song is split into small local windows. The truncated signal segment within each local window is assumed to be stationary, a pre-requisite for the application of various spectral analysis techniques. Standard spectral coefficients are then extracted from each local window, include Fast Fourier Transform Coefficients [10], Mel-Frequency Cepstral Coefficients (MFCC) [9,10,11], Amplitude Spectrum Envelop (ASE) [12,11], and Octave based Spectral Contrast (OSC) [13,11]. Then the coefficients from neighboring local windows are aggregated to produce a single song-level feature. The main difference between timbre and temporal features is in the way local spectral coefficients are aggregated. Timbre features model the distributions of the coefficients, whereas temporal feature modeling treats the coefficients as time series data and concerns their temporal evolution.

Most music classification systems are based on the use of low-level features alone [8,7,10,11]. Alternatively, mid-level features like beats [9] and chords [14] have also been used in some systems to supplement or substitute low-level features. Compared to low-level features, mid-level features can be better interpreted and have more to do with human perception of music. Nevertheless, whether mid-level features are better than low-level features for music classification tasks are still an open question.

Here, our focus is on the combination of different features obtained at different levels to enhance the performance of music classification systems. Each type of feature described above captures some information of music from a different perspective. Hence, they should complement each other for music classification. It is expected that by combining them better classification performance can be achieved. For this purpose, we have used 8 types of individual features in this paper, including three timbre features based on three different spectral coefficients (SMFCC, SASE and SOSC), three temporal features based on the fluctuation pattern [8] of the same three coefficients (TMFCC, TASE and TOSC), as well as two mid-level features of beat (B) [9] and chord (C) [14]. These features, either individually or in combination, have been widely used in music classification and provided good empirical classification performance.

It has been shown in [10] and [11] that combining multiple features can improve over the performance of classification using a single feature type. However, both works are limited in the types of features being investigated. Only low-level features were used in [10], and [11] explored the combination of temporal features alone. It is still unclear whether mid-level features are useful for classification or not. Also, as mentioned earlier, we explore the use of multiple different features under the SVM classification framework, which provides quite strong classification at individual feature level. It is thus not evident whether combination yields any further performance improvement. On the other hand, AdaBoost with decision stumps was used in [10] to combine attributes of features. Many weak classifiers were generated in the process and combined by the AdaBoost framework. Later we can see that the performances of [10] and [11] are inferior to ours on feature combination. It is also worth stressing that our problem is different from feature selection. In feature selection, a subset of attributes is selected from all feature attributes that improves over classification based on the full-length feature vector. In the feature combination problem discussed in this paper, we are given multiple feature vectors for each example. The purpose is to find a way to combine the feature vectors to improve the performance over any single feature vector.

3 Methods for Feature Combination

We begin with a definition of the feature combination problem. Given a labeled data set $\{([x_i^1, \ldots, x_i^M], y_i)\}_{i=1,\ldots,N}$ of size N, where $x_i^m \in \mathbb{R}^{d_m}$ is the mth feature vector with feature dimension d_m for the ith training instance and $y_i \in \{1, \ldots, K\}$ is its class label. The purpose of feature combination is to learn a classification rule $f : \mathcal{X} \to \{1, \ldots, K\}$ from the M feature vectors with $\mathcal{X} \subset \mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_m}$. Depending on the level feature combination is performed, feature combination methods can be categorized into two categories. Decision-level methods learn a classifier for each individual feature type and combine the output of individual classifiers for label prediction without modifying the feature vectors. Feature-level methods combine the individual feature vectors to form a new feature vector for classification. Next, we discuss various feature combination methods in each of the above two categories. Notice that our discussion is far from comprehensive and has omitted many combination methods proposed in the literature. We have only selected a few typical ones that were well represented and most widely used.

3.1 Decision-Level Fusion Methods

In decision level fusion, each individual classifier can return either a single label, a ranking or real-valued output. We assume the latter case that the classifier trained on the mth feature outputs an vector $\mathbf{F}(x^m) = [f^1(x^m), \ldots, f^K(x^m)] \in \mathbb{R}^K$ for a testing instance x. Each entry $f^k(x^m)$ in the vector indicates the confidence value for the kth class. The larger the value of $f^k(x^m)$ relative to the other entries, the more likely that x belongs to class k based on the mth classifier alone. This also includes output labels ($f^k(x^m) = 1$ if x belongs to class k and $f^k(x^m) = 0$ otherwise) and rankings ($\mathbf{F}_i$ is a permutation of $\{1, \ldots, K\}$ with K being the top rank and 1 the lowest rank) as special cases. With the above notations, we now present the fusion schemes in below

Majority Voting is the simplest and most widely used decision level fusion rule. The label of testing instance x is given by

$$\arg \max_{k=1,\ldots,K} \sum_{m=1,\ldots,M} \delta_{k,m}$$

where $\delta_{k,m}$ is a hard decision function which equals 1 if and only if the mth classifier votes for class k, that is, $f^k(x^m)$ is larger than other $f^j(x^m)$'s for $j \neq k$. Despite its simplicity, majority voting ignores the values of classifier output which encode confidence levels on prediction. To fix it, an alternative fusion rule like the sum rule can be used.

Sum Rule uses decision values $f^k(x^m)$'s directly in aggregation

$$\arg \max_{k=1,\ldots,K} \sum_{m=1,\ldots,M} f^k(x^m) \tag{1}$$

Hence the larger $f^k(x^m)$ is, the more it contributes to the final score for class k.

Besides majority voting and sum rules, a number of alternative rules can be used with simple algebra operations. The discussion can be found in the overview paper [3]. A probabilistic framework is also developed in [3] that incorporates all fusion rules as special cases. In the probabilistic framework, $f^k(x^m)$'s become the posterior probabilities. However, to use the above two fusion rules, we only have to assume that classifier output $f^k(x^m)$'s are proportional to the posterior probabilities, instead of requiring them to be probabilistic output. This is a reasonable assumption since larger value of $f^k(x^m)$ indicates a higher likelihood of class k. Thus we can generalize these rules to take real-valued scores.

Stacked Generalization

The above fusion rules are defined in an unsupervised fashion without using the label information in the training data. Stacked generalization [4] provides a principled framework for learning supervised decision rules. It treats the output values $f^k(x^m)$'s returned by individual classifiers as new features that can be used for classification. Specifically, it creates the following feature map for training instance x_i

$$\mathbf{F}_i = [f^1(x_i^1), \ldots, f^K(x_i^1), \ldots, f^1(x_i^M), \ldots, f^K(x_i^M)] \in \mathbb{R}^{KM} \tag{2}$$

where the first K feature elements are taken from decision values returned by the first classifier, the next K feature elements are values returned by the second classifier, and so on. The total feature dimension is KM for K-class classification with M features. A new training set can then be constructed $\{(\mathbf{F}_i, y_i)\}_{i=1,\ldots,N}$ on top of which a new classifier is learned. Any forms of classifier can be used for stacked generalization but linear classifiers are preferred due to its efficiency and performance [5].

Here, we adopted the linear SVM classifier which, in the binary case, can be learned by solving the following optimization problem

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 + \lambda \sum_{i=1}^{N} \max\left(0, 1 - y_i(\langle \mathbf{w}, \mathbf{F}_i \rangle + b)\right) \tag{3}$$

where $y_i \in \{1, -1\}$ denotes the binary class label for the ith instance, $\mathbf{w}$ and b are the weight and bias of the linear discriminant function. In the above equation, the first term is the regularization term, whereas the second term specifies a Hinge loss on misclassification. λ controls the trade-off between the two terms.

The SVM is best solved in its dual form in the following

$$\max_{0 \le \alpha \le \lambda} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j K(\mathbf{F}_i, \mathbf{F}_j) \qquad \text{s.t.} \qquad \sum_i \alpha_i y_i = 0 \tag{4}$$

where α_i's are dual variables and $K(\mathbf{F}_i, \mathbf{F}_j) = \langle \mathbf{F}_i, \mathbf{F}_j \rangle$ is the kernel function defined as the inner product between two feature vectors. Multi-class classification problems are tackled with a one-vs-all strategy by training K classifiers to differentiate between class k and non-class k for $k = 1, \ldots, K$. A testing instance is assigned to the class with the largest output value.

3.2 Feature-Level Combination Methods

Feature-level combination methods are usually developed under specific classification framework. Here we focus on feature-level combination with the SVM classifier. Despite the simplicity of linear SVM, it can not handle nonlinear data. The nonlinear SVM classifier is usually used instead for real-world classification problems. This is achieved by utilizing the kernel trick. Specifically, let ϕ define a nonlinear feature mapping for feature vector $\mathbf{F}_i$. The explicit form of $\phi(\mathbf{F}_i)$ is unknown, but the inner product between two nonlinear features is well defined by the kernel function $K(\mathbf{F}_i, \mathbf{F}_j) = \langle \phi(\mathbf{F}_i), \phi(\mathbf{F}_j) \rangle$. In this case, we can solve the dual formulation in Equation 4 by plugging into a different kernel function. Common nonlinear kernels include Gaussian, polynomial and sigmoid kernels.

Feature Concatenation is the most straightforward feature-level operation to form a composite feature by concatenating all individual features. The composite feature is a long feature vector given by $\mathbf{x} = [\mathbf{x}^1, \ldots, \mathbf{x}^M]$, which can be used for feature classification. In the case of SVM, it is used for computing the new kernel $K(\mathbf{x}_i, \mathbf{x}_j)$.

Kernel Averaging
Alternatively, we can do feature concatenation in the implicit feature space given by the mapping function ϕ. This is equivalent to averaging the kernels induced

by the individual feature maps. Specifically, the concatenated nonlinear feature is given by $\phi(\mathbf{x}) = \frac{1}{\sqrt{M}}[\phi(\mathbf{x}^1), \ldots, \phi(\mathbf{x}^M)]$. This is equivalent to defining a composite kernel function

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = \frac{1}{M} \sum_{m=1}^{M} \langle \phi(\mathbf{x}_i^m), \phi(\mathbf{x}_j^m) \rangle = \frac{1}{M} \sum_{m=1}^{M} K(\mathbf{x}_i^m, \mathbf{x}_j^m) \quad (5)$$

The composite kernel is the average of all individual kernels. We can solve the SVM dual formulation using the composite kernel.

Multiple Kernel Learning (MKL)
Instead of using uniform weights for the composite kernel in Equation 5, a more general formulation is introduced in [2] for learning kernel weights based on the SVM dual formulation,

$$\min_{\beta \geq 0} \max_{0 \leq \alpha \leq \lambda} \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j K_\beta(\mathbf{x}_i, \mathbf{x}_j) \quad (6)$$

$$with \quad K_\beta(\mathbf{x}_i, \mathbf{x}_j) = \sum_{m=1}^{M} \beta_m K(\mathbf{x}_i^m, \mathbf{x}_j^m)$$

$$s.t. \quad \sum_i \alpha_i y_i = 0 \quad \text{and} \quad \sum_m \beta_m = 1$$

The above problem is generally referred to as multiple kernel learning [2]. The objective function is still convex and hence can be minimized effectively. The solution provides both a set of feature kernel weights as well as the dual variables used to define the nonlinear decision function $f(\mathbf{x}) = \sum_i \alpha_i y_i K_\beta(\mathbf{x}_i, \mathbf{x}) + b$. For multiclass classification, we learn kernel weights jointly for all classes by taking the sum of objective functions for each class and fixing β values in Equation 6.

3.3 Recursive Feature Elimination

Inspired by the idea of [15], we develop a procedure for recursively eliminating redundant features based on stacked generalization. Let $w_{k,m}^j$ be the weight of the jth linear SVM for decision value $f^k(x^m)$, we then define the relevance of the mth feature by the measure $\sum_{j=1}^{K} \sum_{k=1}^{K} \left(w_{k,m}^j\right)^2$. The larger the relevance, the more useful the feature is for classification. Starting with the full set of features, we can then take the following steps for recursive feature elimination

1. Learn the feature combination model with the remaining features
2. Eliminate the feature with the lowest relevance
3. Repeat the above two steps until the desired number of features is reached

Through the recursive feature elimination procedure, we can determine the importance of each individual feature for classification and retain a subset of features for combination. It also produces a ranking for the individual features based on the order they are eliminated. From the ranking, we can determine the relative importance of individual features.

4 Experimental Results

In this section, we perform feature combination experiments on a benchmark data set for music genre classification. We used the GTZAN data set [9], which contains 1000 song segments in 30 seconds of duration uniformly distributed from 10 genres. For each song, we have extracted eight individual features as described in Section 2. Each classification experiment was repeated 20 times with different random partitioning of training and testing data. For each round, half of the examples in the data set were randomly selected for training and the remaining for testing. All features were examined on the same training and testing set in each round. The LibSVM package[1] and the Gaussian kernel was used for SVM training. To reduce the scaling effect, each feature attribute has been scaled to zero mean and unit standard deviation for kernel computation. We have also normalized the kernel matrix to unit mean so as to reduce the scaling effect at kernel level. SVM and kernel parameters were chosen via 3 fold cross validation on the training data.

First, we examine the effectiveness of each individual feature set for music genre classification using the SVM classifier. Table 1 shows the average accuracy rates over 20 rounds achieved by individual features for i) each genre class by treating the target genre as the positive class and the other genres as the negative class; and ii) the 10-class problem by classifying each song into one of the 10 genre classes. Accuracy rates for the top performing features are highlighted in bold for each genre and the 10-class problem. These include the feature type with the highest average accuracy and other features with close performances. That is, the differences in accuracy rates between those features and the top feature are not statistically significant based on the outcomes of paired t-tests within 95% of confidence interval. From the results in Table 1, we can see that chord feature is the best individual feature type for genre classification achieving top performances for 6 out of 10 genres as well as the 10-class problem. Most of the other features obtain similar classification performance except the beat feature, which has significantly lower accuracy rates. However, although being ineffective in overall, beat is best in identifying the disco. This is consistent with our perception of the disco, which is distinguished by its faster rhythm and more frequent beats compared to other music genres.

We now turn our attention to feature combination. We compare the various feature combination schemes discussed in Section 3, including three feature-level combination methods ("FC" for feature concatenation, "AvgK" for kernel average and "MKL" for multiple kernel learning) and three decision-level schemes ("Vote" for majority vote, "Sum" for sum rule, "SG" for stacked generalization). The same setup was adopted from the previous experiment with the same training and testing set partitions and tools for SVM training. The output values of SVM classifiers over individual features were directly used by Vote, Sum and SG with optimal parameters selected from cross validation. The regularization parameter for SG was also chosen from cross validation. Table 2 shows the

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Table 1. Accuracy rates in percentage for individual feature sets

	SMFCC	SASE	SOSC	TMFCC	TASE	TOSC	Beat	Chord
Blues	75.90	64.40	76.90	73.20	72.00	78.80	18.60	**83.20**
Classical	93.00	91.50	**94.80**	**95.80**	92.10	**94.30**	29.40	90.20
Country	68.20	72.20	**76.10**	72.70	71.20	**75.50**	17.70	69.90
Disco	63.30	56.60	63.00	63.20	**69.10**	66.20	**71.60**	54.10
Hiphop	68.90	65.20	72.40	73.80	77.50	74.90	27.10	**96.60**
Jazz	82.40	82.50	83.40	87.80	80.10	80.00	16.00	**98.70**
Pop	**90.00**	87.90	86.70	**89.40**	87.20	85.40	19.40	65.00
Metal	**77.10**	**75.60**	71.40	66.10	71.80	**74.80**	17.90	**77.90**
Reggae	63.80	60.40	65.90	67.60	65.90	67.60	18.10	**81.30**
Rock	52.90	23.70	40.40	48.50	45.10	52.50	10.80	**72.30**
10-class	73.55	68.00	73.10	73.81	73.20	75.00	24.66	**78.92**

Table 2. Accuracy rates in percentage for various feature combination methods

	Best	Vote	FC	AvgK	Sum	MKL	SG
Blues	83.20	86.30	**89.60**	**94.20**	**91.70**	**93.70**	**95.70**
Classical	95.80	96.80	97.00	97.20	96.60	97.50	97.00
Country	76.10	**82.90**	**83.60**	**88.50**	**85.50**	**89.40**	**89.40**
Disco	71.60	**77.60**	**83.00**	**83.70**	**86.10**	**86.30**	**86.60**
Hiphop	96.60	85.90	86.60	91.90	93.40	93.00	93.30
Jazz	98.70	92.00	91.10	96.30	98.60	97.90	98.40
Pop	90.00	91.10	90.40	**92.50**	**92.30**	**93.80**	**96.30**
Metal	77.90	78.60	**80.60**	**88.00**	**87.80**	**89.70**	**87.80**
Reggae	81.30	82.90	75.10	**84.60**	**87.50**	**86.50**	**85.10**
Rock	72.30	68.80	70.50	73.90	**78.50**	**76.00**	**78.90**
10-class	78.92	**84.29**	**84.75**	**89.08**	**89.80**	**90.38**	**90.85**

average accuracy rates achieved by different feature combination methods over 20 rounds for each music genre. For comparison purpose, we have also included the best results returned by the optimal individual feature in the table.

We have highlighted in bold the accuracy rates for combination schemes that outperform the best individual feature in each row of Table 2. This is again determined by comparing the differences in their accuracy rates using paired t-tests. It can be seen from Table 2 that feature combination can much improve the performance of music classification, regardless of the specific combination method being used. Even simple fusion rule like majority voting performs significantly better than the top individual feature for the 10-class problem. Feature combination is effective for 7 out of 10 genres with improved accuracy rates, and the improvement is more evident for those genres that no individual feature can do very well, like disco and country music. The columns of Table 2 are ranked by classification performances for the 10-class problem, with increasing average accuracy rates from left to right. Among the top four combination schemes,

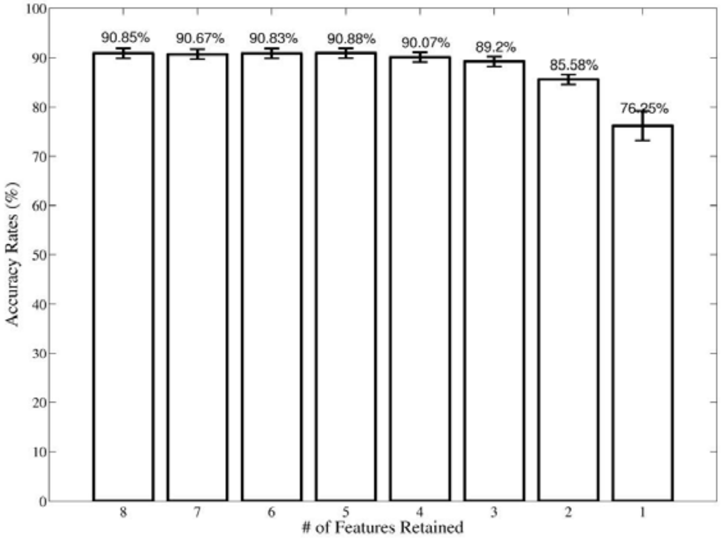


Fig. 1. Results of recursive feature elimination for classification

which achieve significantly better performances than others, decision level fusion schemes (SG and sum rule) perform slightly better than feature level schemes (MKL and average kernel) in overall. Supervised combination schemes (SG and MKL) also outperform their unsupervised counterparts (sum rule and average kernel). Significance tests on the 10-class accuracy rates further corroborates our findings, showing that the differences in accuracy rates obtained by any pair of methods are statistically significant, except for SG versus MKL and MKL versus sum rule.

It is worth mentioning that the best 10-class accuracy rate of 90.9% achieved by SG outperforms the state-of-the-art genre classification results reported in [10] (83%) and [11] (79.6%[2]), while both of them adopted a multiple feature approach albeit with weaker learners on individual features. This empirically justifies the strength of feature combination with a strong classifier like SVM.

Finally, we examine the relative importance of individual features by applying the recursive feature elimination procedure to the eight audio features using stacked generalization. Figure 1 shows the bar plot of accuracy rates varied against the number of features retained. The error bars in the plot represent the standard deviation over 20 rounds. It can be clearly seen that the accuracy rates are quite consistent with the elimination of the least relevant features. The feature elimination scheme also provides feature ranking results. Ranks returned by different testing rounds are different, but an overall ranking can be determined

[2] The accuracy rate of 91% reported in [11] is based on 10-fold cross validation. The result here is based on our implementation of the algorithm and tested for 50%-50% split of training/testing data.

by sorting the average rank order. From the overall ranking, we find that beat is the weakest individual feature and always the first one to be eliminated, whereas chord is the strongest feature and usually the last one to remain.

5 Conclusions

We have studied the problem of multiple feature combination for music classification. Empirical validation showed that the classification performance is much improved by using multiple features at different levels regardless of the combination schemes adopted. Moreover, we have also identified chord as the single best feature for music genre classification and stacked generalization as the optimal combination scheme for multiple feature combination.

References

1. Varma, M., Ray, D.: Learning the discriminative power invariance trade-off. In: Intl. Conf. on Computer Vision (2007)
2. Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L.E., Jordan, M.I.: Learning the kernel matrix with semidefinite programming. Journal of Machine Learning Research 5, 27–72 (2004)
3. Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(3), 226–239 (1998)
4. Wolpert, D.: Stacked generalization. Neural Networks 5(2), 241–259 (1992)
5. Ting, K.M., Witten, I.: Issues in stacked generalization. Journal of Artificial Intelligence Research 10, 271–289 (1999)
6. Boser, B.E., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: ACM Conf. Computational Learning Theory, pp. 144–152 (1992)
7. Mandel, M., Ellis, D.: Song-level features and svms for music classification. In: Intl. Conf. Music Information Retrieval (2005)
8. Pampalk, E., Rauber, A., Merkl, D.: Content-based organization and visualization of music archives. In: ACM Multimedia (2002)
9. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Trans. Speech and Audio Processing 10(5), 293–302 (2002)
10. Bergstra, J., Casagrande, N., Erhan, D., Eck, D., Kegl, B.: Aggregate features and ada boost for music classification. Machine Learning 65(2-3), 473–484 (2006)
11. Lee, C.H., Shih, J.L., Yu, K.M., Lin, H.S.: Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features. IEEE Trans. Multimedia 11(4), 670–682 (2009)
12. Kim, H.G., Moreau, N., Sikora, T.: Audio classification based on mpeg-7 spectral basis representation. IEEE Trans. Circuits and Systems for Video Technology 14(5), 716–725 (2004)
13. Lu, L., Liu, D., Zhang, H.J.: Automatic mood detection and tracking of music audio signals. IEEE Trans. Speech and Audio Processing 14(1), 5–18 (2006)
14. Cheng, H.T., Yang, Y.H., Lin, Y.C., Liao, I.B., Chen, H.H.: Automatic chord recognition for music classification and retrieval. In: Intl. Conf. Multi. Expo. (2008)
15. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46(1-3), 389–422 (2002)

Information Theoretical Kernels for Generative Embeddings Based on Hidden Markov Models

André F.T. Martins[3], Manuele Bicego[1,2], Vittorio Murino[1,2],
Pedro M.Q. Aguiar[4], and Mário A.T. Figueiredo[3]

[1] Computer Science Department, University of Verona - Verona, Italy
[2] Istituto Italiano di Tecnologia (IIT) - Genova, Italy
[3] Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa, Portugal
[4] Instituto de Sistemas e Robótica, Instituto Superior Técnico, Lisboa, Portugal

Abstract. Many approaches to learning classifiers for structured objects (*e.g.*, shapes) use generative models in a Bayesian framework. However, state-of-the-art classifiers for vectorial data (*e.g.*, support vector machines) are learned discriminatively. A generative embedding is a mapping from the object space into a fixed dimensional feature space, induced by a generative model which is usually learned from data. The fixed dimensionality of these feature spaces permits the use of state of the art discriminative machines based on vectorial representations, thus bringing together the best of the discriminative and generative paradigms.

Using a generative embedding involves two steps: (i) defining and learning the generative model used to build the embedding; (ii) discriminatively learning a (maybe kernel) classifier on the adopted feature space. The literature on generative embeddings is essentially focused on step (i), usually adopting some standard off-the-shelf tool (e.g., an SVM with a linear or RBF kernel) for step (ii). In this paper, we follow a different route, by combining several Hidden Markov Models-based generative embeddings (including the classical Fisher score) with the recently proposed non-extensive information theoretic kernels. We test this methodology on a 2D shape recognition task, showing that the proposed method is competitive with the state-of-art.

1 Introduction

Many approaches to the statistical learning of classifiers belong to one of two paradigms: generative and discriminative [24,20]. Generative approaches are built upon probabilistic class models and *a priori* class probabilities, which are learnt from training data and combined via Bayes law to yield posterior probabilities. Discriminative methods aim at learning class boundaries, or posterior class probabilities, directly from data, without resorting to generative class models.

In generative approaches for data sequence, *hidden Markov models* (HMMs) [23] are widely used and their usefulness has been shown in different applications. Nevertheless, generative approaches can yield poor results for a variety of possible reasons, such as model mismatch due to the lack of prior knowledge, poor model estimates due to insufficient training data, for instance. To face this issue,

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 463–472, 2010.

several efforts have been recently made to enrich the generative paradigm with discriminative information. This may be achieved via discriminative training of HMMs using, for example, the *maximum mutual information* (MMI) [2] or the *minimum Bayes risk* (MBR) [15] criteria (see also [11]). Alternatively, there exist generalizations of HMMs towards probabilistic discriminative models, such as *conditional random fields* (CRFs) [16], in which conditional maximum likelihood is used to estimate the model parameters. The so-called generative embeddings methods (or generative score spaces) are another recently explored approach: the basic idea is to use the HMM (or some other generative model) to map the objects to be classified into a feature space, where discriminative techniques, possibly kernel-based, can be used.

The seminal work on generative embedding introduced the so-called *Fisher score* [13]. In that work, the features of a given object are the derivatives of the log-likelihood function under the assumed generative model, with respect to the model parameters, computed for that object. Other examples of generative embeddings can be found in [4,7,22,5], some of which are general while others are specifically tailored to a particular generative model.

Using a generative embedding involves two steps: (i) defining and learning the generative model and using it to build the embedding; (ii) discriminatively learning a (maybe kernel) classifier on the adopted score space. The literature on generative embeddings is essentially focused on step (i), usually using some standard off-the-shelf tool for step (ii) – e.g., some kernel-based classifier, namely, a *support vector machine* (SVM) using classical linear or radial basis function (RBF) kernels.

In this paper, we adopt a different approach, by focusing also on the discriminative learning step. In particular, we combine some HMM-based generative embeddings with the recently introduced information theoretic kernels [17]. These new kernels, which are based on a non-extensive generalization of the classical Shannon information theory, are defined on (possibly unnormalized) probability measures. In [17], they were successfully used in text categorization tasks, based on multinomial (bag-of-words type) text representations. Here, the idea is to consider the points of the generative embedding as multinomial probability distributions, thus valid arguments for the information theoretic kernels.

The proposed approach is instantiated with four different HMM-based generative embeddings into feature spaces (the *Fisher score embedding* [13], the *marginalized kernel space* [27], the *state space* and the *transition space* [5]) and four information theoretic kernels [17] (the *Jensen-Shannon kernel*, the *Jensen-Tsallis kernel*, and two versions of the *weighted Jensen-Tsallis kernel*). The experimental evaluation is performed using a 2D shape classification problem, obtaining results confirming the validity of the proposed approach.

2 HMM-Based Generative Embeddings

2.1 Hidden Markov Models

In this subsection, we briefly summarize the basic concepts of HMMs, mainly to set up the notation.

A discrete-time first order HMM [23] is a probabilistic model that describes a stochastic sequence[1] $\boldsymbol{O} = (O_1, O_2, \ldots, O_T)$ as being an indirect observation of a hidden Markovian random sequence of states $\boldsymbol{Q} = (Q_1, Q_2, \ldots, Q_T)$, where, for $t = 1, ..., T$, $Q_t \in \{1, 2, \ldots, N\}$ (the set of states). Each state has an associated probability function that specifies the probability of observing each possible symbol, given the state. An HMM is thus fully specified by a set of parameters $\boldsymbol{\lambda} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$ where $\boldsymbol{A} = (a_{ij})$ is the transition matrix, i.e., $a_{ij} = P(Q_t = j \,|\, Q_{t-1} = i)$, $\boldsymbol{\pi} = (\pi_i)$ is the initial state probability distribution, i.e., $\pi_i = P(Q_1 = i)$, and $\boldsymbol{B} = (\boldsymbol{b}_i)$, is the set of emission probability functions. If the observations are continuous, each $\boldsymbol{b}_i$ is a probability density function, e.g., a Gaussian or a mixture of Gaussians. If the observations belong to a finite set $\{v_1, v_2 ..., v_S\}$, each $\boldsymbol{b}_i = (b_i(v_1), b_i(v_2), ..., b_i(v_S))$ is a probability mass function with $b_i(v_s) = P(O_t = v_s \,|\, Q_t = i)$ being the probability of emitting symbol v_s in state i.

2.2 The Embeddings

The generative embedding can be defined as a function Φ which maps an observed sequence $\boldsymbol{o} = (o_1, ..., o_T)$ into a vector, by employing a set of HMMs $\boldsymbol{\lambda}_1, ..., \boldsymbol{\lambda}_C$. Different approaches have been proposed to determine the set of models used to build the embedding [3]. Here, we adopt the following method: given a C-ary classification problem, we train one HMM for each class, and concatenate the vectors obtained by the embedding of each model, i.e.,

$$\Phi(\boldsymbol{o}) = [\phi(\boldsymbol{o}, \boldsymbol{\lambda}_1), \cdots, \phi(\boldsymbol{o}, \boldsymbol{\lambda}_C)] \,. \tag{1}$$

Below, we describe how $\phi(\boldsymbol{o}, \boldsymbol{\lambda}_c)$ is defined in the four cases considered in this paper. All the quantities needed to compute the different embeddings can be easily obtained using the forward-backward procedure [23].

The Fisher Score Embedding (FSE). In the FSE, each sequence is represented by a feature vector containing derivatives of the log-likelihood of the generative model with respect to each of its parameters. Formally,

$$\phi^{\mathrm{FSE}}(\boldsymbol{o}, \boldsymbol{\lambda}) = \left[\frac{\partial \log(P(\boldsymbol{O} = \boldsymbol{o} | \boldsymbol{\lambda})}{\partial \lambda_1}, \cdots, \frac{\partial \log(P(\boldsymbol{O} = \boldsymbol{o} | \boldsymbol{\lambda})}{\partial \lambda_L} \right]^\top \in \mathbb{R}^L, \tag{2}$$

where λ_i represents one of the L parameters of the model $\boldsymbol{\lambda}$ (elements of the transition matrices, emission and initial probabilities). For more details, see [9].

The Marginalized Kernel Embedding (MKE). The marginalized kernel (MK) for discrete HMMs is defined as

$$\mathrm{MK}\,(\boldsymbol{o}, \boldsymbol{o}', \boldsymbol{\lambda}) = \sum_{s=1}^{S} \sum_{i=1}^{N} m_{si}\,(\boldsymbol{o}, \boldsymbol{\lambda})\, m_{si}\,(\boldsymbol{o}', \boldsymbol{\lambda})\,, \tag{3}$$

[1] We adopt the common convention of writing stochastic variables with upper case and realizations thereof in lower case.

with

$$m_{si}\left(\boldsymbol{o},\boldsymbol{\lambda}\right)=\frac{1}{T}\sum_{\boldsymbol{q}\in\{1,\dots,N\}^T}P\left(\boldsymbol{Q}=\boldsymbol{q}|\boldsymbol{O}=\boldsymbol{o},\boldsymbol{\lambda}\right)\sum_{t=1}^{T}I\left(o_t=s\wedge q_t=i\right), \quad (4)$$

where the indicator function $I(A)$ is 1 if A is true and 0 otherwise [27].

Let us collect all the $m_{si}\left(\boldsymbol{o},\boldsymbol{\lambda}\right)$ values, for $s=1,...,S$ and $i=1,...,N$, into an (SN)-dimensional vector $\boldsymbol{m}(\boldsymbol{o},\boldsymbol{\lambda})\in\mathbb{R}^{SN}$. Then, it is clear that

$$\mathrm{MK}\left(\boldsymbol{o},\boldsymbol{o}',\boldsymbol{\lambda}\right)=\langle\boldsymbol{m}(\boldsymbol{o},\boldsymbol{\lambda}),\boldsymbol{m}(\boldsymbol{o}',\boldsymbol{\lambda})\rangle \quad (5)$$

showing that the MK is nothing but a linear kernel. The MKE is thus simply given by

$$\phi^{\mathrm{MKE}}(\boldsymbol{o},\boldsymbol{\lambda})=\boldsymbol{m}(\boldsymbol{o},\boldsymbol{\lambda})\in\mathbb{R}^{SN}. \quad (6)$$

The State Space Embedding (SSE). The SSE is a recently introduced generative embedding [5], in which the i-th component of the feature vector mesures, for an observed sequence $\boldsymbol{o}$, the sum (over time) of the probabilities of finding the HMM specified by $\boldsymbol{\lambda}$ in state i. Formally,

$$\phi^{\mathrm{SSE}}(\boldsymbol{o},\boldsymbol{\lambda})=\left[\sum_{t=1}^{T}P(Q_t=1|\boldsymbol{o},\boldsymbol{\lambda}),\cdots,\sum_{t=1}^{T}P(Q_t=N|\boldsymbol{o},\boldsymbol{\lambda})\right]^{\top}\in\mathbb{R}^{N} \quad (7)$$

Each component can be interpreted as the expected number of transitions from the corresponding state, given the observed sequence [23].

The Transition Embedding (TE). This embedding is similar to the SSE but it considers probabilities of transitions rather than states. Naturally, it is defined as

$$\phi^{\mathrm{TE}}(\mathbf{O},\boldsymbol{\lambda})=\begin{bmatrix}\sum_{t=1}^{T-1}P(Q_t=1,Q_{t+1}=1|\boldsymbol{o},\boldsymbol{\lambda})\\ \sum_{t=1}^{T-1}P(Q_t=1,Q_{t+1}=2|\boldsymbol{o},\boldsymbol{\lambda})\\ \vdots\\ \sum_{t=1}^{T-1}P(Q_t=N,Q_{t+1}=N|\boldsymbol{o},\boldsymbol{\lambda})\end{bmatrix}\in\mathbb{R}^{N^2} \quad (8)$$

Each of the N^2 components of the vector can be interpreted as the expected number of transitions from a given state to another state, given the observed sequence [23].

3 Information Theoretic Kernels

Kernels on probability measures have been shown very effective in classification problems involving text, images, and other types of data [10,12,14]. Given two probability measures p_1 and p_2, representing two objects, several information theoretic kernels (ITKs) can be defined [17]. The Jensen-Shannon kernel is defined as

$$k^{\mathrm{JS}}(p_1,p_2) = \ln(2) - JS(p_1,p_2), \tag{9}$$

with $JS(p_1,p_2)$ being the Jensen-Shannon divergence

$$JS(p_1,p_2) = H\left(\frac{p_1+p_2}{2}\right) - \frac{H(p_1)+H(p_2)}{2}, \tag{10}$$

where $H(p)$ is the usual Shannon entropy.

The Jensen-Tsallis (JT) kernel is given by

$$k_q^{\mathrm{JT}}(p_1,p_2) = \ln_q(2) - T_q(p_1,p_2), \tag{11}$$

where $\ln_q(x) = (x^{1-q}-1)/(1-q)$ is the q-logarithm,

$$T_q(p_1,p_2) = S_q\left(\frac{p_1+p_2}{2}\right) - \frac{S_q(p_1)+S_q(p_2)}{2^q} \tag{12}$$

is the Jensen-Tsallis q-difference, and $S_q(r)$ is the Jensen-Tsallis entropy, defined, for a multinomial $r=(r_1,...,r_L)$, with $r_i \geq 0$ and $\sum_i r_i = 1$, as

$$S_q(r_1,...,r_L) = \frac{1}{q-1}\left(1-\sum_{i=1}^{L} r_i^q\right).$$

In [17], versions of these kernels applicable to unnormalized measures were also defined. Let $\mu_1 = \omega_1 p_1$ and $\mu_2 = \omega_2 p_2$ be two unnormalized measures, where p_1 and p_2 are the normalized counterparts (probability measures), and ω_1 and ω_2 arbitrary positive real numbers (weights). The weighted versions of the JT kernels are defined as follows:

- The weighted JT kernel (version A) is given by

$$k_q^A(\mu_1,\mu_2) = S_q(\pi) - T_q^\pi(p_1,p_2), \tag{13}$$

where $\pi = (\pi_1,\pi_2) = \left(\frac{\omega_1}{\omega_1+\omega_2}, \frac{\omega_2}{\omega_1+\omega_2}\right)$ and

$$T_q^\pi(p_1,p_2) = S_q(\pi_1 p_1 + \pi_2 p_2) - \left(\pi_1^q S_q(p_1) + \pi_2^q S_q(p_2)\right).$$

- The weighted JT kernel (version B) is defined as

$$k_q^B(\mu_1,\mu_2) = \big(S_q(\pi) - T_q^\pi(p_1,p_2)\big)(\omega_1+\omega_2)^q. \tag{14}$$

4 Proposed Approach

The approach proposed in this paper consists in defining a kernel between two observed sequences $\boldsymbol{o}$ and $\boldsymbol{o}'$ as the composition of one of generative embeddings with one of the ITKs presented above. Formally,

$$k(\boldsymbol{o}, \boldsymbol{o}') = k_q^i \left(\Phi(\boldsymbol{o}), \Phi(\boldsymbol{o}')\right), \tag{15}$$

where $i \in \{\text{JT, A, B}\}$ indexes one of the Jensen-Tsallis kernels (11), (13), or (14), and Φ is as given in (1), where ϕ is one the embeddings reviewed in Section 2.2. Notice that this kernel is well defined because all the components of $\Phi(\boldsymbol{o})$ are non-negative, for any $\boldsymbol{o}$; see (4), (7), and (8). In the case of the FSE, positivity is guaranteed by adding a positive offset to all the components of ϕ^{FSE}. The family of kernels ${k_q}^{\text{JT}}$ requires the arguments to be proper probability mass functions, which can be easily achieved by normalization. For the kernels k_q^A and k_q^B, this normalization is not required, so we also consider un-normalized arguments.

We use this kernel with support vector machine (SVM) classifiers. Recall that positive definiteness is a key condition for the applicability of a kernel in SVM learning. It was shown in [17] that k_q^A is a positive definite kernel for $q \in [0, 1]$, while k_q^B is a positive definite kernel for $q \in [0, 2]$. Standard results from kernel theory [25, Proposition 3.22] guarantee that the kernel k defined in (15) inherits the positive definiteness of k_q^i, thus can be safely used in SVM learning algorithms.

5 Experimental Evaluation

We tested the proposed approach on a 2D shape recognition task. For each shape, a sequence of curvature values is extracted from the corresponding contour, as in [19]. The sequences of curvatures are subsequently modeled by continuous 3-state HMMs with Gaussian emission densities.

We use the Chicken Pieces Database, denoted also as *Chicken* data[2] [1]. This dataset contains 446 binary images (silhouettes) of chicken pieces, each belonging to one of five classes representing specific chicken parts: wings (117 samples), backs (76), drumsticks (96), thighs and backs (61), and breasts (96). Some examples of this dataset are shown in Fig. 1. This constitutes a challenging classification task, which has been recently used as a benchmark by several authors [3,6,8,18,19,21,22].

The original set is split randomly into training and test sets (of equal size). The classification accuracy values reported in Table 1 are averages over 10 experiments. The constant C of SVMs and the parameter q of the information theoretic kernels was optimized by 10-fold cross validation (CV). The embeddings have been used with or without a space standardization (moving and scaling every feature). Actually, it has shown that, depending on the embedding, adequate standardization may often be crucial in obtaining high accuracy values [5,26].

[2] `http://algoval.essex.ac.uk:8080/data/sequence/chicken/`

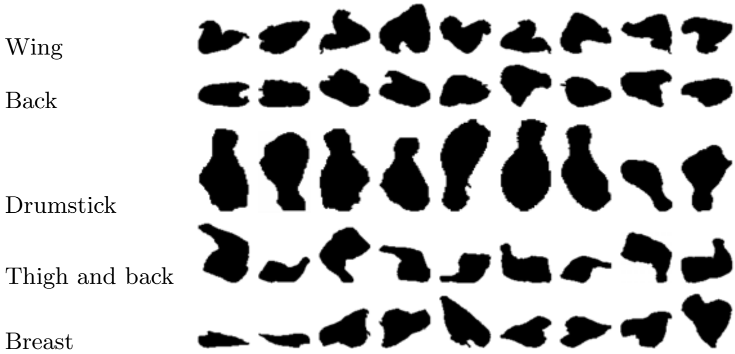


Fig. 1. Examples of Chicken data

Table 1. Classification accuracies obtained with the several embeddings and information theoretic kernels described in the text on the 2D shape recognition experiment. The rows with the indication "standardized" refer to experiments where the embeddings were standardized.

Embedding	Linear	$k^{\text{JS}} = k_1^{\text{JT}}$	k_q^{JT}	k_q^A	k_q^B
States	0.7387	0.7230	0.7095	0.7995	0.8221
States (standardized)	0.7342	0.7230	0.7005	0.8086	0.7950
Transitions	0.7703	0.7545	0.7545	0.8243	0.8356
Transitions (standardized)	0.8311	0.7995	0.7973	0.8176	0.8198
Fisher	0.6171	0.6194	0.6261	0.7568	0.6689
Fisher (standardized)	0.8108	0.8243	0.8243	0.8311	0.8243
Marginalized	0.6712	0.7095	0.7455	0.8243	0.8063
Marginalized (standardized)	0.7477	0.6937	0.7162	0.7995	0.8063

The results in Table 1 show that, except in one case, the best Jensen-Tsallis kernel for each embedding always outperforms the linear kernel, although not by much.

Figure 2 plots the SVM accuracies, for different kernels, as a function of parameter q, for the *transitions embedding* (TE). In line with the results from [17], the best performances are obtained for $q < 1$. Although we do not have, at this moment, a formal justification for this fact, it may be due to the following behavior of the JT kernels. For $q < 1$, the maximizer of $k_q^{\text{JT}}(p, v)$ (or of $k_q^{\text{B}}(p, v)$) with respect to p is not v, but another distribution closer to uniform. This is not the case for the Jensen-Shannon kernel k^{JS} (which coincides with k_1^{JT}), for which the minimizer of $k^{\text{JS}}(p, v)$ with respect to p is precisely v. This behavior of k_q^{JT} plays the role of a smoothing regularizer, by favoring more uniform distributions.

Finally, Table 2 reports some recent state-of-the-art results on the Chicken Pieces dataset. The experimental procedures are not the same in all the references listed in the table (different shape representations, different numbers

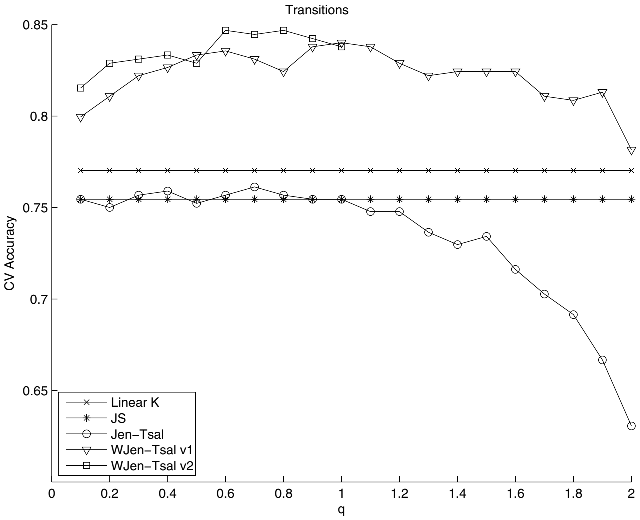


Fig. 2. SVM accuracies with several kernels for the transitions embedding, as a function of q. Notice that the maximum accuracy in this plot is higher than that reported in Table 1, since that value was obtained with q adjusted by cross validation.

Table 2. Comparative Results on the *Chicken* data

Methodology	Accuracy (%)	Reference
1-NN + Levenshtein edit distance	≈ 0.67	[18]
1-NN + approximated cyclic distance	≈ 0.78	[18]
KNN + cyclic string edit distance	0.743	[19]
SVM + Edit distance-based kernel	0.811	[19]
1-NN + mBm-based features	0.765	[6]
1-NN + HMM-based distance	0.737	[6]
SVM + HMM-based entropic features	0.812	[21]
SVM + HMM-based Top Kernel	0.808	[22]
SVM + HMM-based FESS embedding + rbf	0.830	[22]
SVM + HMM-based non linear Marginalized Kernel	0.855	[8]
SVM + HMM-based clustered Fisher kernel	0.858	[3]

of HMM states, different accuracy assessment protocol), so the results should not be interpreted too strictly. However, we can observe that the best result from Table 1 (0.836) would be in third place (2.2% behind the best) in the ranking of methods shown in Table 2, thus we can conclude that this preliminary experimental assessment shows that the proposed approach is competitive with the state-of-the-art.

6 Conclusions

In this paper, we have studied the combination of several HMM-based generative embeddings with the recently introduced non-extensive information theoretic kernels. We have tested these combinations on SVM-based classification of 2D shapes, with the generative embeddings obtained via HMM modeling of the sequence of curvatures of the shape's contour. Experiments on a benchmark dataset allow concluding that the classifiers thus obtained are competitive with the state-of-the-art methods. Current work includes a more thorough experimental evaluation of the method on other data sets of different nature.

Acknowledgements

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (Contract 213250) and from Fundação para a Ciência e Tecnologia (FCT) (grant PTDC/EEA-TEL/72572/2006).

References

1. Andreu, G., Crespo, A., Valiente, J.: Selecting the toroidal self-organizing feature maps (TSOFM) best organized to object recognition. In: Proc. of IEEE ICNN 1997, vol. 2, pp. 1341–1346 (1997)
2. Bahl, L., Brown, P., de Souza, P., Mercer, R.: Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing, Tokyo, Japan, vol. I, pp. 49–52 (2000)
3. Bicego, M., Cristani, M., Murino, V., Pekalska, E., Duin, R.: Clustering-based construction of hidden Markov models for generative kernels. In: Cremers, D., Boykov, Y., Blake, A., Schmidt, F.R. (eds.) Energy Minimization Methods in Computer Vision and Pattern Recognition. LNCS, vol. 5681, pp. 466–479. Springer, Heidelberg (2009)
4. Bicego, M., Murino, V., Figueiredo, M.: Similarity-based classification of sequences using hidden Markov models. Pattern Recognition 37(12), 2281–2291 (2004)
5. Bicego, M., Pekalska, E., Tax, D., Duin, R.: Component-based discriminative classification for hidden Markov models. Pattern Recognition 42(11), 2637–2648 (2009)
6. Bicego, M., Trudda, A.: 2D shape classification using multifractional Brownian motion. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 906–916. Springer, Heidelberg (2008)
7. Bosch, A., Zisserman, A., Munoz, X.: Scene classification via PLSA. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 517–530. Springer, Heidelberg (2006)
8. Carli, A., Bicego, M., Baldo, S., Murino, V.: Non-linear generative embeddings for kernels on latent variable models. In: Proc. ICCV 2009 Workshop on Subspace Methods (2009)
9. Chen, L., Man, H., Nefian, A.: Face recognition based on multi-class mapping of Fisher scores. Pattern Recognition, 799–811 (2005)

10. Cuturi, M., Fukumizu, K., Vert, J.P.: Semigroup kernels on measures. Journal of Machine Learning Research 6, 1169–1198 (2005)
11. Gales, M.: Discriminative models for speech recognition. In: Information Theory and Applications Workshop (2007)
12. Hein, M., Bousquet, O.: Hilbertian metrics and positive definite kernels on probability measures. In: Ghahramani, Z., Cowell, R. (eds.) Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics, AISTATS (2005)
13. Jaakkola, T., Haussler, D.: Exploiting generative models in discriminative classifiers. In: Advances in Neural Information Processing Systems – NIPS, pp. 487–493 (1999)
14. Jebara, T., Kondor, R., Howard, A.: Probability product kernels. Journal of Machine Learning Research 5, 819–844 (2004)
15. Kaiser, Z., Horvat, B., Kacic, Z.: A novel loss function for the overall risk criterion based discriminative training of HMM models. In: International Conference on Spoken Language Processing, Beijing, China, vol. 2, pp. 887–890 (2000)
16. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: probabilistic models for segmenting and labelling sequence data. In: International Conference on Machine Learning, pp. 591–598 (2001)
17. Martins, A., Smith, N., Xing, E., Aguiar, P., Figueiredo, M.: Nonextensive information theoretic kernels on measures. Journal of Machine Learning Research 10, 935–975 (2009)
18. Mollineda, R., Vidal, E., Casacuberta, F.: Cyclic sequence alignments: Approximate versus optimal techniques. Int. Journal of Pattern Recognition and Artificial Intelligence 16(3), 291–299 (2002)
19. Neuhaus, M., Bunke, H.: Edit distance-based kernel functions for structural pattern classification. Pattern Recognition 39, 1852–1863 (2006)
20. Ng, A., Jordan, M.: On discriminative vs generative classifiers: A comparison of logistic regression and naive Bayes. In: Advances in Neural Information Processing Systems (2002)
21. Perina, A., Cristani, M., Castellani, U., Murino, V.: A new generative feature set based on entropy distance for discriminative classification. In: Proc. Int. Conf. on Image Analysis and Processing, pp. 199–208 (2009)
22. Perina, A., Cristani, M., Castellani, U., Murino, V., Jojic, N.: A hybrid generative/discriminative classification framework based on free-energy terms. In: Proc. Int. Conf. on Computer Vision (2009)
23. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. of IEEE 77(2), 257–286 (1989)
24. Rubinstein, Y., Hastie, T.: Discriminative vs informative learning. In: Knowledge Discovery and Data Mining, pp. 49–53 (1997)
25. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
26. Smith, N., Gales, M.: Speech recognition using SVMs. In: Advances in Neural Information Processing Systems, pp. 1197–1204 (2002)
27. Tsuda, K., Kin, T., Asai, K.: Marginalised kernels for biological sequences. Bioinformatics 18, 268–275 (2002)

Dynamic Linear Combination of Two-Class Classifiers

Carlo Lobrano[1], Roberto Tronci[1,2], Giorgio Giacinto[1], and Fabio Roli[1]

[1] DIEE Dept. of Electrical and Electronic Engineering, University of Cagliari, Italy
[2] Laboratorio Intelligenza d'Ambiente, Sardegna DistrICT, Sardegna Ricerche, Italy
c.lobrano@gmail.com, {roberto.tronci,giacinto,roli}@diee.unica.it

Abstract. In two-class problems, the linear combination of the outputs (scores) of an ensemble of classifiers is widely used to attain high performance. In this paper we investigate some techniques aimed at *dynamically* estimate the coefficients of the linear combination on a pattern per pattern basis. We will show that such a technique allows providing better performance than those of *static* combination techniques, whose parameters are computed beforehand. The coefficients of the linear combination are dynamically computed according to the Wilcoxon-Mann-Whitney statistic. Reported results on a multi-modal biometric dataset show that the proposed dynamic mechanism allows attaining very low error rates when high level of precision are required.

Keywords: Classifier ensembles, two-class classification, biometric systems.

1 Introduction

Many applications, such as anomaly detection, biometric authentication, etc., require the design of classifiers that discriminate one class of objects (a.k.a. the *positive* class) from all other objects (a.k.a. the *negative* class). This is usually attained by measuring how similar the sample is with respect to the *positive* class, and classifying the pattern as *positive* if the similarity score is above some predefined threshold. The performance of this kind of classifiers is evaluated by the trade-off between misclassification errors at some significant *threshold* values, or by resorting to threshold-independent measures such as the Area Under the ROC Curve (AUC) [1]. However, in security applications, the performances are usually evaluated for a limited range of threshold values, corresponding to very low error rates. Unfortunately, the required performance for security applications are hardly met by any individual classifiers.

To improve the overall performance with respect to the selection of the "best" single classifier , the approaches based on ensemble of classifiers are widely used [6,7]. On one hand, the selection of the "best" classifier is not a trivial task, and is highly dependent on the criteria used to rank the classifiers. On the other hand, the use of an ensemble of classifiers allows exploiting the complementary discriminatory information that all the ensemble members may encapsulate. When

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 473–482, 2010.

the classifiers produce similarity scores, the combination is generally performed at the score level by producing a new score according to some "fusion" rule (e.g., by a linear combination) [7,12]. So far, most of the solutions presented in the literature adopts a "static" approach, in the sense that the parameters of the combination rule do not depend on the sample to be classified. It is easy to see that further improvements are expected if the combination rule makes use of sample-specific parameters [16,2,9,10]. For instance, a larger degree of separation between the distributions of *positive* and *negative* samples can be attained by dynamically tuning the combination parameters according to an estimation of the probability that the sample belongs to the *positive* class [13].

In this paper, we propose a novel dynamic combination strategy. For each pattern, and for each classifier, we propose the computation of an index called *Score Decidability Index* (SDI) that is based on the Wilcoxon-Mann-Whitney statistic (WMW). Then, the coefficients of the linear combination are computed as a function of these indexes. This index measures, for each classifier, and for each pattern, the confidence in classifying the pattern either as *positive* or *negative*, according to the score assigned to that pattern, and to the scores assigned to a reference set made up of *positive* and *negative.*

The SDI can also be seen as a different representation of the original score assigned by each classifier to a given pattern, as it represents the likelihood with which the original score is drawn from either the *positive* or *negative* distributions of scores. A new fused score can then be computed by averaging the SDI values. Finally, the SDI values produced by the ensemble of classifiers for a given pattern can be further used to compute the coefficients of a simplified combination rule, where only two of the scores produced by the ensemble are used, namely the maximum and the minimum score.

Section 2 presents the Score Decidability Index (SDI), while its use to compute the coefficients of the dynamic linear combination is presented in Section 3. Section 4 illustrates other uses of the SDI to produce a new transformed score, and the coefficients of simplified dynamic combination. Section 5 shows the experimental results on a multi-modal dataset, where the effectiveness of the proposed techniques are outlined.

2 Score Decidability Index

Usually, the parameters (weights) for a linear combination of outputs produced by an ensemble of two-class classifiers are computed through some estimations or measurements performed on the data. One way to compute these parameters is to exploit one (or more) performance measure or statistic. In this paper we propose the use of a measure called *Score Decidability Index* to estimate the parameters of a dynamic linear combination. This index will be derived from the Wilcoxon-Mann-Whitney (WMW) statistic [4].

Let us consider a two-class problem, where the two classes are denoted as *positive* (ω_+) and *negative* (ω_-). For each pattern x_i to be classified, a two-class classifier C_k usually produce an output *score* $s_{ik} = f_k(x_i)$. Then, a decision threshold *th* is set, and patterns whose score is greater than the threshold are assigned to the *positive* class, otherwise they are assigned to the *negative* class.

Let now us consider a set of patterns whose class is known for a generic classifier C_k, and let:

$$\begin{array}{l} S_k^+ = \left\{ s_{ik}^+ = f_k\left(x_i\right) \mid x_i \in \omega_+ \right\} \ , \ \forall i \\ S_k^- = \left\{ s_{ik}^- = f_k\left(x_i\right) \mid x_i \in \omega_- \right\} \ , \ \forall i \end{array}$$

The performance of two-class classifiers C_k for all possible values of the decision threshold *th* can be summarized by the AUC, whose value can be computed by resorting to the WMW statistic [5]:

$$AUC_k = \frac{\sum_{i=0}^{n_+} \sum_{j=0}^{n_-} I(s_{ik}^+, s_{jk}^-)}{n_+ \cdot n_-} \tag{1}$$

where n_+ and n_- represent the number of *positive* and *negative* patterns, and the function $I(a, b)$ is equal to 1 if $a > b$, otherwise it is equal to 0. This formulation of the AUC can be also seen as a measure of the probability that the classifier ranks a randomly chosen *positive* sample higher than a randomly chosen *negative* sample, i.e. $P(S^+ > S^-)$ [4].

Let us define

$$r_-(s) = \frac{\sum_{i=0}^{n_+} I(s_i^+, s)}{n_+} \simeq P(S^+ > s) \tag{2}$$

$$r_+(s) = \frac{\sum_{j=0}^{n_-} I(s, s_j^-)}{n_-} \simeq P(s > S^-) \tag{3}$$

Hence $r_-(s)$ represents the probability that the score s is lesser than a score coming from the *positive* distribution, and $r_+(s)$ represents the probability that the score s is larger than a score coming from the *negative* distribution.

It can be easily seen that the WMW statistic in Eq.(1) can be written in either of the two following formulations:

$$\frac{\sum_{i=0}^{n_+} \sum_{j=0}^{n_-} I(s_{ik}^+, s_{jk}^-)}{n_+ \cdot n_-} \rightarrow \left\{ \begin{array}{l} \dfrac{\sum_{j=0}^{n_-} r_-(s_{jk}^-)}{n_-} \\ \dfrac{\sum_{i=0}^{n_+} r_+(s_{ik}^+)}{n_+} \end{array} \right. \tag{4}$$

Thus, $r_+(s)$ and $r_-(s)$ represent an estimation of the contribution of the score s to the value of the AUC in the case it belongs either to the *positive* or *negative* class, respectively. Given the sets of scores S_k^+ and S_k^- produced by a two-class classifier C_k on a training set, for each score s_{ik} related to a test pattern x_i, the *Score Decidability Index* (SDI) can be defined as

$$\Delta(s_{ik}) = r_+(s_{ik}) - r_-(s_{ik}) \tag{5}$$

that is related to the likelihood the pattern x_i is drawn either from the positive or negative distributions of scores. If $\Delta(s_{ik}) = 1$ (i.e., $P(s_{ik} > S_k^-) = 1$ and

$P(S_k^+ > s_{ik}) = 0)$, the score s_{ik} is larger than any other score in the training set $S_k^+ \bigcup S_k^-$. As a consequence, it is more likely that pattern x_i comes from the *positive* rather than from the *negative* distribution. Similarly, if $\Delta(s_{ik}) = -1$ (i.e., $P(s_{ik} > S_k^-) = 0$ and $P(S_k^+ > s_{ik}) = 1$), the score s_{ik} is lesser than any other score in the training set, so that it is more likely that the sample comes from the *negative* distribution. The other values of $\Delta(s_{ik})$ in the range $[-1, 1]$ accounts for the uncertainty in the classification of the sample whose score is s_{ik}, the larger the value of $|\Delta(s_{ik})|$, the more confident is the class decision.

3 Dynamic Linear Combination

As stated in the Introduction, the linear combination of scores is one of the most widely used way to fuse outputs from different classifiers.

$$s_i^{lc} = \sum_{k=1}^{\mathbf{N}} \alpha_k \cdot s_{ik} \tag{6}$$

Usually, some constraints are introduced to simplify the parameters estimation. For example, affine combinations are obtained if $\sum_k \alpha_k = 1$, conical combinations are those combinations for which $\alpha_k \geq 0$, and convex or weighted combinations require that $\alpha_k \geq 0$, and $\sum_k \alpha_k = 1$.

One of the simplest form of linear combination is obtained by averaging the outputs of the classifiers (a.k.a. the *Mean*-rule). This rule implicitly assume that all the classifiers are assigned the same weight [14]. However, it has been pointed out that a weighted combination outperforms the *Mean*-rule when the classification problem is made up of imbalanced classes [3]. The weights of the combination are usually computed by maximizing a measure of performance on a training set. It is worth noting that usually each classifier is assigned a unique weight that does not depend on the sample to be classified. In other words, typically the weighted combination aims at improving the *average* performance of the classification system. Moreover, the optimization algorithm may exhibit a high computational cost, depending on the heuristic used to maximize the selected performance measure or statistic [8,15]. On the other hand, it is easy to see that an optimal linear combination rule should require weights that depends both on each individual classifier, and on the pattern to be classified.

$$s_i^* = \sum_{k=1}^{\mathbf{N}} \alpha_{ik} \cdot s_{ik} \tag{7}$$

Such techniques are usually called "dynamic" combination techniques. However, if it is difficult and computational costly to estimate the optimal set of weights for each individual classifier, the estimation of dynamic weights may result in a more complex problem [7]. Usually this problem is solved by estimating the behavior of each classifier in the region containing the test sample. Different heuristics have been proposed that are based on different definitions of *classifier behavior*, and different definition of *regions*.

In this paper, we propose to exploit the SDI formulated in Section 2 to derive the weights of the combination so that the distributions of the combined output for the two classes allows effective separation. One way to achieve this goal is to provide large weights for each score related to *positive* samples, so that the combined score is as high as possible, and to provide small weights for each score related to *negative* samples, so that the combined score is as small as possible. Actually, each classifier in the ensemble can provide the information on the most likely class a test pattern belongs to. If a set of reference patterns related to the positive and negative classes are available, the distribution of the outputs on such a set can be representative of the behavior of that classifier for the two classes. Thus, if we compare the output s produced by each classifier in the ensemble with the outputs of the same classifier on the reference set, that classifier supports the following conclusions

$$\begin{array}{l}\text{if } s > s_{ik},\ \forall s_{ik} \in S_k^-\text{, then the pattern is likely to be } \textit{positive}\\ \text{if } s < s_{ik},\ \forall s_{ik} \in S_k^+\text{, then the pattern is likely to be } \textit{negative}\end{array}$$

For any other intermediate case, the classifier may support one decision or the other with different strength, depending on the fractions of the reference set which support the two above propositions.

Actually the value of SDI can be used to compute the weights of the linear combination, as for each classifier and for each score it can provide the information on the most likely class, and the *strength* of the decision. It can be easily seen that the sign of $\varDelta$ indicates the most likely class, while the modulus of $|\varDelta|$ is a measure of the "strength" of the class prediction. Thus we propose to use the SDI to compute the weights of a dynamic linear combination as:

$$\alpha_{ik} = \frac{\varDelta(s_{ik}) + 1}{2} \tag{8}$$

where the value of α_{ik} is in the range $[0, 1]$ in agreement with the normalization used for the outputs of the classifier. We will refer to this technique as DLC.

4 Other Dynamic Rules Based on Score Decidability Index

The rationale behind the computation of the weights for the linear combination shown in the previous section may give rise to other combination rules.

4.1 The Score Decidability Index as a *normalized* Score

In the previous section we claimed that for each pattern and for each classifier, the sign of $\varDelta$ indicates the most likely class, while the modulus of $|\varDelta|$ is a measure of the "strength" of the class prediction. If we normalize $\varDelta$ in the range $[0, 1]$, the resulting value can be used as a new *normalized* score for each classifier. Then, these new values can be combined by any combination mechanism. In order to keep the system simple, and in account of the meaning of these new normalized scores, we propose to average these new values. We will refer to this technique as SDI mean:

$$s_i^* = \frac{1}{N}\sum_{k=1}^{N}\frac{\Delta_{ik}+1}{2} \tag{9}$$

4.2 Simplified Score Combination

The Score Decidability Index can be also used in the framework of a simplified combination scheme called Dynamic Score Combination (DSC) [13]. Two similar formulations of the DSC have been proposed:

$$s_i^* = \beta_{1i}\cdot\max_k(s_{ik}) + \beta_{2i}\cdot\min_k(s_{ik}) \tag{10}$$

$$s_i^* = \beta_i\cdot\max_k(s_{ik}) + (1-\beta_i)\cdot\min_k(s_{ik}) \tag{11}$$

In Eq.(10) the two parameters β_{1i}, and β_{2i} "estimate" the likelihood of x_i being a *positive* or a *negative* pattern. Eq.(11) is similar to Eq.(10), where the constraints $\beta_{2i} = (1-\beta_{1i})$ and $\beta_i \in [0,1]$ are added. DSC basically combines only two values among all the scores produced by the ensemble of classifiers, namely the smallest and the biggest values. On the other hand, the behavior of the ensemble (i.e., all the scores produced by the ensemble) is used to compute the values of the βs. In the following we propose two different methods to embed the SDI into Eq.s(10), and (11).

Dynamic Score Combination by Δ voting. Let us consider the formulation of DSC in Eq.(10) where the values of β are continuous. By taking into account that the decidability of the class of the sample is critical if the value of Δ is close to zero, we can fuse the SDI of the ensemble of classifiers by a Voting mechanism. In particular, we evaluate the "likelihood" of the sample belonging either to the *positive* or the *negative* class, by counting the fraction of the classifiers that exhibit a decidability index larger than an offset α:

$$\beta_{1i} \;= \frac{1}{N}\textstyle\sum_{k=1}^{N} I(\Delta(s_{ik}),\alpha) \tag{12}$$

$$\beta_{2i} = \frac{1}{N}\textstyle\sum_{k=1}^{N} I(-\Delta(s_{ik}),\alpha) \tag{13}$$

Typical values of α are 0.05, 0.1, 0.15, and 0.2. The reported experimental results are related to $\alpha = 0.05$.

Dynamic Score Combination by Δ mean. In this case, we take into account the formulation of the DSC reported in Eq.(11). In this case, the values of Δ can be used to compute the parameter β_i by taking into account the average and the standard deviation of Δ among all the classifiers as follows:

$$\Delta^*(\mathbf{s}_i) = \frac{\frac{1}{N}\sum_{k=1}^{N}\Delta(s_{ik})}{\sigma_{\Delta(s_{ik})}} \tag{14}$$

$$\beta_i \;= \frac{1}{1+e^{-\gamma\cdot\Delta^*(\mathbf{s}_i)}} \tag{15}$$

where the sigmoid in the Eq (14) is used to normalize the value of $\varDelta^*(\mathbf{s}_i)$ in the range $[0, 1]$. Typical values of γ in the normalization process are from 1 to 6. The reported experimental results are related to $\gamma = 3$.

5 Experimental Results

The experiments have been performed on the *Biometric Authentication Fusion Benchmark Database* (BA-Fusion), a multi-modal database of similarity scores artificially created from experiments carried out on the XM2VTS face and speaker verification database [11]. This dataset contains similarity scores from 8 classifiers, and the scores have been normalized by the *Tanh* rule [12].

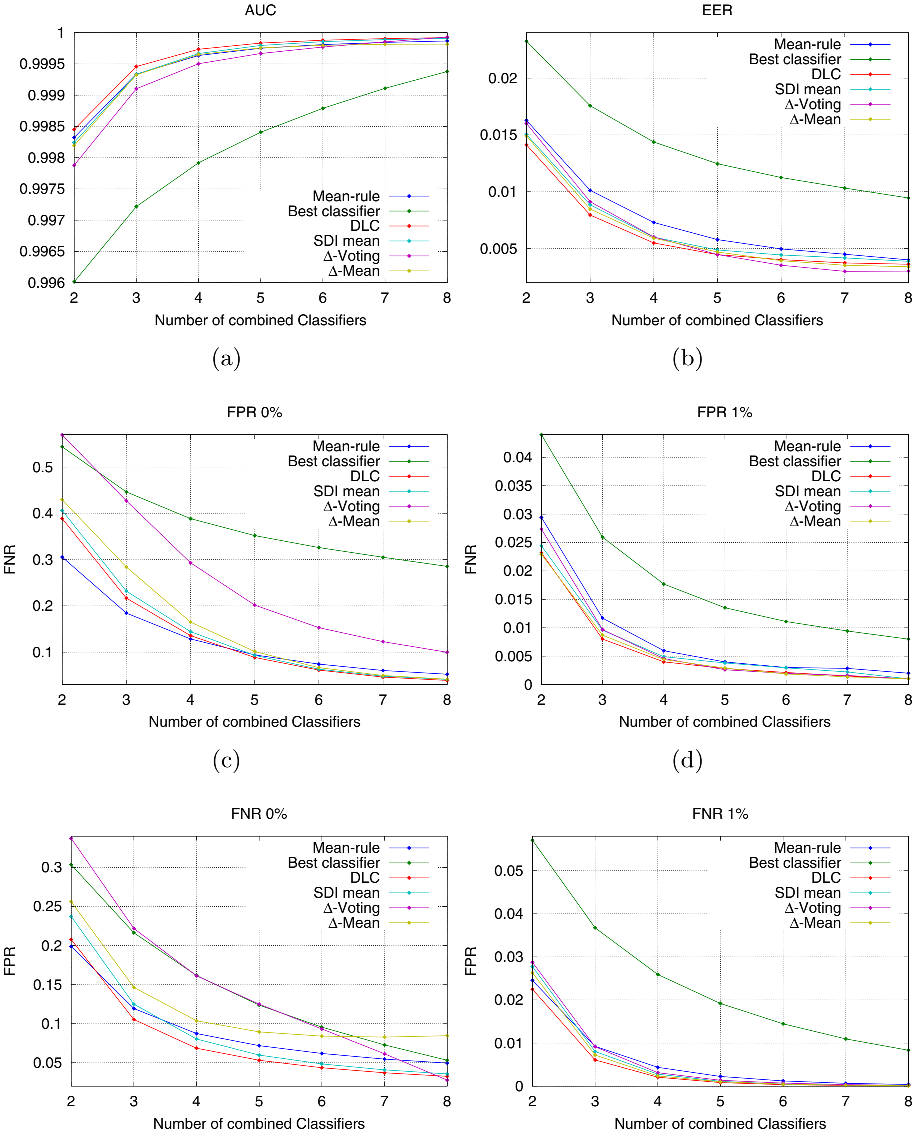
Reported experiments aim at assessing the performance of the proposed techniques in terms of different performance measures. In particular, the AUC, the EER, have been used, as well as error measures at four operating points that are generally used to test security systems, namely FPR 1%, FPR 0%, FNR 1% and FNR 0%. Thus, the FNR (FPR) attained when the FPR (FNR) is equal to 1% or 0% are measured, respectively.

Experiments have been carried out by creating ensembles where the number of classifier in the ensemble ranges from 2 to 8. In this way, we create ensembles that contain all possible subsets of classifiers from the original pool of 8 classifiers. In order to get unbiased results, a 4-fold cross-validation technique has been used. The dataset has been subdivided into 4 subsets, so that one subset at a time was used for training, while the other three have been used for testing. Results are reported in terms of average and standard deviation over the four trials, and over all the possible ensemble of classifiers for a given ensemble size.

The performance of the proposed algorithms have been compared to those of the *Mean-rule*, as this is a simple and effective way of combining multiple scores. Very often experimental results show that the *Mean-rule* provides significant performance improvements not only with respect to individual classifiers, but also with respect to other combination rules. Performance are also compared to the best performance provided by the individual classifiers included in the ensemble. It is worth noting that for each measure of performance, the best value can be related to a different classifier in the ensemble.

Results reported in Fig.1 show that the average performance improve as the size of the ensemble increases. This results shows that the proposed combination mechanisms allow exploiting the complementary information that the individual classifiers may exhibit. In particular, the combination of classifiers always allows outperforming the best classifier, and provide very low error rates. By inspecting the figure, an ensemble size equal to five can be a good compromise between performance and ensemble complexity. For this reason, Table 1 shows the detailed numerical results in terms of the average and standard deviation for an ensemble size equal to five.

Fig. 1(a) shows the results in terms of the AUC. It is easy to see that all the combination methods provide very high AUC values, very close to each other. Fig.s 1(b)-(d) show the performance in terms of EER and FPR 1%, respectively.



(a) (b)

(c) (d)

(e) (f)

Fig. 1. Average performance for each ensemble size

Table 1. Performance in terms of average and standard deviation (between brackets) for all the ensembles of 5 classifiers. Results with a ○ indicate that the difference in performance from those achieved by the Mean-rule are not statistically significant according to the t-test with a 95% confidence. The best performance are in italics.

	AUC	**EER**
Mean-rule	*0.9998(±0.0002)*	0.0058(±0.0019)
Best classifier	0.9984(±0.0014)	0.0125(±0.0046)
DLC	*0.9998(±0.0002)*	*0.0045(±0.0017)*
SDI mean	*0.9998(±0.0002)*	0.0049(±0.0023)
$\varDelta$ Voting	0.9997(±0.0005)	0.0045(±0.0021)
$\varDelta$ mean	○ 0.9998(±0.0003)	0.0047(±0.0019)

	FPR-0%	**FPR-1%**	**FNR-1%**	**FNR-0%**
Mean-rule	0.0941(±0.0342)	0.0040(±0.0026)	0.0023(±0.0017)	0.0719(±0.0827)
Best classifier	0.3518(±0.1148)	0.0135(±0.0092)	0.0192(±0.0181)	0.1237(±0.1120)
DLC	*0.0886(±0.0469)*	0.0028(±0.0024)	*0.0008(±0.0010)*	*0.0532(±0.0599)*
SDI mean	○ 0.0931(±0.0455)	○ 0.0038(±0.0029)	0.0011(±0.0020)	0.0598(±0.0619)
$\varDelta$ Voting	0.2017(±0.1622)	*0.0026(±0.0024)*	0.0014(±0.0013)	0.1250(±0.1981)
$\varDelta$ mean	○ 0.1015(±0.0723)	0.0029(±0.0025)	0.0010(±0.0013)	0.0895(±0.1150)

Regardless the ensemble size, all the proposed methods outperform those of the mean rule. However, when the EER is considered, the DLC outperform all other measures for ensemble sizes smaller than or equal to five, while $\varDelta$-voting provides the best performance for sizes greater than five. On the other hand, when the FPR 1% is considered, the DLC provides the best performance for small ensemble sizes, while differences among the proposed mechanisms tends to be negligible as the ensemble size is greater than 5. A similar behavior can be also seen in Fig. 1(f) where the performance for FNR 1% are shown. A different behavior can be seen in Fig.s 1(c)-(e), where the working point is set to 0% FPR or FNR respectively. In these cases, $\varDelta$-voting provides the worst performance, while the DLC and SDI-mean outperform the *Mean-rule* for any ensemble size in the case of FNR 0%, while in the case of FPR 0% performance improvements are shown for ensemble sizes greater than or equal to 5. Thus, we can conclude that the proposed mechanisms allows exploiting the complementarity of different classifiers, especially in the case of large ensemble size.

In particular, in the case of the dataset at hand, we observed that the DLC and SDI-mean outperform all other techniques in any performance measure for ensembles size greater than or equal to 5. The inspection of the values in the Table 1 clearly shows that the AUC does not allow to see any significant difference among the considered combination mechanisms. On the other hand, the values related to the operating point related to very low error rates, show the effectiveness of the proposed mechanism. This effectiveness has been also validated by performing the t-test with a 95% confidence on the difference in performance with the *Mean-rule*. All the differences, except those marked with a circle, are statistically significant. In addition, it is worth noting that in security applications even small differences in performances are of great value.

The reported results allow to conclude that the proposed DLC and SDI-mean techniques based on the *Score Decidability Index* allows exploiting effectively the complementarity among different classifiers. In addition, depending on the performance measure of interest, the other two techniques based on a simplified combination can also provide good performances. In conclusion, it can be pointed out that the proposed Index provide an useful measure for the estimation of the parameters for combining an ensemble of two-class classifiers.

References

1. Fawcett, T.: An introduction to ROC Analysis. Pattern Recognition Letters 27(8), 861–874 (2006)
2. Fiérrez-Aguilar, J., Chen, Y., Ortega-Garcia, J., Jain, A.K.: Incorporating Image Quality in Multi-algorithm Fingerprint Verification. In: Zhang, D., Jain, A.K. (eds.) ICB 2005. LNCS, vol. 3832, pp. 213–220. Springer, Heidelberg (2005)
3. Fumera, G., Roli, F.: A theoretical and experimental analysis of linear combiners for multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 942–956 (2005)
4. Hanley, J.A., McNeil, B.J.: The meaning and the use of the area under a receiver operanting charateristic curve. Radiology 143, 29–36 (1982)
5. Huang, J., Ling, C.: Using AUC and Accuracy in Evaluating Learning Algorithms. IEEE Transactions on Knowledge and Data Engineering 17, 299–310 (2005)
6. Kittler, J., Hatef, M., Duin, R., Matas, J.: On combing classifiers. IEEE Trans. on PAMI 20(3), 226–239 (1998)
7. Kuncheva, L.: Combining Pattern Classifiers: Methods and Algorithms. John Wiley & Sons Inc., Chichester (2004)
8. Marcialis, G.L., Roli, F.: Fusion of multiple fingerprint matchers by single layer perceptron with class-separation loss function. Pattern Recognition Letters 26, 1830–1839 (2005)
9. Nandakumar, K., Jain, A., Dass, S.: Quality-based Score Level Fusion in Multibiometric Systems. In: ICPR 2006, pp. 473–476 (2006)
10. Poh, N., Bengio, S.: Improving Fusion with Margin-Derived Confidence In Biometric Authentication Tasks. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 474–483. Springer, Heidelberg (2005)
11. Poh, N., Bengio, S.: Database, protocols and tools for evaluating score-level fusion algorithms in biometric authentication. Pattern Recognition 39(2), 223–233 (2006)
12. Ross, A.A., Nandakumar, K., Jain, A.K.: Handbook of multibiometrics. Springer, Heidelberg (2006)
13. Tronci, R., Giacinto, G., Roli, F.: Dynamic Score Combination: A Supervised and Unsupervised Score Combination. In: Perner, P. (ed.) MLDM 2009. LNCS (LNAI), vol. 5632, pp. 163–177. Springer, Heidelberg (2009)
14. Tumer, K., Gosh, J.: Linear and order statistics combiners for pattern classification. In: Combining Artificial Neural Nets, pp. 127–162. Springer, Heidelberg (1999)
15. Marrocco, C., Duin, R.P.W., Tortorella, F.: Maximizing the area under the ROC curve by pairwise feature combination. Pattern Recognition 41(6), 1961–1974 (2008)
16. Bigun, E., Bigun, J., Duc, B., Fischer, S.: Expert conciliation for multi modal person authentication systems by Bayesian statistics. In: Bigün, J., Borgefors, G., Chollet, G. (eds.) AVBPA 1997. LNCS, vol. 1206, pp. 291–300. Springer, Heidelberg (1997)

Large-Scale Text to Image Retrieval Using a Bayesian K-Neighborhood Model[*]

Roberto Paredes

ITI-UPV
Camino de Vera S/N, 46022 Valencia (Spain)

Abstract. In this paper we introduce a new approach aimed at solving the problem of image retrieval from text queries. We propose to estimate the word relevance of an image using a neighborhood-based estimator. This estimation is obtained by counting the number of word-relevant images among the K-neighborhood of the image. To this end a Bayesian approach is adopted to define such a neighborhood. The local estimations of all the words that form a query are naively combined in order to score the images according to that query. The experiments show that the results are better and faster than the state-of-the-art techniques. A special consideration is done for the computational behaviour and scalability of the proposed approach.

1 Introduction

This paper addresses the problem of image retrieval from text queries. This problem is commonly treated and it is a fundamental part of web search engines and photographic databases. Image retrieval from text is a particular example of an information retrieval system where the user uses text queries in order to search for the requested information. Therefore the methodology proposed here could be easily applied to other scenarios such as video and audio retrieval. The precision of the image retrieval systems has been improved during the last years due to the introduction of new image descriptors and methodologies. In the case of web search engines the current image retrieval technology is mainly based on the text that appears around the images in the web pages. On the other hand, in online photographic databases like Flicker or Picasa, the textual information related to the images is extracted from the tags with which the user described the pictures during the uploading process. Despite of the straightforward implementation and relatively good results of such an approach, it can not be extended to other scenarios where that textual information is not available. The problem of image retrieval from text queries is usually solved by means of ranking the images according to their relevance to the query meaning. The images are sorted with regard to the *scores* that they obtain for a particular query and the images with the highest scores are presented to the user. Therefore the retrieval problem is reduced to the computation of the scores for any pair query-image.

[*] Work supported by the UPV project PAID09015 and Spanish projects: TIN2008-04571 and Consolider Ingenio 2010 MIPRCV (CSD2007- 00018).

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 483–492, 2010.

Recently different approaches for solving this problem using a neighborhood model have been proposed [13,9]. In [13] the authors proposed two different ways of linearly combining different distances that define the image neighborhood. In [9] the authors propose two different approaches, weighted nearest neighbor for tag prediction and word-specific logistic discriminant models. These methods show the capabilities of the neighborhood-based estimators to solve the image annotation problem, but these methods scale poorly and can not be applied to large scale problems with a huge number of images (millions) and large size of the vocabulary (thousands).

For large scale problems a linear discriminative model (PAMIR) has been proposed in order to rank images from text queries [8]. This method outperforms other techniques like Cross-Media Relevance Model (CMRM) [11], Cross-Media Translation Table (CMTT) [17], Probabilistic Latent, Semantic Analysis (PLSA) [15] and Support Vector Machines (SVM) [16],[19]. This approach has demonstrated good performance and a good scalability behaviour following an online learning approach. This method can be considered the state-of-the-art for *large scale* image retrieval from text queries. The current paper proposes a new approach based on a *local* word relevance estimation. This local estimation is accomplished by considering the K-neighborhood of the images but the present work aims at guaranteeing the scalability capabilities.

The paper is organized as follows. The new approach is presented in section 2. Computational issues are considered in section 3. Experiments with two different datasets are carried out in section 4. Finally, some conclusions are drawn in section 5.

2 Approach

In this section the new approach based on a K-neighborhood word relevance estimation is presented. The new method will be denoted as KNIR (K-Neighborhood Image Retrieval).

In order to retrieve images given a text query, a *score* for any image given this text query is needed. This score should be high when the image content is relevant to the text query and should be low when the image content is not relevant to the text query.

Given an image **p** and a text query q represented by a bag of words $q = \{w_1, w_2, \dots, w_n\}$. We propose the following score

$$sc(q, \mathbf{p}) = p(q \mid \mathbf{p}) \tag{1}$$

A linearly smoothed naive Bayes decomposition of $p(q \mid \mathbf{p})$ yields:

$$p(q \mid \mathbf{p}) = \prod_{i=1}^{n} p(w_i \mid \mathbf{p}) = \prod_{i=1}^{n} \left(\beta\, \hat{p}(w_i \mid \mathbf{p}) + (1-\beta) \frac{1}{\mid d \mid} \right) \tag{2}$$

where d is the size of the text vocabulary.

The expression to estimate is $\hat{p}(w_i \mid \mathbf{p})$, an estimation of the conditional probability of the word w_i given the image **p**. To estimate this conditional distribution we define the set P_i^+. This set is the set of pictures that are relevant to queries where the word w_i appears.

The following K-neighborhood estimator is proposed:

$$\hat{p}(w_i \mid \mathbf{p}) = \frac{C_{Ki}}{K} \tag{3}$$

where C_{Ki} is the number of pictures that belong to P_i^+ among the K-nearest of **p**.

2.1 Parameter Selection

To compute the proposed score we have to define two parameters K and β. In the experiments the parameter β was found not to be critical. On the other hand, the parameter K has an important impact since this parameter defines the image neighborhood considered. Here we are going to focus on this parameter and how to estimate it adequately. Instead of trying different values of this parameter and selecting the best one using a validation set, a Bayesian approach is proposed:

$$p(w_i \mid \mathbf{p}) = \sum_{\forall K} p(w_i \mid \mathbf{p}, K) p(K \mid \mathbf{p}) \tag{4}$$

The term $p(w_i \mid \mathbf{p}, K)$ is what we are estimating in equation 3 for a particular value of K. Therefore applying this Bayesian approach to our problem and limiting the values of K to some maximum value K_{max}, equation 4 can be expressed as:

$$\hat{p}(w_i \mid \mathbf{p}) \approx \sum_{K=1}^{K_{max}} \frac{C_{Ki}}{K} p(K \mid \mathbf{p}) \tag{5}$$

Here, usually, a Markov chain Monte Carlo (MCMC) procedure is used in order to draw parameters from the $p(K \mid \mathbf{p})$ distribution, selecting those parameters with the highest likelihood, see for instance [6] and [14]. In this work a more simple, yet effective approach is proposed by assuming an uniform distribution of the parameter space: $p(K \mid \mathbf{p}) \sim U(1, K_{max})$. Therefore equation 5 becomes:

$$\hat{p}(w_i \mid \mathbf{p}) \approx \frac{1}{K_{max}} \sum_{K=1}^{K_{max}} \frac{C_{Ki}}{K} \tag{6}$$

and finally equation 2 can be rewritten as:

$$p(w_i \mid \mathbf{p}) \approx \beta \frac{1}{K_{max}} \sum_{K=1}^{K_{max}} \frac{C_{Ki}}{K} + (1-\beta) \frac{1}{\mid d \mid} \tag{7}$$

The parameter K_{max} is indeed a parameter to be tuned but, as the experiments will show, this parameter leads to better results than the parameter of the *conventional* K-neighborhood in a wide range of values. It is important to note that the Bayesian approach does not entail more computations than the standard one. The Bayesian approach only requires to define the K_{max}-neighborhood. Only one search is needed, and then the *votes* of each image among the K_{max}-nearest are adequately weighted depending on the rank obtained. That is, nearest images have higher weight while farther images have lower weight.

The Bayesian approach and the naive Bayes decomposition of the query lead to the following expression of the required score:

$$sc(q, \mathbf{p}) = \prod_{i=1}^{n} \beta \frac{1}{K_{max}} \sum_{K=1}^{K_{max}} \frac{C_{Ki}}{K} + (1-\beta) \frac{1}{\mid d \mid} \tag{8}$$

3 Computational Issues

In this section we discuss the computational issue to be considered in order to achieve a good computational behaviour and scalability properties.

First of all it is important to distinguish between *training* and *test* images. The difference between training and test images is that for the training images we know the relevance judgment for a set of training queries while for the test images we do not have such judgment. This is an important issue to take into account because in a practice scenario the number of images whose description is known uses to be very reduced in comparison with the total number of images available. However we aim at performing the image retrieval process over *all* the images instead of over the annotated set only. Moreover in order to evaluate the generalization capabilities of the approaches, training images used to be discarded and the image retrieval used to be performed over non-annotated images. Therefore the training images are used to define the sets P_i^+ while for the test images, **p**, we have to estimate the conditional probability $p(w_i \mid \mathbf{p})$ for any word of the vocabulary. This conditional probability is estimated in training time and it is stored in a table for each pair $(w_i, \mathbf{p})$. The computation of the score $sc(q, \mathbf{p})$ is reduced to the product of the values that appear in this table in the positions corresponding to the words that belong to that query and that particular image. This process is performed in test time for a given query.

An efficient implementation of the computation of the term $p(w_i \mid \mathbf{p})$ is required in order to achieve a fast training time and scalability. Two important components affect this computational behaviour. First, to obtain the set P_i^+ for every word w_i of the vocabulary. Second, to compute the K-neighborhood of any image **p**. Clearly the most costly part is the computation of the neighborhood where the distances between **p** and all the training images must be obtained. To this end fast search algorithms can be used in order to alleviate such task. Although different techniques have been tested, here only the best one are described. Furthermore, since only vectorial representation of the images have been considered, only vectorial approaches have been tested. Two different approaches have been finally used depending on the dataset, Visual Word Hashing (VWH) and Local Sensitive Hashing (LSH)[7]. The approach that uses fast search algorithm will be denote as FKNIR (Fast KNIR).

Visual Word Hashing

The first dataset used in the experiments is the same dataset that was used in [8]. In this dataset the image representation is very sparse and the well known LSH does not provide the best computational performance. Due to the sparsity of the image representation VWH has been proposed. VWH is a very simple method, the idea is to store each image in several buckets. The maximum number of buckets is D^2 being D the number of visual words of the image representation, so $\mathbf{p} \in \Re^D$. Therefore each bucket stores the indexes of the images with a particular pair of visual words. The buckets usually form a sparse matrix and standard hashing is used to deal with such data structure. Higher order visual word correlations could be used but the second order leads to good enough results.

Local Sensitive Hashing

The image representation used on the second experiments does not lead to an important sparsity of the data. In this case the well known LSH [7] lead to the best performance. To this end r 2-stable-random projections of the image $\mathbf{p}$ are computed [10]. The projection value then is binarized depending on its sign. These r binary numbers are randomly selected forming w different words of b bits, clearly $b < r$ must be fulfilled, $b \ll r$ is advisable. Finally the index of the image $\mathbf{p}$ is stored in the buckets associated to the w words. So it is expected that this method split the image representation space into 2^b buckets but each image can appear in up to w different buckets.

In both methods the search of the K-neighbors for a test image is performed computing the original distance (usually L_p family) between the image and all the images that fall into the same buckets.

4 Experiments

The experiments have been carried out with two different datasets, Corel and ImageClef. Both data sets are split into *development* and *test* set. The development set is further split into *training* and *evaluation*. Each partition contains pictures, text queries and the relevance judgment for any pair picture-query. The evaluation set is used for tuning the model parameters and the test set is finally used to evaluate the different models. This test set evaluation is performed by means of two measures: average precision (AvgP) and precision at top 10 (P10) both in percentage. The training time of the different methods is also measured.

In general in all the experiments the validation set was used to tune the parameters of the different methods. The parameter C of PAMIR was varied from 0.001 to 1.0 and the number of iterations from 10^5 to 10^8. The parameter K_{max} of the KNIR method was varied from 10 to 1000.

The results obtained with our implementation of PAMIR are almost identical to the results reported in [8].

4.1 Corel Dataset

These experiments were carried out using the same dataset used in [8]. In fact preprocesed version of this dataset was provided by the authors of [8] so an exact comparison can be made. This data is composed by two different partitions, Corel-small and Corel-large. Both sets originate from the Corel stock photography collection, which offers a large variety of pictures, ranging from wilderness scenes to architectural building pictures or sport photographs.

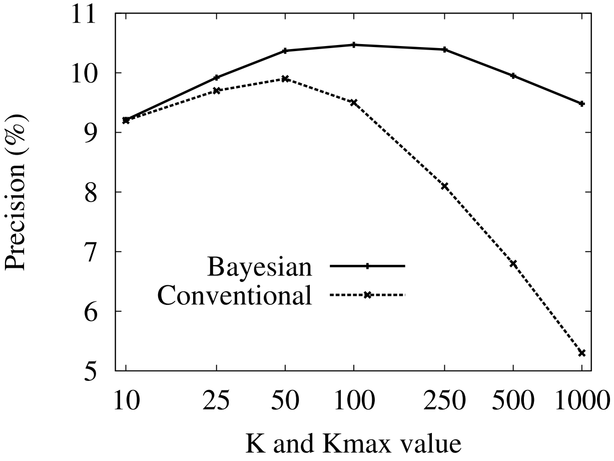


Fig. 1. Comparison between conventional K- neighborhood and Bayesian K-Neighborhood performance on the Corel-small validation dataset

One common issue in both partitions is that the images are represented using a very high dimensional vectorial representation; namely, 10,000 components that come from the concatenation of two different sets of image features: Local Binary Patterns and Color histograms. This high dimensional representation is not a casual selection but this high dimensional representation is somewhat required by the PAMIR approach. This approach relies on the *linear* separation of the image representation space for a given word. That is, the image representation space should be linearly split into the relevant and not-relevant images for a given word. This linear separation is not fully accomplished when the image representation selected has not such very high dimensionality. In this sense, PAMIR requires that the practitioners use such high dimensionality representation reducing in some situations the practitioners choices.

Corel-small dataset. Corel-small corresponds to the 5,000-picture set presented in [4]. This set, along with the provided split between development and test data, has been used extensively in the query-by-text literature, e.g. [1], [12],[15]. The development set is composed by 4,500 pictures that are further split into 4,000 pictures for training and 500 pictures for evaluation. The test set is composed by 500 pictures. The number of queries are 7,221, 1,962 and 2,241 for training, validation and test respectively.

Figure 1 shows the precision of the retrieval system comparing the Bayesian and the conventional neighborhood for different values of K_{max} and K respectively. The Bayesian approach shows a better behaviour and the selection of parameter K_{max} is less critical than the selection of parameter K for the conventional approach.

Table 1 shows the results obtained for the small dataset. The precision, average precision and training time are compared for PAMIR, KNIR and FKNIR. As commented before, FKNIR uses a VWH method, similar but slower results were obtained using LSH. It is important to note that the sparsity of the image representation is a very

Table 1. Results on Corel-small test set

Method	P@10(%)	AvgP (%)	Training (ms)
PAMIR	9.97	25.8	2937
KNIR	9.9	26.7	524
FKNIR	10.0	27.1	1571

Table 2. Results on Corel-large test dataset

Method	P@10(%)	AvgP (%)	Training (secs)
PAMIR	2.69	4.98	63.7
KNIR	2.73	4.78	26.9
FKNIR	2.84	4.92	18.2

important factor in the computational cost of the algorithms. Thus, depending on the sparsity of the dataset the selection of the fast search algorithm could be reconsider.

The results obtained for the KNIR methods and PAMIR are almost identical. Only the average precision results of KNIR methods show some small improvements. The computation time of KNIR is clearly better than the PAMIR. In these experiments FKNIR has an important overhead, to obtain the buckets of the training images, and the time results showed in table 1 are not better than the standard KNIR, but a slight average precision improvement is obtained.

Corel-large dataset. Corel-large was proposed in [8] and contains 35,379 images and corresponds to a more challenging retrieval problem than Corel-small. This dataset is split into development and test partitions. The development set is composed by 25,120 pictures that are further split into 14,861 training pictures and 10,259 evaluation pictures. The test set is composed of 10,259 pictures. The number of queries are 55,442, 39,690 and 39,613 for training, validation and test respectively.

Table 2 shows the results obtained for the large dataset. The precision, average precision and training time are compared for PAMIR, KNIR and FKNIR.

The results obtained for the KNIR methods and PAMIR are again almost identical. The computation time of KNIR is clearly better than the PAMIR while the FKNIR is now faster than KNIR.

These two experiments do not show a significant improvement on the precision of the here proposed approaches. On the other hand, the faster computation has been demonstrated. It seems that the Corel dataset, more concretely the image representation used in [8] has some properties that fulfill the PAMIR restrictions, that is, to be able to define an hyperplane to split the image representation space into two different regions, relevant an no relevant for a given word. Then, the K-neighborhood model proposed here does not have anything to add and any further improvement can be achieved in this sense.

The following experiments are conducted to show that in some circumstances the linear separation of the image representation space is not completely fulfilled and the here proposed approach can provide improvements.

4.2 ImageCLEF Dataset

This experiment was conducted using the ImageClef photo annotation dataset [5]. The total of 20,000 images were split into 18,000 training images, 1,000 validation images and 1,000 test images. A small vocabulary of 124 words was extracted from the image annotations. A total number of 10,647, 1,572 and 1,610 queries were generated for the training, validation and test sets respectively. The validation set was used to adjust the parameters as in the previous experiments.

The images were represented using two different features: Local image descriptors and color histograms. In this work the local image descriptors are patches that are extracted from the images at regular grid positions and dimensionality reduced using PCA transformation [3]. These local descriptors are finally represented using an histogram of visual words [2],[18]. On the other hand, color histograms are among the most basic approaches, widely used in image retrieval and it gives reasonably good results. The color space is partitioned and for each partition the pixels with a color within its range are counted, resulting in a histogram representation.

As mentioned before, PAMIR works better as the dimensionality of the image representation grows. To this end we have selected two different images representation. The first, small one, is composed of 512 visual words and 512 color histogram bins, leading to a total of 1024 dimensions. The second, large one, is composed of 4092 visual words and 512 color histogram bins, leading to a total of 4604 dimensions.

In this experiment LSH was used for the FKNIR approach. The LSH parameters were tuned using the validation dataset, selecting those parameters that show a good balance between speed and precision. The parameters were set to $r = 100$, $w = 100$ and $b = 14$. One more experiment is carried out tuning the parameters of LSH in order to measure the time needed to obtain similar precision results as with PAMIR. This approach is referred as FKNIR*.

Table 3 and 4 show the results obtained for the small and large datasets respectively. In both experiments the KNIR methods clearly outperform PAMIR in both precision and speed. As expected, the results of PAMIR are better for higher dimensions but still the precision at top 10 is far from the KNIR method which represents a 30% of relative improvement over PAMIR. Moreover the computational time of PAMIR is clearly higher than in the here proposed methods. The difference between PAMIR and FKNIR* training time is remarkable. It is important to note that the results of KNIR are almost identical in both image representations, so small (and fast) image representations are enough in order to obtain good results using the here proposed model.

Table 3. ImageCLEF results for the 512-512 representation

Method	P@10(%)	AvgP (%)	Training (secs)
PAMIR	3.0	9.33	96
KNIR	4.1	11.93	35
FKNIR	3.8	11.27	8
FKNIR*	3.3	9.0	4

Table 4. ImageCLEF results for the 4096-512 representation

Method	P@10(%)	AvgP (%)	Training (secs)
PAMIR	3.1	10.3	196
KNIR	4.0	11.62	45
FKNIR	3.9	11.78	16
FKNIR*	3.7	10.5	9

Comparing these results with the results obtained in Corel-large, the PAMIR behaviour is slower due to the sparsity of the image representation. In Corel the images have 40 non-zero visual words in average while in ImageClef-large the images have more than 300 non-zero visual words in average. Another important computational issue that affects PAMIR is the number of iterations of the online learning approach. However the computational cost of the here proposed approach is fixed and does not depend on an iterative procedure at all.

5 Conclusions

The here proposed approach has shown to be effective for the problem of image retrieval from text queries. The results obtained show that KNIR outperform the state-of-the-art techniques while preserving a very good computational behaviour. The Bayesian approach adopted has shown to be very appropriate for this particular problem and does not entail any complex learning stage neither more computations on the test phase. The fast implementation of KNIR, FKNIR, obtains further computational benefits while keeping the precision performance on similar values than KNIR. On the other hand, KNIR can be used in applications where the image representation has to keep some structural information. In this sense the proposed approach only requires to be able to compute distances between the objects represented. Moreover KNIR do not require any particular high dimensional representation of the images to be effective.

References

1. Barnard, K., Duygulu, P., de Freitas, N., Forsyth, D., Blei, D., Jordan, M.I.: Matching words and pictures. Journal of Machine Learning Research 3, 1107–1135 (2003)
2. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV (2004)
3. Deselaers, T., Keysers, D., Ney, H.: Discriminative training for object recognition using image patches. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005), pp. 157–162 (2005)
4. Duygulu, P., Barnard, K., de Freitas, N., Forsyth, D.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Seventh European Conference on Computer Vision, vol. IV, pp. 97–112 (2002)
5. Escalante, H.J., Hernández, C., Gonzalez, J., López, A., Montes, M., Morales, E., Sucar, E., Grubinger, M.: The segmented and annotated iapr tc-12 benchmark. Computer Vision and Image Understanding (2009)

6. Everson, R.M., Fieldsend, J.E.: A variable metric probabilistic k-nearest-neighbours classifier. In: Intelligent Data Engineering and Automated Learning-IDEAL, pp. 654–659 (2004)
7. Gionis, A., Indyk, P., Motwani, R.: Similarity search in high dimensions via hashing. In: Proceedings of the 25th Very Large Database (VLDB) Conference, pp. 518–529 (1999)
8. Grangier, D., Bengio, S.: A discriminative kernel-based model to rank images from text queries. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 30(8), 1371–1384 (2008)
9. Guillaumin, M., Mensink, T., Verbeek, J., Schmid, C.: Tagprop: Discriminative metric learning in nearest neighbor models for image auto- annotation. In: International Conference on Computer Vision (2009)
10. Indyk, P.: Stable distributions, pseudorandom generators, embeddings, and data stream computation. J. ACM 53(3), 307–323 (2006)
11. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: SIGIR 2003: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp. 119–126. ACM, New York (2003)
12. Jeon, J., Manmatha, R.: Using maximum entropy for automatic image annotation. In: International Conference on Image and Video Retrieval, pp. 2040–2041 (2004)
13. Makadia, A., Pavlovic, V., Kumar, S.: A new baseline for image annotation. In: European Conference on Computer Vision (2008)
14. Manocha, S., Girolami, M.A.: An empirical analysis of the probabilistic k-nearest neighbour classifier. Pattern Recognition Letters 28(13), 1818–1824 (2007)
15. Monay, F., Gatica-Perez, D.: Plsa-based image auto-annotation: constraining the latent space. In: ACM Multimedia, pp. 348–351 (2004)
16. Naphade, M.: On supervision and statistical learning for semantic multimedia analysis. Journal of Visual Communication and Image Representation 15(3) (2004)
17. Pan, J.Y., Yang, H.J., Duygulu, P., Faloutsos, C.: Automatic image captioning. In: International Conference on Multimedia and Expo, ICME (2004)
18. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV (2003)
19. Vogel, J., Schiele, B.: Natural scene retrieval based on a semantic modeling step. In: Enser, P.G.B., Kompatsiaris, Y., O'Connor, N.E., Smeaton, A., Smeulders, A.W.M. (eds.) CIVR 2004. LNCS, vol. 3115, pp. 207–215. Springer, Heidelberg (2004)

Maximum *a Posteriori* Based Kernel Classifier Trained by Linear Programming

Nopriadi and Yukihiko Yamashita

Tokyo Institute of Technology
2-12-1, Ookayama, Meguro-ku, Tokyo 152-8552, Japan
nopriadi@yahoo.com, yamasita@ide.titech.ac.jp
http://www.titech.ac.jp/

Abstract. We propose a new approach for classification problem based on the maximum a posteriori (MAP) estimation. The necessary and sufficient condition for the cost function to estimate a posteriori probability was obtained. It was clarified by the condition that a posteriori probability cannot be estimated by using linear programming. In this paper, a kernelized function of which result is the same as that of the MAP classifier is estimated. By relieving the problem from to estimate a posteriori probability to such a function, the freedom of cost function becomes wider. We propose a new cost function for such a function that can be solved by using linear programming. We conducted binary classification experiment by using 13 datasets from the UCI repository and compared the results to the well known methods. The proposed method outperforms the other methods for several datasets. We also explain the relation and the similarity between the proposed method and the support vector machine (SVM). Furthermore, the proposed method has other advantages for classification. Besides it can be solved by linear programming which has many excellent solvers, it does not have regularization parameter such as C in the cost function in SVM and its cost function is so simple that we can consider its various extensions for future work.

Keywords: Maximum a posteriori, Kernel Function, Linear Programming, Cost Function.

1 Introduction

In statistics, the method of maximum a posteriori (MAP) estimation can be used to obtain a point estimate of an unobserved quantity on the basis of empirical data. The MAP based method is adopted and studied in machine learning and pattern recognition. Many methods are then developed and implemented based on this method for the purpose of classification, detection, decision, and also estimation. We can explore some of them in references [1], [2], [3], [4], [5]. Many patterns (data sets) such as speech recognition [14], DNA sequences classification [18], image watermark identification [6], face image recognition [1], breast-cancer detection [15] have been used on the methods and achieve promising performance.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 493–502, 2010.

It is important to note that designing a classifier based on MAP depends on the available information about *a posteriori* probabilities. Otherwise, based on Bayes' theorem, the information about *a prior* probabilities and the *likelihood* are imperative. In fact, *a posteriori* probabilities is difficult to be determined directly from data, as well as the *likelihood* is. Even some MAP based method needs other parameters such as a mean vector and covariance matrix for each class [1]. However, the methods to estimate *a posteriori* probability are available and in particular neural network can be used to estimate *a posteriori* probabilities [7], [8], [10], [19].

The papers [7], [8] adress their discussion on the problem of designing cost functions to estimate *a posteriori* probabilities. Any cost function which provides *a posteriori* class probabilities is called *Strict Sense Bayesian* (SSB)[7]. General conditions for the SSB cost function can be found in [7], [9].

In this paper we propose a new approach for the classification problem. It is based on the maximum *a posteriori* (MAP) estimation. A kernelized function $w(x, y)$ that provides the same result as the MAP classifier is estimated. We do not estimate directly *a posteriori* probability $P(y|x)$. By relieving the problem from to estimate *a posteriori* probability to such a function, the freedom to choose the cost functions becomes wider. In other words, we do not need to consider if the cost function is SSB or not.

Beside that, the SSB cost function in [7] is nonlinear, then it can be solved by nonlinear optimization. We can not use linear programming to solve the optimization problem if the cost function is nonlinear function [15], [16]. In this paper we provide a cost function that can be solved by linear programming, which has many excellent solvers.

In order to evaluate the perpormance of our proposed method we conducted binary classification experiment by using 13 datasets. Based on the results we compare the performance of the proposed method to widely known methods. The conclusion is that our proposed method is competitive to the others. We also discuss the relation and the similarity between the proposed method and SVM. Finally, we explain that the cost function of our proposed method is so simple and there will be much room to explore its extension.

2 Maximum a Posteriori (MAP) Estimation

In this section we start explaining some important probability formulas and also the definition of maximum *a posteriori* classification, then reviewing a former method to estimate *a posteriori* probability and finally proposing our new approach.

Let y be the category to be estimated from a data x. $P(x)$, $P(y)$, $P(x|y)$, and $P(y|x)$ denote respectively as a prior probability density function (p.d.f) of x, a prior probability of y, a conditional p.d.f of x given y, and *a posteriori* of y. Bayes theorem can be derived from the joint probability of x and y (i.e. $P(x, y)$) as follows:

$$P(x, y) = P(x|y)P(y) = P(y|x)P(x) \; . \tag{1}$$

The expectation value of a function $f(x)$ in a data x is written as:

$$E_x\{f(x)\} = \int f(x)P(x)dx \ . \tag{2}$$

Then the MAP estimates category $\hat{y}$ that is defined as the mode of the posterior probability as follows:

$$\hat{y} = \arg\max_y \ P(y|x) \ . \tag{3}$$

A classifier system is designed to estimate category $\hat{y}$ for an unlearned pattern x. As shown in eq.(3) we need information about $P(y|x)$, even in neural networks the estimation of $P(y|x)$ is imperative in making decision.

For instance, in [7] Suerri *et al.* proposed a cost function $C(h,d)$ to estimate *a posteriori* probabilities. We substitute a vector of functions that should be *a posteriori* into h and a vector that expresses a category into d. We describe the criterion using $C(h,d)$ for a binary classification problem as follows:

$$\sum_{y\in\{+1,-1\}} E_x P(y|x) C\left((h(x), \begin{pmatrix}\delta_{y,+1}\\ \delta_{y,-1}\end{pmatrix}\right)$$

where $h(x)$ is a 2-dimensional vector of functions of x to be optimized and δ is the Kronecker delta. If $C(h,d)$ is SSB, the criterion above, the function h becomes *a posteriori* probability, i.e.

$$h(x) = \begin{pmatrix}P(+1|x)\\ P(-1|x)\end{pmatrix} .$$

They also found the necessary and sufficient condition for a symmetric and separable SSB cost function, that is $C(h,d)$ is expressed in the following form

$$C(h,d) = \sum_{i=1}^{2} \int_{d_i}^{h_i} g_i(\alpha)(\alpha - d_i)\, d\alpha + r(d)$$

where $g_i(\alpha)$ is any positive function ($g_i(\alpha) > 0, 0 \le \alpha \le 1$) which does not depend on d_i and $r(d)$ is an arbitrary function which does not depend on h. We can see that $\int_{d_i}^{h_i} g_i(\alpha)(\alpha - d_i)\, d\alpha + r(d)$ is a nonlinear function, then it is to be solved by nonlinear optimization.

In this research we do not estimate $P(y|x)$ directly for classification, but estimating a function $w(x,y)$. We could use $w(x,y)$ if it satisfies:

$$\arg\max_y \ w(x,y) = \arg\max_y \ P(y|x) \ . \tag{4}$$

Furthermore in the next section we will explain the advantage of our new approach that the cost function can be directly optimized with linear programming.

3 Model Formalization

Now we consider a set of pattern samples $\{(x_i, y_i)\}_{i=1}^N$ and we restrict a problem to binary classification, i.e. $y \in \{-1, +1\}$. We need a criterion to choose the function $w(x, y)$ in order to satisfies (4). The criterion of $w(x, y)$ proposed in this paper is written as follows:

maximize

$$\sum_{y\in\{-1,+1\}} E_x P(y|x)\ \min(w(x,y), 1)\ . \tag{5}$$

subject to:

$$\sum_{y\in\{-1,+1\}} E_x w(x,y) = 1\ , \tag{6}$$

$$w(x,y) \geq 0\ . \tag{7}$$

To achieve optimum $w(x, y)$ we maximize the expectation function of $P(y|x)$ times $\min(w(x, y), 1)$. In this cost function we evaluate the value of $w(x, y)$ up to one. Beside that the constraint (6) and (7) are also consistent with probability laws.

It is clear that the solution of eqs.(5), (6), and (7) is given as

$$w(x,+1) = \begin{cases} 1 & \text{if } P(+1|x) > P(-1|x) \\ 0 & \text{if } P(+1|x) < P(-1|x) \\ \beta_x & \text{if } P(+1|x) = P(-1|x) \end{cases}$$

$$w(x,-1) = \begin{cases} 0 & \text{if } P(+1|x) > P(-1|x) \\ 1 & \text{if } P(+1|x) < P(-1|x) \\ 1-\beta_x & \text{if } P(+1|x) = P(-1|x) \end{cases} \tag{8}$$

where β_x is an arbitrary number $(0 \leq \beta_x \leq 1)$. We can see $w(x, y)$ provides the same results of the MAP classifier.

We define $w(x, y)$ by using a kernel function $k(x, z)$, that can be written in the form

$$w(x,y) = \sum_{j=1}^{N} \alpha_{y,j} k(x, x_j) \tag{9}$$

and we use Gaussian kernel function which can be expressed by

$$k(x,z) = \exp\left(-\gamma \left\|x - z\right\|^2\right)\ . \tag{10}$$

The parammeter γ determines the width of the Gaussian kernel and in the training mode we adjust it for each pattern samples.

By exchanging the ensemble mean by sample mean and subtituting eq.(9) to eq.(5), we have a cost function in the form

$$\sum_{y\in\{-1,+1\}} E_x P(y|x)\ \min(w(x,y),1) \backsimeq \frac{1}{N}\sum_{i=1}^{N} \min\left(\sum_{j=1}^{N} \alpha_{y_i,j} k(x_i, x_j), 1\right)\ . \tag{11}$$

The constraints (6) and (7) respectively become:

$$\sum_{y\in\{-1,+1\}} E_x w(x,y) \backsimeq \frac{1}{N} \sum_{y\in\{-1,+1\}} \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{y,j} k(x_i,x_j) = 1 \ . \tag{12}$$

$$w(x_i,y) = \sum_{j=1}^{N} \alpha_{y,j} k(x_i,x_j) \geq 0 \ , \quad \forall i \ . \tag{13}$$

Now, we have an optimization problem to be solved. It consist of a cost function (11), of which the maximum value we want to find, along with a set of constraints (12) and (13). To simplify the calculation in linear programming problem, the condition (13) could be changed with

$$\alpha_{y,j} \geq 0 \ , \tag{14}$$

if $k(x,y) \geq 0$. However, even if we use condition (13), the optimization problem still could be solved by linear programming. In many cases if $k(x,y) \geq 0$, the adoption (14) allows us to reduce the number of variables in a linear programming problem and the experiment shows better results.

In order to realize eq.(11), we introduce a slack variables $\xi_i \geq 0$, then we write it in the form

$$\min\left(\sum_{j=1}^{N} \alpha_{y_i,j} k(x_i,x_j), 1\right) = \sum_{j=1}^{N} \alpha_{y_i,j} k(x_i,x_j) - \xi_i .$$

By introducing the above form to eq.(11), we have a linear programming problem of $3N$ variables $\alpha_{y,j}$ and ξ_i ($y \in \{-1,+1\}$, $i,j = 1,2,...,N$) that can be expressed in the form as follows:

maximize

$$\sum_{y\in\{-1,+1\}} \sum_{j=1}^{N} \left(\sum_{i=1}^{N} \delta_{y,y_i} k(x_i,x_j)\right) \alpha_{y,j} - \sum_{i=1}^{N} \xi_i \ ,$$

In other word $w(x,y)$ is a surrogate function that behaves in as similar way to *a posteriori* probability. subject to:

$$\sum_{y\in\{-1,+1\}} \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{y,j} k(x_i,x_j) = N,$$

$$\alpha_{y,j} \geq 0 \qquad (y \in \{-1,+1\}, j = 1,2,...,N),$$

$$\sum_{j=1}^{N} \alpha_{y_i,j} k(x_i,x_j) - \xi_i \leq 1 \qquad (i = 1,2,...,N),$$

$$\xi_i \geq 0 \qquad (i = 1,2,...,N).$$

Now we have a cost function that can be solved by using a linear programming.

4 Experiment

In the experiment we used an open source package GNU Linear Programming Kit (GLPK) to carry out optimization problem. The GLPK is a set of routines designed to solve large-scale linear programming (LP), mixed integer programming (MIP), and other related problems. Then, in order to evaluate the performance of our proposed method we conducted experiment with two-class classification problem, using 13 data sets from the UCI repository. The properties of the data sets we used are shown in Table 1.

Table 1. Overview of the 13 data sets used in the experiment

Data set	# training samples	# test samples	# realization	Dimension
Banana	400	4900	100	2
Breast cancer	200	77	100	9
Diabetis	468	300	100	8
Flare-Solar	666	400	100	9
German	700	300	100	20
Heart	170	100	100	13
Image	1300	1010	20	18
Ringnorm	400	7000	100	20
Splice	1000	2175	20	60
Thyroid	140	75	100	5
Titanic	150	2051	100	3
Twonorm	400	7000	100	20
Waveform	400	4600	100	21

In order to have a more sophisticated model selection we considered the generalization performance of the model. For this purpose we ran 5-fold cross validation to estimate the parameter γ. We treated each pattern (from banana to waveform) separately by the cross validation. Our goal is to have an appropriate parameter for each pattern. We used the first 5 realizations of train data for validation. For each realization we performed a cross validation in the following manner: We split a training sample set into 5 equally sized and disjoint subsamples. Of the 5 subsamples, a single subsample was retained as test data and the remaining 4 subsamples were combined to form a training data for cross validation. We performed validation with the new train and test data then calculated the error. The cross validation process was repeated 5 times and each of the 5 subsamples was used exactly once as the test data. The 5 results of the folds then was averaged to produce a single estimation error. Since we used 5 realizations, then we chose a median of the best values of parameter (with minimum error).

Based on the parameter γ we performed experiment by using test data sets as shown in Table 1. The result of experiment is summarized in Table 2.

Table 2. Result of Experiment. Comparison with other methods. The best result is in bold face.

Data set	γ	Proposed	SVM	RBF	AB	AB$_R$	KFD
Banana	4.217	**10.7 ± 0.6**	11.5 ± 0.7	10.8 ± 0.6	12.3±0.7	10.9±0.4	10.8±0.5
B.Cancer	0.316	**25.8 ± 4.0**	26.0 ± 4.7	27.6 ± 4.7	30.4±4.7	26.5 ± 4.5	**25.8 ± 4.6**
Diabetis	0.649	25.0 ± 1.9	23.5 ± 1.7	24.3 ± 1.9	26.5±2.3	23.8±1.8	**23.2±1.6**
F.Solar	1.778	33.0 ± 7.8	**32.4 ± 1.8**	34.4± 2.0	35.7±1.8	34.2±2.2	33.2±1.7
German	0.270	25.3 ± 2.3	**23.6 ± 2.1**	24.7±2.4	27.5±2.5	24.33±2.1	23.7±2.2
Heart	0.237	17.8 ± 3.2	**16.0 ± 3.3**	17.6±3.3	20.3±3.4	16.5±3.5	16.1±3.4
Image	17.783	4.0 ± 2.7	3.0 ± 0.6	3.3±0.6	**2.7±0.7**	**2.7±0.6**	4.8±0.6
Ringnorm	0.090	2.5 ± 1.0	1.7 ± 0.1	1.7±0.2	1.9±0.3	1.6±0.1	**1.5±0.1**
Splice	0.129	24.2 ± 2.4	10.9 ± 0.7	10.0±1.0	10.1±0.5	**9.5±0.7**	10.5±0.6
Thyroid	1.685	5.00 ± 2.3	4.8 ± 2.2	4.5±2.1	4.4±2.2	4.6 ±2.2	**4.2±2.1**
Titanic	0.562	**21.6 ± 5.0**	22.4 ± 1.0	23.3±1.3	22.6±1.2	22.6±1.2	23.2 ±2.0
Twonorm	0.140	**2.5 ± 0.2**	3.0 ± 0.2	2.9±0.3	3.0±0.3	2.7 ±0.2	2.6 ±0.2
Waveform	0.225	11.0 ± 1.8	9.9 ± 0.4	10.7±1.1	10.8 ±0.6	**9.8±0.8**	9.9±0.4

Table 3. Computational time of learning and classification process for all realizations

Data set	# realization	Proposed (in seconds)	SVM (in seconds)
Banana	100	135.3	4.030
B.Cancer	100	20.6	0.700
Diabetis	100	216	2.332
F.Solar	100	498.2	4.536
German	100	765.7	5.734
Heart	100	19.9	0.596
Image	20	$1.252\text{x}10^4$	1.938
Ringnorm	100	164.6	14.104
Splice	20	365.4	8.402
Thyroid	100	13.9	0.308
Titanic	100	13.4	1.170
Twonorm	100	181.4	9.800
Waveform	100	200.9	10.444

In the experiment we also measure the computational time which is needed in learning and classification process for all realizations of each data set. Table 3 shows the computational time.

5 Discussion

To evaluate the performance of our proposed method we compare our experiment result to other state-of-the-art methods, as shown in Table 2. Here we choose Support Vector Machine (SVM), a single RBF classifier, AdaBoost (AB), regularized AdaBoost (ABR), and Kernel Fisher Discriminant (KFD) as comparators. The experiment data in [11] is used for comparison. We are interested in comparing our new approach to the other methods because we know that the SVM and the boosting (in general the margin-based classifiers) have demonstrated their excellent performances in binary classification. Meanwhile, KFD is very competitive and in some cases even superior to the other algorithms on almost all data sets [11].

The result in Table 2 shows that our proposed method has promising performance. We can say that it is competitive to the others and superior on some data sets (banana, breast cancer, titanic and twonorm).

Regarding with the computational complexity we can see in Table 3 that the proposed method is slower than LIBSVM (library for support vector machines) [20]. However, LIBSVM is a specialized program for SVM. On the other hand we used GLPK that is a general purpose library. Therefore, it is difficult to compare the computational complexity from these results. Furthermore, the computational time of the proposed method is enough fast to apply it to many problems.

The next part of this section we discuss in particular relation and similarity between our proposed method and SVM. Maximization of eq.(11) can be translated to minimization of a loss function. Many loss functions are proposed such as the hinge loss and the fisher consistent loss [12]. Originally the SVM is designed for the binary classification problem and its paradigm has a nice geometrical interpretation of discriminating one class from another by a hyperplane with the maximum margin [13]. The cost function of SVM to obtain the optimal separating hyperplane is written as

$$\min \; \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j k(x_i,x_j) + C\sum_{i=1}^{N}\zeta_i \; , \tag{15}$$

where α_i is Lagrange multiplier which is optimized, C is the tuning parameter, ζ_i is the non negative slack variables. In the SVM the extra term $C\sum_{i=1}^{N}\zeta_i$ in the cost function is to accommodate some data which is not linearly separable by the hyperplane. Then the slack variables ζ_i express the hinge loss. If we neglect the treshold, the ζ_i can be expressed as follows:

$$\zeta_i = \max\left(0,\; 1-\sum_{j=1}^{N}\alpha_j y_j k(x_i,x_j)\right) \; .$$

The parameter C is also called as the margin parameter that determines the tradeoff between the maximization of the margin and the minimization of the classification error. This term is to balance the goals of maximum margin separation and the correctness of training set classification.

The concept of hinge loss in SVM is quite similar to our criterion (11). We have the following arithmetic relation:

$$\min(a,1) = 1-\max(0,(1-a)).$$

In both criteria, the classification functions are substituted into a. Therefore eq.(11) is considered as a hinge loss. Instead of the regularization term (the first term in eq.(15)), the eq.(12) is introduced. The advantage of our proposed method is that we do not have a regularization parameter such as C in the cost function of SVM. We only need one parameter, that is γ.

6 Conclusion

In this paper we proposed a new approach for classification problem based on maximum *a posteriori* probability. We do not estimate directly $P(y|x)$, but we use a kernelized function $w(x, y)$ that can be regarded as a surrogate function that behaves in a similar way to MAP Classifier. The advantage of this approach is the cost function can be directly optimized with linear programming. The experiment using 13 data sets from the UCI repository shows that our proposed method has promising performance and it is competitive enough to the other state-of-the-art classification methods. We also explained the relation between the proposed method and the support vector machine (SVM). The similarity between the hinge loss and the proposed criterion was discussed. Furthermore, we can consider its various extensions, similar to the extensions of SVM, to improve the classifier performance in the future work. It is very possible because our proposed cost function is so simple.

References

1. Xu, Z., Huang, K., Zhu, J., King, I., Lyu, M.R.: A Novel Kernel-Based Maximum a Posteriori Classification Method. J. Neural Networks 22, 121–146 (2009)
2. Gauvain, J.L., Lee, C.H.: Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. J. IEEE Trans. Speech and Audio Processing 2, 291–298 (1994)
3. Chen, K., Wang, H.: Eigenspace-Based Maximum a Posteriori Linear Regression for Rapid Speaker Adaptation. In: Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing, Salt Lake City, USA, vol. 1, pp. 917–920 (2001)
4. Igual, J., Camachoa, A., Bernabeua, P., Vergarab, L.: A Maximum a Posteriori Estimate for the Source Separation Problem with Statistical Knowledge about the Mixing Matrix. J. Pattern Recognition Letters 24, 2519–2523 (2003)
5. Siohan, O., Myrvoll, T.A., Lee, C.H.: Structural Maximum a Posteriori Linear Regression for Fast HMM adaptation. J. Computer Speech & Language 16, 5–24 (2002)
6. Ng, T.M., Garg, H.K.: A Maximum a Posteriori Identification Criterion for Wavelet Domain Watermarking. International Journal of Wireless and Mobile Computing 3, 265–270 (2009)
7. Sueiro, J.C., Arribas, J.I., Munoz, S.E., Vidal, A.R.F.: Cost Functions to Estimate a Posteriori Probabilities in Multiclass Problems. J. IEEE Trans. Neural Networks 10, 645–656 (1999)
8. Arribas, J.I., Sueiro, J.C., Adali, T., Vidal, A.R.F.: Neural Architetures for Parametric Estimation of a Posteriori Probabilities by Constrained Conditional Density Functions. In: Proc. IEEE Workshop on Neural Networks for Signal Processing (NNSP), Madison, Wisconsin, USA, pp. 263–272 (1999)

9. Miller, J.W., Goodman, R., Smyth, P.: Objective Functions for Probability Estimation. In: International Joint Conference on Neural Networks, Seattle, USA, vol. 1, pp. 881–886 (1991)
10. Ruck, D.W., Rogers, S.K., Kabrisky, M., Oxley, M.E., Suter, B.W.: The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminant Function. J. IEEE Trans. Neural Network 1, 296–298 (1990)
11. Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Müller, K.R.: Fisher Discriminant Analysis with Kernels. In: IEEE Signal Processing Society Workshop In Neural Networks for Signal Processing IX, Madison, Wisconsin, USA, vol. 10, pp. 41–48 (1999)
12. Zou, H., Zhu, J., Hastie, T.: New Multicategory Boosting Algorithms Based on Multicategory Fisher-Consistent Losses. Annals of Applied Statistics 2, 1290–1306 (2008)
13. Lee, Y., Lin, Y., Wahba, G.: Multicategory Support Vector Machines Theory and Application to the Classification of Microarray Data and Satellite Radiance Data. Journal of the American Statistical Association 99, 67–81 (2004)
14. Huy, T., Takeda, K., Itakura, F.: Maximum a Posterior Probability and Cumulative Distribution Function Equalization Methods for Speech Spectral Estimation with Application in Noise Suppression Filtering. In: Faundez-Zanuy, M., Janer, L., Esposito, A., Satue-Villar, A., Roure, J., Espinosa-Duro, V. (eds.) NOLISP 2005. LNCS (LNAI), vol. 3817, pp. 328–337. Springer, Heidelberg (2006)
15. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (2004)
16. Kolman, B., Beck, R.E.: Elementary Linear Programming with Applications, 2nd edn. Academic Press, Elsevier Science & Technology Books (1995)
17. Arribas, J.I., Sueiro, J.C., Lopez, C.A.: Estimation of Posterior Probabilities with Neural Networks. In: Hanbook of Neural Engineering, IEEE Press, A John Wiley and Sons Inc. (2007)
18. Loewenster, D.M., Berman, H.M., Hirsh, H.: Maximum A Posteriori Classification of DNA Structure from Sequence Information. In: Proceedings of Pacific symposium on Biocomputing, Hawaii, USA, pp. 667–668 (1998)
19. Jaroudi, A.E., Makhoul, J.: A New Error Criterion for Posterior Probability Estimation with Neural Nets. In: International Joint Conference on Neural Networks, San Diego, CA, USA, vol. 90, pp. 185–192 (1990)
20. Chang, C.C., Lin, C.-J.: LIBSVM: a Library for Support Vector Machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm

Improvement of the Disc Harmonic Moments Descriptor by an Exponentially Decaying Distance Transform

Noureddine Ennahnahi, Mohammed Oumsis, and Mohammed Meknassi

LISQ Laboratory, Computer Science Department, Sciences faculty, Dhar Mehraz,
PO.Box 1796 (Atlas) Fez, Morocco
nahnnourd@yahoo.fr, oumsis@yahoo.com, m.meknassi@gmail.com

Abstract. The authors propose an improvement of a recent region-based shape descriptor inspired by the 3D spherical harmonics: the Disk Harmonic Moments Descriptor (DHMD). The binary image is weighted by an exponentially decaying distance transform (EDDT) before applying the disc harmonic transform (DHT) introduced recently as a good shape representation. The performance of the improved DHMD is compared to other recent methods from the same category. Set B of the MPEG-7 CE-1-Shape database is used for experimental validation. To benchmark the performance of the compared descriptors precision-recall pair is employed. The proposed approach seems be more efficient and effective if compared to its competitors.

Keywords: Spherical harmonics, Legendre polynomials, Distance transform, Salience Distance Transform, region-based shape descriptor, Content-based image retrieval.

1 Introduction

Content-based image retrieval (CBIR) has become one of the most important applications in computer vision. The demand for higher retrieval quality delivered at a short time has brought on a vast amount of research activity to improve the underlying techniques of CBIR. The main task in CBIR resides in highlighting stable information which should allow the similarity measurement between images. Shape descriptors have interesting properties. This kind of image descriptors is suitable to rather represent an object on an image than the whole image itself.

Two categories of shape descriptors exist: region-based and contour-based. We can cite the Fourier Descriptor FD [1] and the well-known CSSD, namely the Curvature Scale-Space [2][3], as contour-based descriptors. The Angular Radial Transformation [4], the geometrical moments [5], the Legendre moments [6][7], the Zernike moments [6][7][8], the pseudo-Zernike moments [6][7][9] and the Generic Fourier Descriptor [10] are some occurrences of region-based descriptors. Many comparative studies have been proposed in the literature [1] [11].

Funkhouser & al. [12] propose a 2D analog version of their three dimensional spherical harmonics based descriptor, noted (2DSHT), where the process consists in the following steps: (1) the shape boundary undergoes the distance transform. (2)

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 503–511, 2010.

Then the authors sample a collection of circular functions by restricting to different radii. (3) Each circular function is expanded as a sum of trigonometric functions. (4) Using the fact that rotations do not change the amplitude within a frequency, the signature of each circular function is a list of the amplitudes of its trigonometrics. (5) Finally, they combine these different signatures to obtain a 2D feature vector for the boundary contour.

An alternative work was proposed by Pu & al. in [13] and based on an analog strategy 2.5D Spherical Harmonic Transform (2.5DSHT): (1) a bounding sphere for a given 2D drawing is calculated, (2) rays are cast in different directions from the center of mass. The intersection points of the rays with the edges of the drawing are represented in a 3-dimensional coordinate system where the z-value is the distance from the centroïd. (3) An angular mapping of the 2D view is generated. (4) Finally, to they use the fast spherical harmonics transformation method in order to obtain the rotational invariant descriptor.

Sajjanhar & al. in [14] exploit the spherical harmonics to design a 2D shape descriptor, namely (DT_3DSHT). The 2D image undergoes firstly the distance transform. A 3D points cloud is built so that each image pixel is relocated in direction OZ proportionally with its distance to the nearest feature. Then the authors pass to the construction of a 3D model, which requires a triangulation of the points cloud. Then, the spherical harmonics are obtained for the 3D model. This is an approach which requires an intricate preparation of the 2D data to enable, finally, the application of Funkhouser's 3D method. Sajjanhar & al. have proposed in a previous work [15] a similar strategy based on the connectivity information. We retain in this paper the work cited in [14] because it is the most recent.

In [16][17] Ennahnahi et al. propose a disc-sphere mapping method, which has allowed the authors to formulate a novel set of orthogonal basis functions, namely the Disc Harmonic functions DHF. These basis functions, noted in polar coordinates for any point of the unit disc as $H_{l,m}(r,\theta)$ are written in formula (1):

$$H_{l,m}(r,\varphi)=\overline{P_l^m\left(\sqrt{1-r^2}\right)e^{jm\varphi}}\ . \tag{1}$$

where $\overline{P_l^m}$ denotes the normalized associated Legendre polynomial:

$$\overline{P_l^m}(x)=N_l^m P_l^m(x)\ . \tag{2}$$

P_l^m designates the associated Legendre polynomial, N_l^m a normalization factor, θ and φ denote the usual spherical coordinates. More details concerning spherical harmonics should be found in [18].

Ennahnahi et al. have designed, helped by these orthogonal harmonic functions, a robust region-based shape descriptor in [16][17]: the Disc Harmonic Moments Descriptor (DHMD).

In this paper, we propose an improvement of the DHMD descriptor by introducing the Exponentially Decaying Distance Transform EDDT as image function in order to weight pixels within the shape by more expressive information with no significant decrease in contrast of the computation efficiency.

The paper is set out as follows. First, the proposed method is detailed in the two first sections. Second, Experimental results are presented in section 4. Finally, a conclusion and perspectives are underlined in section 5.

2 Our Method

2.1 Choice of the Image Function

The algorithm for computing the Disc Harmonic transformation (DHT) takes an image function as its input. In [16][17], Ennahnahi et al. have used a binary grid and the results were sufficiently promising. To improve the retrieval quality of DHMD, we propose in the present work the use of an image function that describes not only where the points on the shape are, but also how far an arbitrary point is from the boundary. Furthermore, the values of the input grid should fall off to zero for pixels further from the boundary, allowing us to treat correctly the information encapsulated in the shape. To address these issues we define the input grid as the result of an exponentially decaying Euclidean Distance Transform (EDDT). In particular, given a pixel P on a binary image B we define the implicit function $f(P)$ by:

$$f(P)=\begin{cases} 0 & if \quad V(P)=0 \\ EDDT_B(P) & if \quad V(P)=1 \end{cases} . \tag{3}$$

Where *V(P)* is the binary value at the pixel P, and $EDDT_B(P)$ designates the exponentially decaying Euclidean Distance Transform:

$$EDDT_B(P)=\exp(-\frac{D_B(P)}{DM_B}) \cdot \tag{4}$$

Where $D_B(P)$ is the Euclidean Distance Transform [19], giving the distance from P to the nearest point on the shape boundary, and (DM_B) is the average distance from a point on (B) to the center of mass.

The use of the famous Distance transform deserves the following suggestions: Generally, distance transforms are not robust for some kinds of binary images where false edges are detected and some true edges are missed. The basic distance transform is sensitive to local distortions. So we retain the Salient Distance Transform (SDT) which is more stable: we weight the distance from the edges by the salience of the edges [20].

2.2 Overview of the Disc Harmonic Transform

The harmonic moments C_l^m based on the basis functions $H_{l,m}$ are obtained by the following integration formula:

$$C_l^m = \int_0^1 \int_0^{2\pi} H_{l,m}^{*} (r,\varphi) f(r,\varphi) r dr d\varphi \; . \tag{5}$$

Numerically this formula can be calculated as a weighted summation over all the pixels sampled on the unit disc. We retain a stratified sampling with uniform weights.

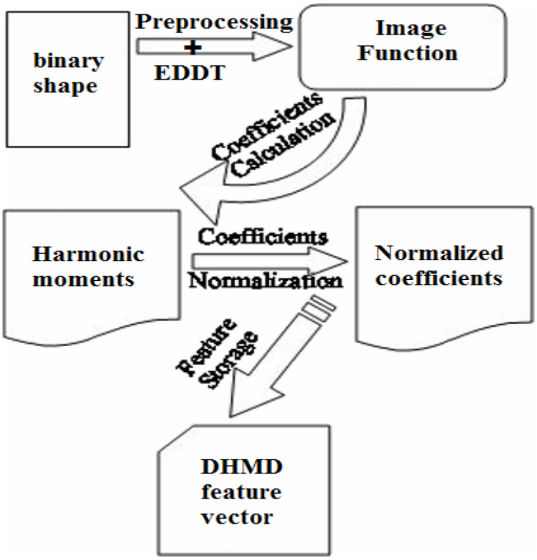


Fig. 1. Steps to extract the feature vector of DHMD descriptor

2.3 DHMD Extraction Algorithm

In [16][17] Ennahnahi et al. expect initially a preprocessing phase which consists of the following steps:

(a) They first delimit the enclosing shape rectangle (the smallest rectangle containing the shape). The object centroïd and the dimension of this area are calculated. Dimension is the double distance from the center of mass to the most distant pixel of the object.

(b) Then they resize the enclosing rectangle to a standard scale. Once these stages are achieved, they ensure translation and scaling invariance, as well as the convenience of the Disc Harmonic Transformation (DHT) using the harmonic basis functions.

(c) They apply, to the preprocessed image, the EDDT transform.

(d) And they convert pixels coordinates so that they hold on a unit disc: by centering and normalizing by half-dimension.

The feature vector is constructed as a triangular matrix, see (Table.1).

Table 1. The DHMD feature vector construction

$\frac{\left\|C_0^0\right\|}{2}$					
$\frac{\left\|C_1^0\right\|}{2}$	$\left\|C_1^1\right\|$				
$\frac{\left\|C_2^0\right\|}{2}$	$\left\|C_2^1\right\|$	$\left\|C_2^2\right\|$			
$\frac{\left\|C_k^0\right\|}{2}$	$\left\|C_k^1\right\|$	⋮	$\left\|C_k^k\right\|$		
⋮	⋮	⋮	⋮	⋱	
$\frac{\left\|C_l^0\right\|}{2}$	$\left\|C_l^1\right\|$	…	$\left\|C_l^k\right\|$	…	$\left\|C_l^l\right\|$

The harmonic coefficients are calculated up to a preferred range of l: The resulting image function undergoes the Disk Harmonic Transformation (DHT) formulated in formula (5). In practice, the values of the harmonic basis functions should be initially pre-calculated over the unit disc and stored for reuse.

3 Experiments Results

We validate the performances of the modified version of DHMD through the MPEG7 CE-1 Part B shape dataset [21]. This database gathers 1400 shapes, classified in 70 classes. Each class gathers 20 similar shapes. Part B of MPEG-7 CE-1 is intended for evaluating the performance regarding similarity-based retrieval. Each shape in the test database is indexed by these shape descriptors (2DSHT, 2.5DSHT, DT_3DSHT, and this improved DHMD) and is used as a query. The feature vector of a query image is extracted then compared to the feature vectors of all the images contained in the database. The measurement of similarity between two shapes is performed using the L1-Norm.

To evaluate the performances of these descriptors, the commonly employed Precision-Recall pair measurement was generated.

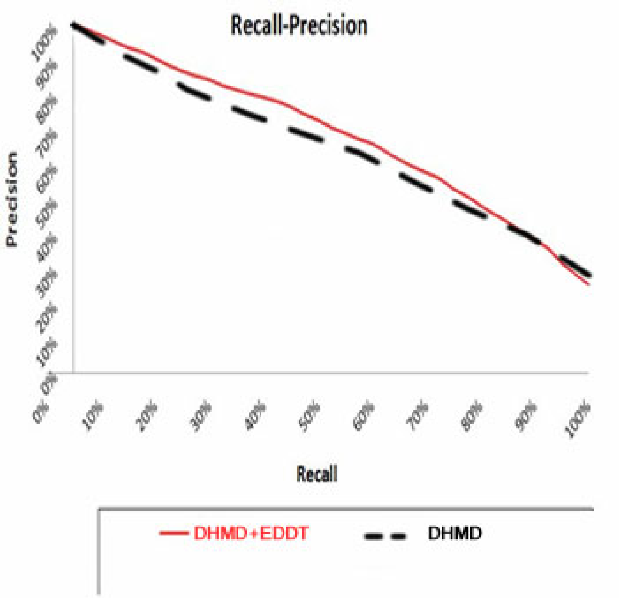


Fig. 2. DHMD with binary function versus DHMD with EDDT

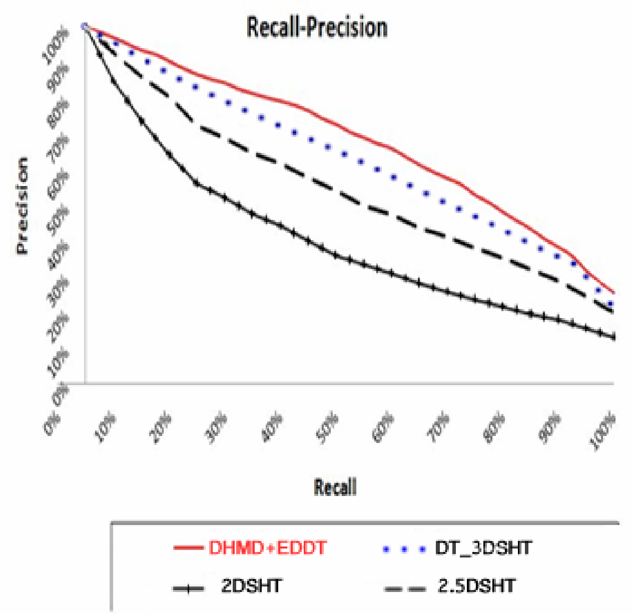


Fig. 3. Recall-precision: DHMD+EDDT and 2.5DSHT use feature vectors with dimension equal to 65 coefficients, whereas the two others descriptors use 512 coefficients

In (Fig.2), it is clear that the use of a function image based on the EDDT transform has improved the performances of the DHMD descriptor. In (Fig.3) we demonstrate that the proposed version of DHMD outperforms (DT_3DSHT) even if Zhang et al. use in [14] 512 components. Compared to (2.5DSHT), one can see that with the same feature vector size the improved DHMD performs sufficiently better. The DT_3DSHT performances are better than the 2.5DSHT, this may be due to the fact that the quality of a descriptor increases with the feature vector size, generally speaking. A visible superiority in performances must be underlined between the improved DHMD descriptor and the 2DSHT.

The results obtained by the older version of the descriptor DHMD can be seen in the previous papers proposed by Ennahnahi and al. in [16][17].

We propose also a comparative study regarding computational efficiency in (Fig.4) where one can see clearly that introducing the EDDT transform in the extraction process of DHMD features hasn't introduced any increase in computational cost. We have tested these descriptors on a Celeron D 1.8 GHz PC with 256 Mo of Memory. The new version of DHMD performs better retrieval results than 2.5DSHT and features also a comparable timing in the extraction stage. The handicap of DT_3DSHT method resides in its processing steps which consume a remarkable computational time. We have obtained these performances because we pre-calculate all the Legendre polynomials required for the DHT. Whereas, 2.5DSHT is faster because they exploit the Fast Sperical Harmonic Transform implemented in the well-known spahrmonickit tool [22].

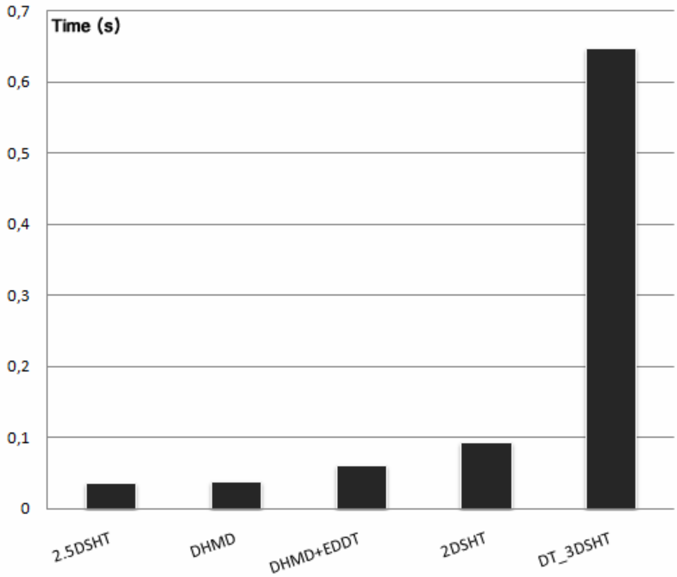


Fig. 4. Comparison of the computational efficiency

4 Conclusion and Perspective

We presented an improvement of a novel region-based 2D shape descriptor easy to implement, inspired by the 3D spherical harmonics. It can describe complex objects consisting of multiple disconnected regions as well as simple objects with or without holes. The salience distance transform has dressed some weak behaviors of the classical distance transform algorithms.

The introduction of the exponentially decaying distance transform to the process of extracting the feature vector sounds benefic in terms of retrieval performances with no significant increase regarding the computational complexity.

In a succeeding work, we will propose a faster version of our descriptor DHMD, with no need to store the pre-calculated Legendre polynomials, in order to have better ratio Quality/time.

References

1. Mokhtarian, F., Mackwoth, A.K.: A theory of multiscale curvature-based shape representation for planar curves. IEEE PAMI 14, 789–805 (1992)
2. Mokhtarian, F., Abbasi, S., Kittler, J.: Efficient and robust retrieval by shape content through curvature scale space. In: International Workshop on Image DataBases and Multimedia Search, Amsterdam, The Netherlands, pp. 35–42 (1996)
3. Kim, W.-Y., Kim, Y.-S.: A New Region-Based Shape descriptor. ISO/IEC MPEG99/M5472, Maui, Hawaii (1999)
4. Hu, M.: Visual pattern recognition by moment invariants. IRE Trans. Infor. Theory IT-8, 179–187 (1962)
5. The, C.-H., Chin, R.T.: On Image Analysis by the Methods of Moments. IEEE Transactions on Pattern Analysis and Machine Intelligence 10(4), 496–513 (1988)
6. Teague, M.R.: Image analysis via the general theory of moments. Journal of Optical Society of America 70(8), 920–930 (1980)
7. Chong, C.-W., Raveendran, P., Mukunda, R.: A comparative analysis of algorithms for fast computation of zernike moments. Pattern Recognition 3, 731–742 (2003)
8. Haddadnia, J., Ahmadi, M., Faez, K.: An efficient feature extraction method with pseudo-zernike moment in rbf neural network-based human face recognition system. EURASIP Journal on Applied Signal Processing, 890–901 (2003)
9. Zhang, D.S., Lu, G.: Shape-based image retrieval using Generic Fourier Descriptor. Signal Processing: Image Communication 17, 825–848 (2002)
10. Zhang, D.S., Lu, G.: A comparative study of Fourier descriptors for shape representation and retrieval. In: Proc. Fifth Asian Conf. on Computer Vision (ACCV 2002), Melbourne, Australia, pp. 646–651 (2002)
11. Zhang, D.S., Lu, G.: Evaluation of MPEG-7 Shape descriptors Against Other Shape Descriptors. ACM Journal of Multimedia Systems 9(1), 15–30 (2003)
12. Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., Jacobs, D.: A Search Engine for 3D Models. ACM Transactions on Graphics 22(1), 83–105 (2003)
13. Pu, J.T., Karthik, R.: On Visual Similarity based 2D Drawing Retrieval. Computer Aided Design 38(3), 249–259 (2006)

14. Sajjanhart, A., Lu, G., Zhang, D., Hou, J., Chen, Y.-P.P.: Spherical Harmonics and Distance Transform for Image Representation and Retrieval. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 309–316. Springer, Heidelberg (2009)
15. Sajjanhart, A., Lu, G., Zhang, D.: Spherical Harmonics Descriptor for 2D-Image Retrieval. In: IEEE International Conference on Multimedia and Expo., ICME 2005, pp. 105–108 (2005)
16. Ennahnahi, N., Bouhouch, A., Oumsis, M., Meknassi, M.: A novel moments generation inspired by 3D spherical harmonics for robust 2D shape description. In: 16th IEEE International Conference on Image Processing (ICIP), pp. 421–424 (2009)
17. Ennahnahi, N., Oumsis, M., Bouhouch, A., Meknassi, M.: Fast shape description based on a set of moments defined on the unit disc and inspired by three-dimensional spherical harmonics. Image Processing IET 4(2), 120–131 (2010)
18. Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation Invariant Spherical Harmonic Representation of 3d Shape Descriptors. In: Symposium on Geometry Processing, (June 2003) pp. 167–175 (2003)
19. Fabbri, R., Costa, L., Da, F., Torelli, J.C., Bruno, O.M.: 2D Euclidean distance transform algorithms A comparative survey. ACM Computing surveys 2008 40(1), 1–44 (2008)
20. Rosi, P.L., West, G.A.W.: Salience Distance transforms. Graphical Models Image Processing 57, 483–521 (1995)
21. Latecki, L.J., Lakamper, R., Eckhardt, U.: Shape Descriptors for Non-rigid Shapes with a Single Closed Contour. In: IEEE Conf. On Computer Vision and Pattern Recognition (CVPR), pp. 424–429 (2000)
22. The SpharmonicKit, http://www.cs.dartmouth.edu/~geelong/sphere/

Feature Level Fusion of Face and Palmprint Biometrics

Dakshina Ranjan Kisku[1], Phalguni Gupta[2], and Jamuna Kanta Sing[3]

[1] Department of Computer Science and Engineering,
Dr. B. C. Roy Engineering College, Durgapur – 713206, India
[2] Department of Computer Science and Engineering,
Indian Institute of Technology Kanpur, Kanpur – 208016, India
[3] Department of Computer Science and Engineering,
Jadavpur University, Kolkata – 700032, India
{drkisku,jksing}@ieee.org, pg@cse.iitk.ac.in

Abstract. This paper presents a feature level fusion of face and palmprint biometrics. It uses the improved K-medoids clustering algorithm and isomorphic graph. The performance of the system has been verified by two distance metrics namely, K-NN and normalized correlation metrics. It uses two multibiometrics databases of face and palmprint images for testing. The experimental results reveal that the feature level fusion with the improved K-medoids partitioning algorithm exhibits robust performance and increases its performance with utmost level of accuracy.

Keywords: Biometrics, Feature Level Fusion, Face, Palmprint, Isomorphic Graph, K-Medoids Partitioning Algorithm.

1 Introduction

Feature level fusion fuses feature sets of different biometric traits under different fusion rules. Since feature level fusion integrates richer and most relevant information of biometric evidences and it is expected to provide more accurate authentication results. It is found to be effective compared to fusion based on match scores, decision, ranks, etc. But fusion of incompatible biometric evidences at feature level is a very hard task. Moreover, the feature spaces for different biometric evidences may be unknown and this may lead to the problem of curse of dimensionality [1]. Also, poor feature representation may degrade the performance of recognition.

Unibiometric identifiers are often affected by problems like lack of invariant representation, non-universality, noisy sensor data and lack of individuality of the biometric trait and susceptibility to circumvention. These problems can be minimized by using multibiometric systems that consolidate evidences obtained from multiple biometric sources. Multibiometrics fusion at match score level, decision level and rank level have extensively been studied while there exists a few feature level fusion approaches. There is enough scope to design an efficient feature level fusion approach. The feature level fusion of face and palmprint biometrics proposed in [3] uses single sample of each trait. Discriminant features using graph-based approach and principal component analysis techniques are used to extract features from face and palmprint.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 512–521, 2010.

Further, a distance separability weighing strategy is used to fuse two sets at feature extraction level. Another approach consisting of face and hand biometrics has been proposed in [4]. In [5], a feature level fusion has been studied where phase congruency features are extracted from face and Gabor transformation is used to extract features from palmprint. These two feature spaces are then fused using user specific weighting scheme. Another approach of face and palmprint biometrics is given in [6]. It makes use of correlation filter bank with class-dependence feature analysis method for feature fusion.

This paper proposes a feature level fusion of face [7] and palmprint [8] biometrics using isomorphic graph [9] and K-medoids [10]. SIFT feature points [11] are extracted from face and palmprint images. The partitioning around medoids (PAM) algorithm [12] is used to partition the face and palmprint images of a set of *n* invariant feature points into *k* number of clusters. For each cluster, an isomorphic graph is drawn on SIFT points belonging to the clusters. Graphs are drawn on each cluster by searching the most probable isomorphic graphs using iterative relaxation algorithm [13] from all possible isomorphic graphs while the graphs are compared between face and palmprint templates. Each pair of clustered graphs are then fused by concatenating the invariant SIFT points and all pairs of isomorphic graphs of clustered regions are further fused to make a single concatenated feature vector. The similar feature vector is also constructed from query pair of face and palmprint. Finally, matching between these two vectors is done by computing the distance using K-Nearest Neighbor [14] and normalized correlation [15] distance. Two multimodal databases are used for testing the proposed technique.

The paper is organized as follows. Next section discusses SIFT features extraction from face and palmprint images. Section 3 presents K-Medoids partitioning of SIFT features into a number of clusters. The method of obtaining isomorphic graphs on the sets of the SIFT points which belong to the clusters is also discussed. Next section presents feature level fusion of clustered SIFT points by pairing two graphs of a pair of clustered regions drawn on face and palmprint images. Experimental results and a comparative study are presented in Section 5 while conclusion is made in the last section.

2 SIFT Keypoints Extraction

David Lowe [11] has proposed a technique to extract features from images which are called Scale Invariant Feature Transform (SIFT). These features are invariant to scale, rotation, partial illumination and 3D projective transform. SIFT provide a set of features of an object that are not affected by occlusion, clutter and unwanted noise in the image. In addition, the SIFT features are highly distinctive in nature which have accomplished correct matching on several pair of feature points with high probability between a large database and a test sample. Initially, the face and palmprint images are normalized by adaptive histogram equalization [1]. Localization of face is done by the face detection algorithm proposed in [16] while the algorithm in [17] is used to localize palmprint. SIFT features [11] are extracted from the face and palmprint images. Each feature point is composed of four types of information – spatial location (x, y), scale (S), orientation (θ) and Keypoint descriptor (K). It uses only keypoint descriptor which consists of a vector of 128 elements showing change in neighborhood intensity of each keypoint. Local image gradients are measured at the selected scale in the region around each keypoint. These gradients are then transformed into a vector that contains 128 elements. These vectors represent local shape distortions and illumination changes.

3 Feature Partitioning and Isomorphic Graph Representation

Lack of well feature representation often degrades the performance. A well representation of feature space and template in terms of invariant feature points may help to increase the overall performance of the system. Clustering of all SIFT feature points into a number of clusters with limited number of invariant points can be an efficient approach of feature space representation. Clustering approach [18] gathers together the keypoints found to be more relevant members of a particular cluster.

3.1 SIFT Keypoints Partitioning

K-medoids clusters is an adaptive version of K-means clustering approach and is used to partition the dataset into a number of groups which minimizes the squared error between the points that belong to a cluster and a point designated as the center of the cluster. The generalization of K-medoids algorithm is the Partitioning around Medoids (PAM) algorithm [12] which is applied to the SIFT keypoints of face and palmprint images to obtain the partitioned of features which can provide more discriminative and meaningful clusters of invariant features.

After applying the PAM clustering technique [12] to the sets of SIFT keypoints for face and palmprint images, each cluster can be verified by Silhouette technique. For each keypoint, let i, $x(i)$ be the average distance of i with all the keypoints in cluster c_m. Consider $x(i+1)$ is an average distance next to $x(i)$. These two successive distances $x(i)$ and $x(i+1)$ are considered to verify the matching of these keypoints i and $(i+1)$ to the cluster where these points are assigned. Then the average distances of i and $(i+1)$ with the keypoints of another single cluster are found. Repeat this process for every cluster in which i and $(i+1)$ are not a member. If the cluster with lowest average distances to i and $(i+1)$ are $y(i)$ and $y(i+1)$, ($y(i+1)$ is the next lowest average distance to $y(i)$), the cluster is known to be the neighboring cluster of the former cluster in which i and $(i+1)$ are assigned. It can be defined by

$$S(i) = \frac{(y(i) + y(i+1))/2 - (x(i) + x(i+1))/2}{\max[((x(i) + x(i+1)), (y(i) + y(i+1))]} \tag{1}$$

From Equation (1) it can be written that $-1 \leq S(i) \leq 1$. When $x(i)+x(i+1) << y(i)+y(i+1)$, $S(i)$ would be close to 1. Distances $x(i)$ and $x(i+1)$ are the measures of dissimilarity of i and $(i+1)$ to its own cluster. If $y(i)+y(i+1)$ is small, then it is well matched; otherwise when the value of $y(i)+y(i+1)$ is large then match is bad. Keypoint is well clustered when $S(i)$ is closer to 1 and when $S(i)$ is negative then it belongs to another cluster. $S(i)$ is zero for the keypoint on the border of any two clusters. The existing algorithm has been extended by taking average distances between $x(i+1)$

and *y(i+1)* for a pair of clusters and a better approximation could be found while PAM algorithm is used to partition the keypoints. The precision of each cluster is increased by this approximation where more relevant keypoints instead of restricted number of keypoints for fusion are taken.

3.2 Establishing Correspondence

To establish correspondence between any two clusters of face and palmprint images, it is observed that more than one keypoint on face image may correspond to single keypoint on the palmprint image. To eliminate false matches and to consider only minimum pair distance from a set of pair distances for making correspondences, it needs to verify the number of feature points available in the cluster of face and that in the palmprint cluster. When the number of feature points in the cluster for face is less than that of the palmprint cluster, many points of interest from the palmprint cluster need to be discarded. If the number of points of interest on the face cluster is more than that of the palmprint cluster, then a single interest point on the palmprint cluster may act as a match point for many points of interest of face cluster. Also, many points of interest on the face cluster may have correspondences to a single point of interest on the palmprint cluster. From all such making correspondences, minimum distance pair is paired. Isomorphic graph for each cluster has been formed by removing few more keypoints from the paired clusters. Iterative relaxation algorithm [13] is used for searching the best possible pair of isomorphic graphs from all possible graphs.

3.3 Isomorphic Graph Representations

To interpret each pair of clusters for face and palmprint, isomorphic graph representation has been used. Each cluster contains a set of SIFT keypoints [11] and each keypoint is considered as a vertex of the proposed isomorphic graph. A one-to-one mapping function is used to map the keypoints of the isomorphic graph constructed on a face cluster to a palmprint cluster while these two clusters have been made correspondence to each other. When two isomorphic graphs are constructed on a pair of face and palmprint clusters with equal number of keypoints, two feature vectors of keypoints are found for fusion. Let F_G and P_G be two graphs and also let *f* be a mapping function from the vertex set of F_G to vertex set of P_G. So when *f* is one-to-one and $f(v_k)$ is adjacent to $f(_{wk})$ in P_G if and o*nly i*f $v_{k\ i}$s adjacent to $w_{k\ i}$n F_G, the fu*nc*tion *f* is known as an isomorphism and two graphs F_G and P_G are isomorphic.

4 Fusion of Keypoints and Matching

4.1 Fusion of Keypoints

To fuse the SIFT keypoint descriptors obtained from each isomorphic pair of graphs for face and for palmprint images, two different fusion rules are applied serially, viz. sum [2] and concatenation [1] rules. Let $F_G\,(v_k) = (v_{k1}, v_{k2}, .., v_{kn})$ and $P_G(w_k) = (w_{k1}, w_{k2}, \ldots, w_{kn})$ be the two sets of keypoints obtained from two isomorphic graphs for a pair of face and palmprint clusters. Suppose there are *m* numbers of clusters in each of face and palmprint images. These two sets of clusters are fused using sum rule and the concatenation rule is further applied to form an integrated feature vector. Suppose, F_{G1}, F_{G2}, …, F_{Gm} sets of keypoints are obtained from a face image after clustering and isomorphism and P_{G1}, P_{G2}, …, P_{Gm} are the sets of keypoints obtained from a palmprint image. The sum rule for the fusion of keypoints is as follows

$$\begin{aligned}
S_{FP1} &= F_{G1} + P_{G1} = \{(v_{k1}^{1} + w_{k1}^{1}), (v_{k2}^{1} + w_{k2}^{1}), (v_{k3}^{1} + w_{k3}^{1}), \ldots, (v_{kn}^{1} + w_{kn}^{1})\} \\
S_{FP2} &= F_{G2} + P_{G2} = \{(v_{k1}^{2} + w_{k1}^{2}), (v_{k2}^{2} + w_{k2}^{2}), (v_{k3}^{2} + w_{k3}^{2}), \ldots, (v_{kn}^{2} + w_{kn}^{2})\} \\
&\text{----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----} \\
&\text{----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----} \\
S_{FPm} &= F_{Gm} + P_{Gm} = \{(v_{k1}^{m} + w_{k1}^{m}), (v_{k2}^{m} + w_{k2}^{m}), (v_{k3}^{m} + w_{k3}^{m}), \ldots, (v_{kn}^{m} + w_{kn}^{m})\}
\end{aligned} \tag{2}$$

where v_{kj} ($j = 1,2,\ldots,n$) and w_{kj} ($j = 1,2,\ldots,n$) of S_{FPj} ($i = 1,2,..,m$) refer to a keypoint of a face graph and a keypoint of a palm graph respectively. In the next step, concatenation rule is applied to the sets of keypoints to form a single feature vector.

4.2 Matching Criterion and Verification

The K-Nearest Neighbor (K-NN) distance [14] and correlation distance [15] approaches are used to compute distances from the concatenated feature sets. In K-NN approach, Euclidean distance metric is used to get K best matches. Let d_i be the Euclidian distance of the concatenated feature set of subject S_i, $i = 1, 2, \ldots. K$, which belong to the K best matches against a query subject. Then S_t is verified against the query if $d_t \leq Th$ where d_t is the minimum of $d_1, d_2, \ldots, d_K$ and Th is the threshold.

On the other hand, the correlation distance metric is used for computing distance between a pair of reference set and probe set. Similarity between two concatenated feature vectors f_1 and f_2 can be computed as follows

$$d = \frac{\sum f_1 f_2}{\sqrt{\sum f_1 \sum f_2}} \tag{3}$$

Equation (3) denotes the normalized correlation between feature vectors f_1 and f_2. Let d_i be the similarity of the concatenated feature set of subject S_i, $i = 1, 2, \ldots K$, with respect to that of a query subject. Then the subject S_t is verified against the query subject if $d_t \geq Th$ where d_t is the maximum of $d_1, d_2, \ldots, d_K$ and Th is the threshold.

5 Experimental Evaluation and Databases

5.1 Databases

The proposed approach has been tested on IIT Kanpur and chimeric multimodal databases. Chimeric database contains face images of ORL face database [20] and palmprint images of Hong Kong Polytechnic University (PolyU) database [21]. IIT Kanpur multibiometrics database consists of 800 face and 800 palmprint images and each subject contributes 2 face and 2 palmprint images. ORL face database contains 400 face images of 40 subjects while PolyU database contains 7,752 palmprint images of 193 subjects (386 palm impressions). From 400 face images of ORL database [20], only 160 face images are taken and 4 face images are taken for each subject. From PolyU database [21], only 160 palm images of 40 subjects having 2 right and 2 left palm images per subject are taken.

In IIT Kanpur face database, images are in controlled environment with maximum tilt of head by 20° from the origin. For evaluation, frontal view faces are used with uniform lighting and minor change in facial expression. These images are acquired in two different sessions. Among the two face images, one image is used as a reference face and the other one is used as a probe face. After preprocessing of face images, it uses the face detection algorithm [16] to get face portion only. On the other hand, face images in ORL database [20] are taken at different sessions with varying the lighting conditions, facial expressions (open/closed eyes, smiling/not smiling) and different facial details (glasses / no glasses). The face images are taken against a dark homogeneous background with the subjects in an upright, frontal position. For the experiment, only frontal view faces are taken with neutral facial expressions and uniform changes in lighting. Among the 4 face images, 2 images are used for reference and remaining two are used for probe face images. Since it contains cropped images, one does not require getting the face portion.

Palmprint images in IIT Kanpur database are also taken in controlled environment with a flat bed scanner having spatial resolution of 200 dpi. Impressions are taken on the scanner with rotation of at most $\pm35^0$ to each user. There are 800 palmprint images of 400 subjects and each subject is contributed 2 images. An image enhancement technique is used to achieve uniform spatial resolution. Finally, palm portion is detected with the help of the technique proposed in [17]. In PolyU palmprint database [21], images are captured at two different sessions and these images are taken under different lighting conditions and by changing the focus of CCD camera. Change in focus is regarded as different palm capturing devices. The images which are of two different sizes, viz. 384×284 and 768×568, are resized to 160×160 and palm portion is detected by the algorithm presented in [17].

5.2 Experimental Results

The performance of the proposed approach is determined using one-to-one matching strategy. Experimental results are obtained with the help of two distance approaches namely, K-Nearest Neighbor (K-NN) distance [14] and normalized correlation [15]. We have also determined the performance of face and palmprint independently. Fused feature set which is obtained from reference face and palmprint images is matched with the feature set obtained from probe pair of face and palmprint images by computing the distance between these two sets. Experiments are for the six distinct cases: (i) face modality using K-NN, (ii) face modality using normalized correlation, (iii) palmprint modality using K-NN, (iv) palmprint modality using normalized correlation, (v) feature fusion using K-NN and (vi) feature fusion using normalized correlation.

False Accept Rate (FAR), False Reject Rate (FRR) and recognition rate are determined from the IIT Kanpur database of 800 face and palmprint images of 400 subjects. Feature level fusion method using normalized correlation outperforms other proposed methods including individual matching of face and palmprint modalities. The correlation metric based feature level fusion has 98.75% recognition rate with 0%

FAR while K-NN based method has the recognition rate of 97.5% with 2% FAR. It can be noted that FAR of all the proposed methods are found to be less than its corresponding FRR. On the other hand, palmprint modality performs better than face modality while K-NN and correlation metrics are used. The distance metrics play an important role irrespective of use of invariant features and isomorphic graphs representations. However, the robust representation of face and palmprint images using isomorphic graphs with use of invariant SIFT keypoints and PAM characterized K-Medoids algorithm makes the proposed fusion method more efficient. In single modality, the same approach has been used. Therefore, the error rates obtained from the single modalities and fusion method are determined under a uniform framework. However, the methodology used for feature level fusion found to be not only superior to other methods and also shows significant improvements in terms of recognition rate and FAR. Table 1 shows different error rates determined on IIT Kanpur database for the methods

Table 1. Different Error Rates on IIT Kanpur Database

METHOD	FAR (%)	RECOGNITION RATE (%)
Face Recognition (K-NN)	7.0	92.50
Face Recognition (Correlation)	6.0	93.75
Palmprint Verification (K-NN)	4.5	94.75
Palmprint Verification (Correlation)	2.5	96.00
Feature Level Fusion (K-NN)	2.0	97.50
Feature Level Fusion (Correlation)	0.0	98.75

In the second phase, when the proposed fusion is applied with both the correlation based and K-NN based distance metrics for the chimeric multibiometric database, the FAR is found to be much lesser than that of FRR. The correlation based distance metric has 99.5% recognition rate with 0% FAR while the K-NN distance metric has 99.25% recognition rate with 1.5% FAR. It is found that the palmprint modality performs better than face modality under both the distance metrics. The combination of SIFT features and isomorphic graph representation is found to be robust for the proposed feature level fusion approach while the IIT Kanpur and chimeric multibiometric databases are evaluated. However, the recognition rates determined from chimeric database is found to be more than that of IIT Kanpur database. This is because of the small size compared to IIT Kanpur database. Table 2 shows the error rates and recognition rates for the proposed techniques on chimeric database.

Table 2. Error and Recognition Rates Determined on Chimeric Database

METHOD	FAR (%)	RECOGNITION RATE (%)
Face Recognition (K-NN)	5.5	93.75
Face Recognition (Correlation)	5	94.5
Palmprint Verification (K-NN)	4	95
Palmprint Verification (Correlation)	2.25	96.75
Feature Level Fusion (K-NN)	1.5	99.25
Feature Level Fusion (Correlation)	0.0	99.5

Sub-graph isomorphism is robust and optimal routing representation where most of the feature points construct good representative graph for the other biometric sample on which the feature points of the first graph is mapped. This characteristic of sub-graph isomorphism makes the feature level fusion more robust.

Table 3. Best Recognition Accuracies for Proposed Fusion and for Fusion Approach in [19]

METHOD	DATABASE	NUMBER OF FEATURE POINTS	RR (%)
Feature level fusion [Experiment I] [19]	Local database (480 faces, 120 hand geometry, 30 individuals)	21 points (8 points from eyes, 4 points from nose and 9 points from hand)	99.23
Feature level fusion [Experiment I] [19]	Local database (480 faces, 120 hand geometry, 30 individuals)	25 points (16 points from eyes, and 9 points from hand)	99.22
Feature level fusion [Experiment II] [19]	Local database (480 faces, 120 hand geometry, 30 individuals)	21 points (8 points from eyes, 4 points from nose and 9 points from hand)	99.43
Feature level fusion [Experiment II] [19]	Local database (480 faces, 120 hand geometry, 30 individuals)	21 points (8 points from eyes, 4 points from nose and 9 points from hand)	99.31
Feature Level Fusion (K-NN)	IIT Kanpur (800 faces, 800 palms, 400 individuals)	Feature points are not fixed	97.5
Feature Level Fusion (Correlation)	IIT Kanpur (800 faces, 800 palms, 400 individuals)	Feature points are not fixed	98.75
Feature Level Fusion (K-NN)	Chimeric (160 faces, 160 palms, 40 individuals)	Feature points are not fixed	99.25
Feature Level Fusion (Correlation)	Chimeric (160 faces, 160 palms, 40 individuals)	Feature points are not fixed	99.5

5.3 Comparison with Other Technique

The proposed fusion of face and palmprint is compared with a multibiometrics system [19] where the features of face and hand evidences are fused. In the proposed fusion, SIFT features are extracted from face and palmprint and on these feature points, isomorphic graphs are drawn. These isomorphic representations are fused in terms of matched points found on isomorphic subgraphs. On the other hand, in [19] local facial features, such as eyes, mouth and nose features are localized using point distribution model and active shape models. Similarly, same methodology is applied to find some distinctive points on hand geometry. Gabor filter is applied to face image and feature vector is constructed by extracting the key points using active shape models. Similarly the hand feature vector is constructed. To verify the identity of users, Support Vector Machine is used. The technique presented in [19] is tested on a multibiometrics database which contains 480 face images and 120 hand images of 30 peoples. 16 faces and 4 hand images are taken from each person. Two experiments are conducted on the entire database. In the first experiment, features of 12 faces and 2 hands are fused for training and for testing, feature of 4 faces and remaining 2 hands are fused. The system is trained on 12 feature vectors which contain information about face and hand geometry of each individual. One SVM is trained on each individual. In this experiment numbers of hand features are fixed to every combination of features, where the number of features for eyes, nose and mouth are changing in every combination. The best recognition accuracy obtained from the first experiment is 99.23%. In the second experiment features of 12 faces and 3 hands are fused. This combination achieves best average accuracy while the system is trained with SVM. The best average accuracy obtained by the feature vector which contains Gabor features of 8 eye points, 4 nose points and 9 hand geometry. Table 3 shows the best average accuracy of different combinations of feature points. The best average recognition accuracy obtained from the second experiment is 99.43%.

The proposed approach shows the best recognition accuracies (RR) on IIT Kanpur and chimeric databases. Test on IIT Kanpur reveals 98.75% and 97.5% accuracies under normalized correlation and K-NN distance metrics respectively. In case of the chimeric database, they are 99.5% and 99.25%. The accuracies of the proposed approach are better than that of the approach in [19]. Since, the number of invariant features on both the face and palmprint images is not fixed, the performance shows outmost level of robust system. However, the fusion approach in [19] takes fixed number of features obtained from eyes, mouth and nose. And some distinctive features are determined from hand geometry. Number of SIFT feature points in the proposed fusion is changed dynamically and the combination of subgraph isomorphism and SIFT descriptor exhibits robustness of the fusion system. The system in [19] shows certain variations in selection of local feature points and it also shows good accuracies. However, due to fixed number of feature points and number of less feature points exhibit robustness to some extent. In the proposed fusion the whole face is used for feature extraction while the fusion approach in [19] uses the local features only.

6 Conclusion

This paper has proposed a feature level fusion approach of face and palmprint biometrics using invariant SIFT descriptor and isomorphic graph representation. The performance of feature level fusion has verified by two distance metrics namely, K-NN and normalized correlation metrics. Normalized correlation metric is found to be superior to K-NN metric for all the proposed methods. The proposed feature fusion has evaluated with two different multibiometrics databases and a comparative study has been presented with another fusion approach of different paradigm.

References

1. Kisku, D.R., Gupta, P., Sing, J.K.: Feature Level Fusion of Biometric Cues: Human Identification with Doddington's Caricature. In: International Conference on Security Technology, CCIS, pp. 157–164. Springer, Heidelberg (2009)
2. Rattani, A., Kisku, D.R., Bicego, M., Tistarelli, M.: Feature Level Fusion of Face and Fingerprint Biometrics. In: 1st IEEE International Conference on Biometrics, Theory, Applications and Systems, pp. 1–6. IEEE Press, Los Alamitos (2007)

3. Yao, Y.-F., Jing, X.-Y., Wong, H.-S.: Face and Palmprint Feature Level Fusion for Single Sample Biometrics Recognition. Neurocomputing 70(7), 1582–1586 (2007)
4. Ross, A., Govindarajan, R.: Feature Level Fusion using Hand and Face Biometrics. In: SPIE Conference on Biometric Technology for Human Identification II, pp. 196–204 (2005)
5. Fu, Y., Ma, Z., Qi, M., Li, J., Li, X., Lu, Y.: A Novel User-specific Face and Palmprint Feature Level Fusion. In: 2nd International Symposium on Intelligent Information Technology Application, pp. 296–300 (2008)
6. Yan, Y., Zhang, Y.-J.: Multimodal Biometrics Fusion using Correlation Filter Bank. In: International Conference on Pattern Recognition, pp. 1–4. IEEE Press, Los Alamitos (2008)
7. Kisku, D.R., Rattani, A., Grosso, E., Tistarelli, M.: Face Identification by SIFT-based Complete Graph Topology. In: 5th IEEE International Workshop on Automatic Identification Advanced Technologies, pp. 63–68. IEEE Press, Los Alamitos (2007)
8. Jain, A.K., Feng, J.: Latent Palmprint Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(6), 1032–1047 (2009)
9. Whitney, H.: Congruent Graphs and the Connectivity of Graphs. Am. J. Math. 54, 160–168 (1932)
10. Zhang, O., Couloigner, I.: A New and Efficient K-medoid Algorithm for Spatial Clustering. In: International Conference on Computational Science and Its Applications, pp. 181–189 (2005)
11. Lowe, D.G.: Object Recognition from Local Scale-invariant Features. In: International Conference on Computer Vision, pp. 1150–1157 (1999)
12. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, p. 635
13. Horiuchi, T.: Colorization Algorithm using Probabilistic Relaxation. Image and Vision Computing 22(3), 197–202 (2004)
14. Dasarathy, B.V.: Nearest neighbor (NN) norms: NN Pattern Classification Techniques (1991)
15. Kumar, A., Wong, D.C.M., Shen, H.C., Jain, A.K.: Personal Verification using Palmprint and Hand Geometry Biometric. In: 4th International Conference on Audio- and Video-Based Biometric Authentication, pp. 668–675 (2003)
16. Kisku, D.R., Tistarelli, M., Sing, J.K., Gupta, P.: Face Recognition by Fusion of Local and Global Matching Scores using DS Theory: An Evaluation with Uni-classifier and Multi-classifier Paradigm. In: IEEE Computer Vision and Pattern Recognition (CVPR) Workshop on Biometrics, pp. 60–65 (2009)
17. Ribarí, C.S., Fratríc, I.: A Biometric Identification System Based on Eigenpalm and Eigenfinger Features. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(11), 1698–1709 (2005)
18. Dubes, R., Jain, A.K.: Clustering Techniques: The User's Dilemma. Pattern Recognition 8(4), 247–260 (1976)
19. Rokita, J., Krzyzak, A., Suen, C.Y.: Multimodal Biometrics by Face and Hand Images taken by a Cell Phone Camera. International Journal of Pattern Recognition and Artificial Intelligence 22(3), 411–429 (2008)
20. Rattani, A., Kisku, D.R., Logario, A., Tistarelli, M.: Facial Template Synthesis based on SIFT Features. In: 5th IEEE International Workshop on Automatic Identification Advanced Technologies, pp. 69–73 (2007)
21. Zhang, D., Kong, W.-K., You, J., Wong, M.: On-line Palmprint Identification. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(9), 1041–1050 (2003)

Scale and Rotation Invariant Detection of Singular Patterns in Vector Flow Fields

Wei Liu and Eraldo Ribeiro

Computer Vision and Bio-Inspired Computing Laboratory
Department of Computer Sciences
Florida Institute of Technology
Melbourne, FL 32901, USA
{lwei,eribeiro}@fit.edu

Abstract. We present a method for detecting and describing features in vector flow fields. Our method models flow fields locally using a linear combination of complex monomials. These monomials form an orthogonal basis for analytic flows with respect to a correlation-based inner-product. We investigate the invariance properties of the coefficients of the approximation polynomials under both rotation and scaling operators. We then propose a descriptor for local flow patterns, and developed a method for comparing them invariantly against rigid transformations. Additionally, we propose a SIFT-like detector that can automatically detect singular flow patterns at different scales and orientations. Promising detection results are obtained on different fluid flow data.

1 Introduction

Detecting patterns in vector flow fields is key to many computer vision and engineering applications including texture analysis [16], fingerprint classification [6,14], and fluid dynamics [4,17]. In principle, singular flow-pattern detection is similar to the interest point detection problem in scalar images [13]. However, the number of flow-field descriptor approaches in the computer vision literature is relatively limited. In this paper, we introduce a novel scale-rotation invariant framework for detecting singular patterns in vector flow data.

Vector field data usually originate from continuous physical processes such as motion and dynamic textures. As a result, model-based approaches for singular pattern detection are common in the literature. For example, template-matching approaches using correlation [17] or filtering operations [14] are generally robust. However, pattern detected by these approaches are often restricted to the template's size and shape. Another class of singular-pattern detection methods are based on locally-affine flow-field models [16]. An extension to a nonlinear flow model was proposed by Ford *et al.* [8], and most recently Kihl *et al.* [11] improved it further to detect multi-scale singular points. Finally, flow fields can also be represented using complex functions. For instance, Fan *et al.* [6] used the complex zero-pole model to detect singular points in fingerprint images. An earlier work by Nogawa *et al.* [15] modeled singular patterns based on Cauchy's residue

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 522–531, 2010.

theorem. Nevertheless, residue calculation can be sensitive to noise. Corpetti *et al.* [4] detected singular-flow patterns as the maxima of complex potential and streamline functions that were obtained from flow fields' irrotational and solenoidal components. Corpetti's detected singular patterns were quite general as they did not need to contain a center vanishing point.

In this paper, we propose a novel framework for the detection and description of singular patterns in vector fields under rigid transformations (i.e., rotation and scale invariant). We commence by approximating the flow field locally using a linear combination of complex analytic basis functions (Section 2). We use the approximation coefficients as a flow descriptor. Our selected set of complex basis functions can be shown to be eigenfunctions of the rotation operator. This observation allows us to define a concept equivalent to a flow pattern's principle orientations by aligning it to the analytic bases. We will show that scaling a flow field corresponds to scaling our descriptor. By aligning descriptors using the estimated principle orientations, and normalizing them in scale, flow patterns can be directly compared (Section 3). Finally, we introduce a multi-scale singular pattern detector (Section 4). As in [4], we are able to detect singular patterns in a broader sense than the commonly used vanishing singular points. Our experimental results (Section 5) demonstrate the effectiveness of our descriptor by both detecting and clustering singular patterns on various flow field sequences.

2 Higher-Order Model of Flow Field

We begin by representing a 2-D vector-flow field as a complex-valued function $F(z)$ defined on a finite domain $\Omega \subset \mathbb{C}$. Locally, a flow field can be represented by an analytic function centered at $z_0 \in \mathbb{C}$, i.e., $f(z) \approx F(z + z_0)$.[1] The Taylor expansion of $f(z)$ about the origin (i.e., $z_0 = 0$) can be written as a linear combination of complex (orthogonal) basis functions $\phi_k(z)$ as follows:

$$f\,(z) = \sum_{k=0}^{N} a_k \phi_k(z) + R_N(z), \tag{1}$$

where $a_k = \frac{f^{(k)}(0)}{k!}$ are the coefficients, and $R_N(z)$ is the residue. Here, $f^{(k)}(0)$ is the k-th derivative of f evaluated at $z_0 = 0$. There are number of choices of polynomial bases $\phi_k(z)$ that are equivalent from both the functional analysis and approximation theory viewpoints, e.g., complex-domain Zernike polynomials [10] and real-domain Legendre polynomials [11]. Our goal is to approximate flow fields locally. While this goal can be accomplished using linear models based on real-domain basis functions [11,16], we believe that complex functions are valuable bases to model smooth natural motions [10,15,4]. Additionally, complex bases are usually more compact than their real orthogonal counterparts.

[1] The analytic assumption was also used in [15,4]. While theoretically some linear flow fields are not analytic, they can be considered less physically relevant [4].

It is worth pointing out that both the orthogonality condition and basis-function projection depend on the choice of inner product in the analytic functions space $A(\Omega)$ on Ω. Using the standard inner product defined for complex functions in $\mathbf{L}^2$ [5] results in complex numbers, making projection calculations difficult. Instead, we adopt the following alternative inner product:

$$\langle f(z), g(z)\rangle = \int_{\mathbb{C}} f(z) \cdot g(z)\, dz, \tag{2}$$

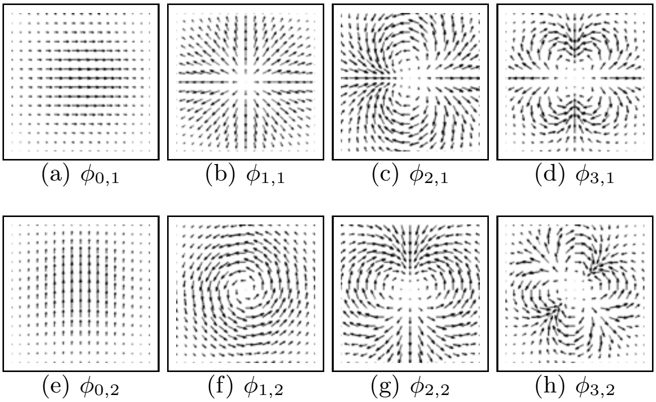
where $\cdot$ is the standard inner product on $\mathbb{C}$ (i.e., dot product between two complex numbers). Equation 2 satisfies the three inner product axioms [5]: symmetry, linearity, and positive-definiteness, and it can be used to project flow field $f(z)$ onto the basis function $\phi_k(z)$, with real-domain projection coefficients given by $a_k = \frac{\langle f(z), \phi_k(z)\rangle}{\langle \phi_k(z), \phi_k(z)\rangle}$. Furthermore, we can re-write Equation 2 as:

$$\langle f(z), g(z)\rangle = (F \otimes g)(z_0) = \int_{\mathbb{C}} F(z+z_0) \cdot g(z)\, dz, \tag{3}$$

which is similar to the cross-correlation operator used in [17], and can be implemented efficiently using the Fast Fourier Transform (FFT).

In this paper, we use the complex-domain monomials $\{z^k\}_{k=1}^N$ as basis functions $\phi_k(z)$. We can show that iz^k belongs to the same basis formed by z^k, and that the basis is complete. Intuitively, iz^k can be thought as a counterclockwise 90-degree rotation of the vectors in z^k. Without affecting the orthogonality of z^k and iz^k, we control the basis' local support size by weighting the basis with a zero-mean Gaussian function $w_\sigma(y)$. Our basis flows can then be written as:

$$\phi_{k,1}(z) = \frac{z^k w_\sigma(z)}{\|z^k w_\sigma(z)\|} \qquad \text{and} \qquad \phi_{k,2}(z) = \frac{iz^k w_\sigma(z)}{\|iz^k w_\sigma(z)\|}, \tag{4}$$


(a) $\phi_{0,1}$ (b) $\phi_{1,1}$ (c) $\phi_{2,1}$ (d) $\phi_{3,1}$

(e) $\phi_{0,2}$ (f) $\phi_{1,2}$ (g) $\phi_{2,2}$ (h) $\phi_{3,2}$

Fig. 1. Basis flows $\phi_{k,i}$, $k = 0, \ldots, 3$ and $i = 1, 2$. Row 1: polynomials derived from z^k. Row 2: polynomials derived from iz^k. Increasing k produce higher-order fluctuations.

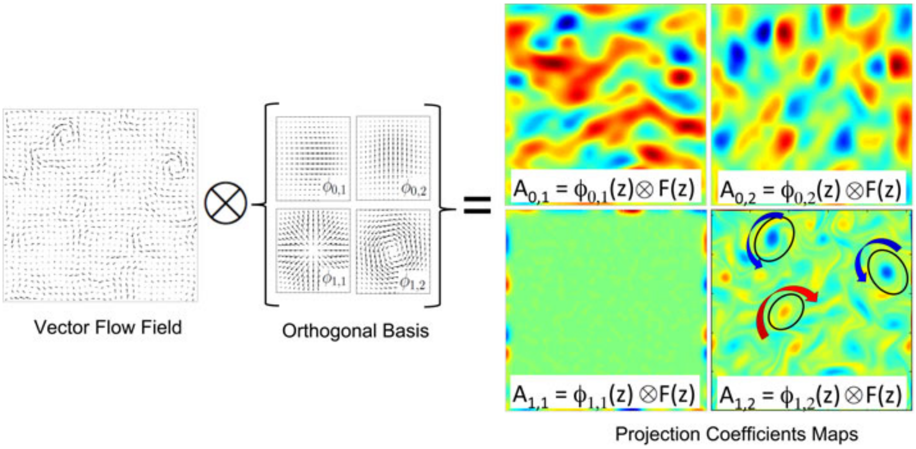

Vector Flow Field
Orthogonal Basis
$\phi_{0,1}$
$\phi_{0,2}$
$\phi_{1,1}$
$\phi_{1,2}$
$A_{0,1} = \phi_{0,1}(z) \otimes F(z)$
$A_{0,2} = \phi_{0,2}(z) \otimes F(z)$
$A_{1,1} = \phi_{1,1}(z) \otimes F(z)$
$A_{1,2} = \phi_{1,2}(z) \otimes F(z)$
Projection Coefficients Maps

Fig. 2. Cross-correlation between the flow field and the first four bases $\phi_{k,1}(z)$. Map $A_{1,1}$ indicates a divergence-free flow field. Peaks in $A_{1,2}$ indicate vortices. Blue: matching orientation between filter and flow data. Red: reverse orientation.

where $\|\phi\|^2 = \langle \phi, \phi \rangle$. The orthonormal basis $\phi_{k,i}$ for $k = 0, 1, 2, 3$ are shown in Figure 1. Using (1), the N-th order flow field approximation at $p \in \Omega \subset \mathbb{C}$ is:

$$F(z + z_0) \approx f(z) = \sum_{k=0}^{N} \left[a_{k,1}\phi_{k,1}(z) + a_{k,2}\phi_{k,2}(z)\right], \tag{5}$$

where $a_{k,i} = \langle f(z), \phi_{k,i}(z) \rangle$, for $k = 1, \ldots, N$, and $i = 1, 2$. The approximation produces $2(N + 1)$ real coefficients $a^p = a^p_{0,1}, a^p_{0,2}, \ldots, a^p_{N,1}, a^p_{N,2}$ for location p. According to (3), the coefficients are local values of the cross-correlation between $F(z)$ and $\phi_{k,i}(z)$. Figure 2 shows the correlation of the first two basis pairs with a turbulent flow, i.e., $A_{k,1} = F(z) \otimes \phi_{k,1}(z)$ and $A_{k,2} = F(z) \otimes \phi_{k,2}(z)$, $k = 0, 1$.

3 Flow Field Descriptor

In the previous section, a local approximation of local flow fields was presented. We will now show how the projection coefficients can be used to derive descriptors that are invariant to both rotation and scaling transformations.

The rotation operator. Let us consider the flow-field expansion given by (5). The rotation operator $\Gamma_\theta(\cdot)$ rotates the flow $f(z)$ by an angle θ as follows:

$$\begin{aligned}\Gamma_\theta\left(f(z)\right) = e^{-\theta i} f(ze^{\theta i}) &= e^{-\theta i} \sum_{k=0}^{N} \left[a_{k,1}\phi_{k,1}(ze^{\theta i}) + a_{k,2}\phi_{k,2}(ze^{\theta i})\right] \\ &= \sum_{k=0}^{N} \left[a_{k,1}\Gamma_\theta\left(\phi_{k,1}(z)\right) + a_{k,2}\Gamma_\theta\left(\phi_{k,2}(z)\right)\right].\end{aligned} \tag{6}$$

Here, $e^{-\theta i}$ is the contravariant factor to ensure coordinate invariance of the vector field [1]. $\mathit{\Gamma}_\theta$ is linear on the analytic function space $\mathbb{A}^n$. Furthermore, our choice of basis monomials, z^k and iz^k, are eigenfunctions for $\mathit{\Gamma}_\theta$, i.e.,

$$\mathit{\Gamma}_\theta(z^k) = e^{(k-1)\theta i}\, z^k \qquad \text{and} \qquad \mathit{\Gamma}_\theta(iz^k) = e^{(k-1)\theta i}\, iz^k, \tag{7}$$

with eigenvalues equal to $e^{(k-1)\theta i}$. The bases' Gaussian weighting and the normalizing constant in (4) are rotation invariant so they were dropped. By plugging (7) into (6) and re-arranging the basis monomial terms, we obtain:

$$\begin{aligned} a'_{k,1}(\theta) &= \cos\left[(k-1)\theta\right] a_{k,1} - \sin\left[(k-1)\theta\right] a_{k,2} \\ a'_{k,2}(\theta) &= \sin\left[(k-1)\theta\right] a_{k,1} + \cos\left[(k-1)\theta\right] a_{k,2}, \end{aligned} \tag{8}$$

where $a_{k,i}$ and $a'_{k,i}$ are the coefficients for the original and rotated flow fields, respectively. Equation 8 shows that rotating a flow field also rotates their projection coefficients. Our goal is to compare flows by rotating the coefficients to a standard orientation. However, the rotation angle is unknown. We solve this problem by finding the angle that maximizes the alignment between the local flow and a subset of our eigenfunctions that are not rotationally symmetric (i.e., except z and iz), and calculate θ that maximizes the inner-product projection:

$$\widetilde{\theta} = \arg\max_{\theta} \sum a'_{k,1}(\theta)\,. \tag{9}$$

The above maximization of a trigonometric polynomial function can be solved using standard optimization algorithms (e.g., Gauss-Newton method). Computationally, rotating the coefficients is far more efficient than rotating the flow field itself. We call the values of θ at these local maxima the *Principle Orientations*. Once these directions are at hand, we can compare two flow fields, $f_p(z)$ and $f_q(z)$, by defining a distance between them. We use the minimum Euclidean distance between their rotated coefficients defined as follows:

$$d(f_p, f_q) = \min_{i,j} \|\mathit{\Gamma}_{\theta_i} a_p - \mathit{\Gamma}_{\theta_j} a_q\|. \tag{10}$$

Vector fields' directional nature generate multiple principle orientations, and Equation 9 has at most $2N$ roots [9]. Rather than finding the "best" orientation, we accept all principle orientations for which this equation exceeds a threshold.

The scaling operator. Let us now consider the scaling operator $\mathit{\Psi}_s(.)$, $s > 0$ applied on the weighted basis flow defined in (4). This operator is also linear, and scaling effects are fully defined on the basis functions $\phi_{k,i}$ as follows:

$$\begin{aligned} \mathit{\Psi}_s(\phi_{k,1}(z)) = s\,\phi_{k,1}(s^{-1}z) &= s\,\frac{(s^{-k})}{|(s^{-k})|}\,\frac{z^k w_\sigma(s^{-1}z)}{\|z^k w_\sigma(s^{-1}z)\|} \\ &= s\,\frac{z^k w_\sigma(s^{-1}z)}{\|z^k w_\sigma(s^{-1}z)\|} = s\,\frac{z^k w_{s\sigma}(z)}{\|z^k w_{s\sigma}(z)\|}. \end{aligned} \tag{11}$$

Therefore, scaled bases can be obtained by scaling the variance of the Gaussian weighting function, and then multiplying them by s. The relationship holds for both z^k and iz^k bases. Next, we use these ideas to develop a method for detecting interest flow patterns under scaling and rotation transformations.

4 Detection of Singular Patterns

Singular points (or critical points) in vector fields can be defined as locations where the flow field vanishes [11,16,7], i.e., $F(z) = 0$. If we consider the expansion in Equation 5, then $F(z) = 0$ implies $a_{0,1} = a_{0,2} = 0$. As a result, a local flow pattern containing a singular point at the center can be linearly approximated by $\phi_{k,i}$ with $k \geq 1$. We will name $\phi_{k,i}, k \geq 1$ the *singular basis*, and will assume that the flow field's linear expansion can be separated into two components: the background constant flow expanded by $\phi_{0,1}, \phi_{0,2}$, and a singular component expanded by the singular basis. The constant flow is similar to the laminar component mentioned in [4]. We define a singular point as maxima of the flow field energy projected onto the singular basis. The singular energy function is defined as the squared sum of projection coefficients on the singular basis, i.e., $E_{sig}(z) = \sum_{k=1}^{N} \left(\|a_{k,1}\|^2 + \|a_{k,2}\|^2\right)$. As in [4], the separation of background constant flow makes our definition of singular patterns more general than the singular points defined in [11,16,7], since a flow field may not have any vanishing points when a background constant flow or a laminar flow exists.

Comparing flows of similar sizes can be achieved by using Equation 10. We now look into the case of detecting singular patterns at multiple scales. Here, we will approach the multiple-scale problem in a similar way as done in scale-space theory for scalar images. We begin by applying a Gaussian smoothing to the vector field followed by a down-sampling operation [13]. In the case of vector fields, the Gaussian smoothing step might actually destroy singular points [7]. Using the properties described in Section 3, we keep the flow field unchanged,

Algorithm 1. Scale-Rotation Invariant Singular Flow Pattern Detection

```
1 Given an input flow $F(z)$, create octaves of $F^o(z), o = 0, 1, \ldots, N$ by
  down-sampling $F(z)$ by half, i.e., $F^{o+1}(z) \leftarrow \downarrow_2 F^o(z)$.
2 Create multiscale bases $\phi^s_{k,i}, s = 0, 1, \ldots, M$ by increasing their variance by
  a step $\Delta\sigma$ (e.g., $\Delta\sigma = 2^{1/M}$). As in SIFT [13], we generate $M + 3$ images to
  cover a complete octave, with starting scale $\sigma_0 = 1.6$.
3 Calculate the coefficients in each octave using cross-correlation, i.e.,
  $A^{o,s}_{k,1} \leftarrow F^o(z) \otimes \phi^s_{k,1}(z)$ and $A^{o,s}_{k,2} \leftarrow F^o(z) \otimes \phi^s_{k,2}(z)$, $k = 0, 1$.
4 Calculate the singular energy $E^o_{sig}(z, s)$ at each octave.
  $E^o_{sig}(z, s) \leftarrow \sum_{k=1}^{N} \left(\|a^s_{k,1}(z)\|^2 + \|a^s_{k,2}(z)\|^2\right)$.
5 Detect the singular points at spatial position $(x, y)$ and scale $s$ that locally
  maximize the singular energy $E^o_{sig}(z, s)$.
6 Calculate descriptor and principle orientations (Equation 9)
  at detected positions.
```

and instead vary the scale of the basis function. Scaling the basis function only involves changing the variance parameter of the Gaussian weighting function, and it does not destroy the singular points. However, increasing the basis flow size increases the computation due to the correlation operation in Equation 3.

To address these problems, we adopt a hybrid method for multiscale singular-pattern detection. Similarly to the SIFT descriptor [13], we divide the scale space into octaves using Gaussian smoothing and down-sampling. However, scaling is applied to the basis flows within each octave. Singular pattern candidate scales are selected as extrema of singular energy E_{sig} along both the scale and spatial dimensions. Algorithm 1 summarizes the detection process.

5 Experiments

We tested our detector on sequences from European FLUID Project [3], and satellite imagery obtained from the SSEC Data Center[2]. Additionally, we tested the flow descriptor by automatically clustering singular patterns of varying scale and orientation that were extracted from the JHU Turbulence dataset [12].

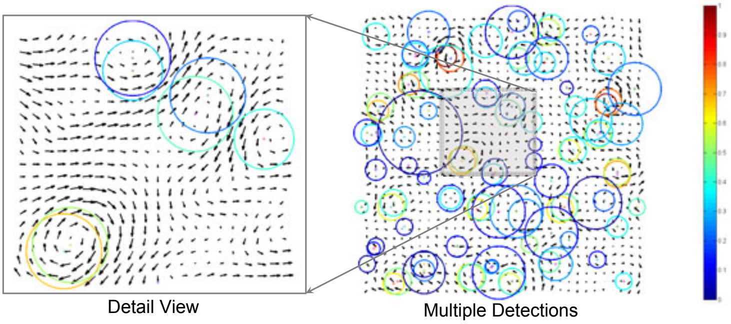


Fig. 3. Right: detected patterns. Color indicates the relative log magnitude of singular energy. Vortices are the strongest patterns; Left: detail view of detected patterns.

Detection on FLUID sequences. Detected singular patterns from a FLUID sequence are shown in Figure 3. The patterns' singular energy was color-mapped for visualization clarity. This dataset contains sourceless vector fields, and most singular patterns resemble vortices appearing at multiple scales. Our method detected all vortices. Elongated-shaped vortices were detected in pairs. In these cases, some detections could have been discarded based on their singular energy.

Detection on satellite images. In this experiment, we extracted velocity field data from SSEC satellite image sequences using the CLG optical-flow algorithm [2]. CLG produced fairly good estimation results considering that accurate

[2] http://www.ssec.wisc.edu/data/us_comp/

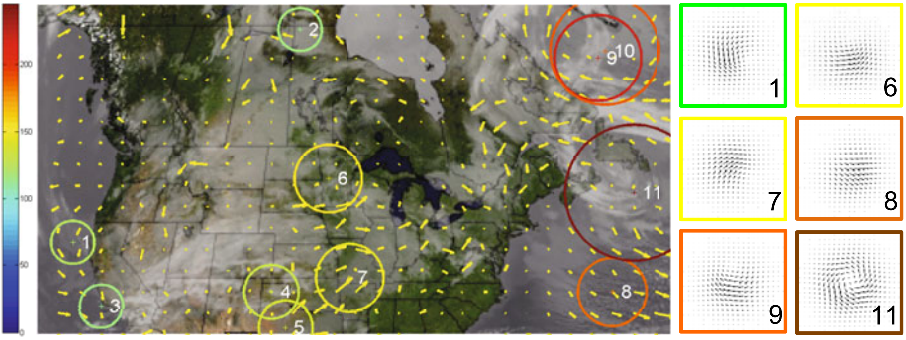


Fig. 4. Singular pattern detection in satellite image sequence. Estimated flow field is downsampled for visualization. Strong patterns to the north-east (9,10,11) corresponds to vortices. South-east singular pattern (8) corresponds to sudden clouds divergence.

fluid-motion estimation is not our method's main focus. Detection results produced by our detector on motion clouds are shown in Figure 4. The figure shows a satellite image of a U.S. weather system on February 20th, 2010. For better visualization, singular patterns smaller than 20 pixels in diameter were removed. On the northeast corner, large vortices were detected. On the southeast corner, a strong singular pattern corresponds to clouds disappearance and divergence. Most detected patterns are consistent with cloud motion changes.

Detecting and Clustering. In this experiment, we clustered singular patterns detected on the JHU 3-D Turbulence dataset. Here, we selected 2-D slices that were perpendicular to the flow's convecting direction. For better visualization, we created two groups of detected patterns according to their similarity to vortices and sources (or sinks). We did that by examining whether the singular energy was concentrated on the basis functions $\phi_{1,1}, \phi_{1,2}$. If $\|a_{1,1}\|^2 + \|a_{2,2}\|^2$ consisted of more than 60% of the total singular energy, then we labeled the singular pattern as *symmetric*, otherwise, we call it *asymmetric*.

We then scaled and aligned the features. For patterns having multiple principle orientations, we generated multiple aligned copies, and created four groups using k-means. Clusters for symmetric features are shown in Figure 5 (top), while clusters of asymmetric features are shown in Figure 5 (bottom). Symmetric patterns mostly corresponded to vortices in both directions, sources, and swirls. Due to the flow's divergent nature, few sinks were detected, and no sink clusters were obtained. Asymmetric patterns mostly correspond to vortices skewed by a background laminar. Clusters were distinguished by their rotation direction, and their divergence or convergence. Most patterns in this group did not have a center vanishing point, yet they still exhibited interesting sudden flow field changes. This suggests the generality of our singular pattern definition.

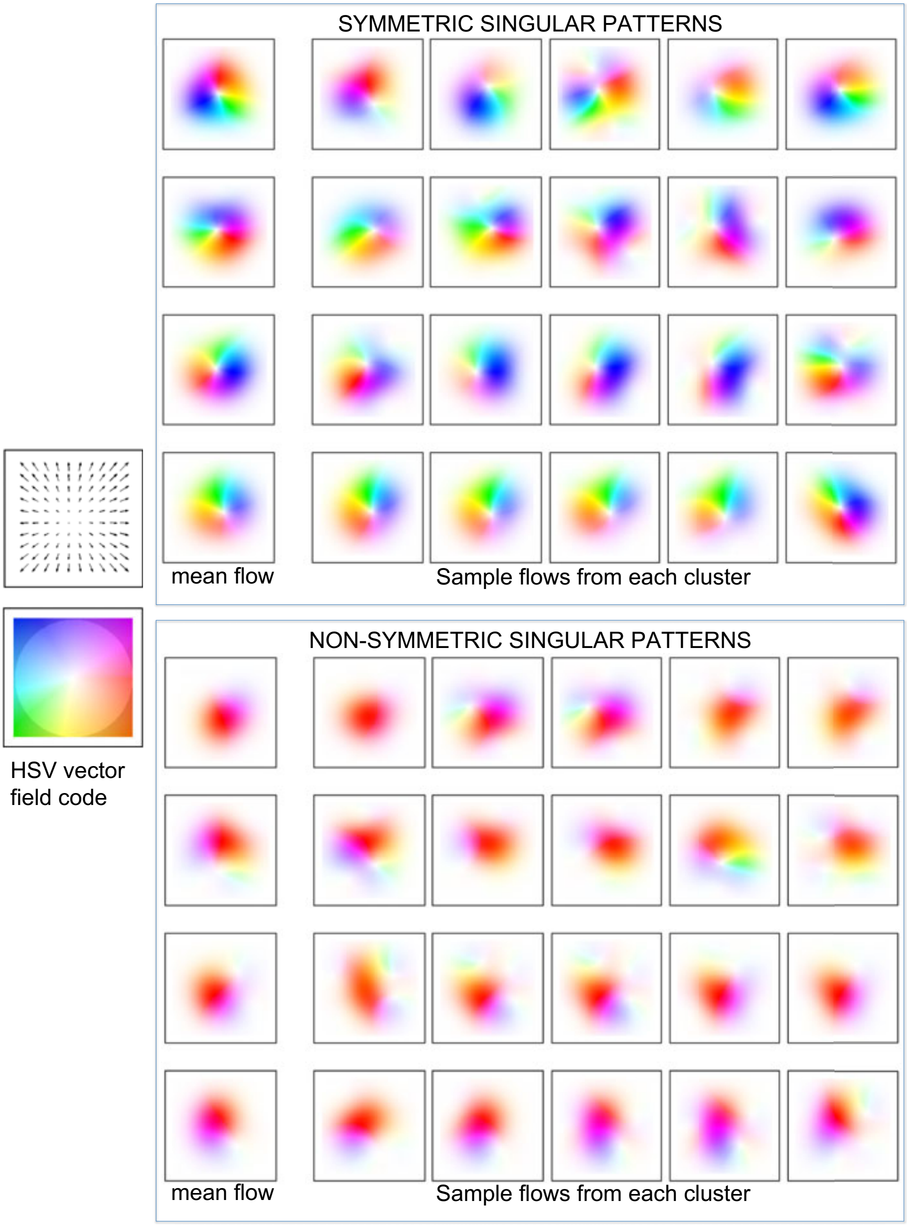

SYMMETRIC SINGULAR PATTERNS
mean flow
Sample flows from each cluster
HSV vector field code
NON-SYMMETRIC SINGULAR PATTERNS
mean flow
Sample flows from each cluster

Fig. 5. Clusters of symmetric and asymmetric singular patterns detected on the JHU 3-D Turbulence dataset. Each row displays cluster means and sample flows.

6 Conclusion

We proposed a flow-field descriptor based on coefficients of a local flow field approximation. Based on this descriptor, we designed a SIFT-like detector for singular patterns that is invariant to rigid transformations. The detector was tested on both synthetic and real fluid flows. Future work includes an extension to 3-D flow fields and exploring new applications.

References

1. Bronshtein, I., Semendyayev, K., Musiol, G., Muehlig, H.: Handbook of mathematics. Springer, Berlin (1997)
2. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. Int. J. Comp. Vis. 61(3), 211–231 (2005)
3. Carlier, J.: Second set of fluid mechanics image sequences. In: European Project 'Fluid image analysis and description, FLUID (2005), http://www.fluid.irisa.fr/
4. Corpetti, T., Mémin, E., Pérez, P.: Extraction of singular points from dense motion fields: An analytic approach. J. Math. Imaging Vis. 19(3), 175–198 (2003)
5. Davies, B.: Integral Transforms and Their Applications. Springer, Heidelberg (2002)
6. Fan, L., Wang, S., Wang, H., Guo, T.: Singular points detection based on zero-pole model in fingerprint images. Trans. Patt. Anal. Mach. Intell. 30(6), 929–940 (2008)
7. Florack, L.: Scale-space theories for scalar and vector images. In: Scale-Space 2001, London, UK, 2001, pp. 193–204. Springer, London (2001)
8. Ford, R.M., Strickland, R.N., Thomas, B.A.: Image models for 2-D flow visualization and compression. Graph. Models Image Process. 56(1), 75–93 (1994)
9. Forray, M.J.: Approximation Theory and Methods. Cambridge Univ. Press, Cambridge (1981)
10. Hoey, J., Little, J.J.: Bayesian clustering of optical flow fields. ICCV 2, 1086 (2003)
11. Kihl, O., Tremblais, B., Augereau, B.: Multivariate orthogonal polynomials to extract singular points. In: ICIP, pp. 857–860 (2008)
12. Li, Y., Perlman, E., Wan, M., Yang, Y., Meneveau, C., Burns, R., Chen, S., Szalay, A., Eyink, G.: A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence. Journal of Turbulence 9(31), 1–29 (2008)
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60(2), 91–110 (2004)
14. Nilsson, K., Bigun, J.: Localization of corresponding points in fingerprints by complex filtering. Pattern Recogn. Lett. 24(13), 2135–2144 (2003)
15. Nogawa, H., Nakajima, Y., Sato, Y., Tamura, S.: Acquisition of symbolic description from flow fields: a new approach based on a fluid model. IEEE Trans. Patt. Anal. Mach. Intell. 19(1), 58–63 (1997)
16. Rao, A.R., Jain, R.C.: Computerized flow field analysis: Oriented texture fields. IEEE Trans. Pattern Anal. Mach. Intell. 14(7), 693–709 (1992)
17. Schlemmer, M., Heringer, M., Morr, F., Hotz, I., Hering-Bertram, M., Garth, C., Kollmann, W., Hamann, B., Hagen, H.: Moment invariants for the analysis of 2D flow fields. IEEE Trans. on Vis. and Comp. Graph 13(6), 1743–1750 (2007)

Using K-NN SVMs for Performance Improvement and Comparison to K-Highest Lagrange Multipliers Selection

Sedat Ozer[1], Chi Hau Chen[2], and Imam Samil Yetik[3]

[1] Electrical & Computer Eng. Dept, Rutgers University, New Brunswick, NJ, USA
sozer@umassd.edu
[2] Electrical & Computer Eng. Dept, University of Massachusetts,
Dartmouth, N. Dartmouth, MA, USA
cchen@umassd.edu
[3] Electrical & Computer Eng. Dept, Illinois Institute of Technology, Chicago, IL, USA
yetik@iit.edu

Abstract. Support Vector Machines (SVM) can perform very well on noise free data sets and can usually achieve good classification accuracies when the data is noisy. However, because of the overfitting problem, the accuracy decreases if the SVM is modeled improperly or if the data is excessively noisy or nonlinear. For SVM, most of the misclassification occurs when the test data lies closer to the decision boundary. Therefore in this paper, we investigate the effect of Support Vectors found by SVM, and their effect on the decision when used with the Gaussian kernel. Based on the discussion results we also propose a new technique to improve the performance of SVM by creating smaller clusters along the decision boundary in the higher dimensional space. In this way we reduce the overfitting problem that occurs because of the model selection or the noise effect. As an alternative SVM tuning method, we also propose using only K highest Lagrange multipliers to summarize the decision boundary instead of the whole support vectors and compare the performances. Thus with test results, we show that the number of Support Vectors can be decreased further by using only a fraction of the support vectors found at the training step as a post-processing method.

Keywords: Support Vector Machine, KNN SVM, Post-processing, Support Vector Reduction.

1 Introduction

Support Vector Machine (SVM) is a well known learning algorithm that has been widely used in many applications including classification, estimation and tracking as in [1], [2], [3] and [4]. SVM finds the closest data vectors called support vectors (SV), to the decision boundary in the training set and it classifies a given new test vector by using only these closest data vectors [5],[6].

In order to find the optimal nonlinear decision boundary, SVM uses kernel functions, along the optimization step to find the optimal hyperparameters, [5]. However, in practice, the iterative techniques used at the optimization step, can also affect the classification accuracy of SVM within the margin.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 532–539, 2010.

Besides the SVM algorithm, the K nearest neighbor (KNN) technique is another well known learning technique and being used in several pattern recognition applications as in [7]. There have been some previous studies where KNN technique was combined with SVM as in [8], [9] and [10].

The combination of these two techniques by switching between them could perform better only for certain cases in which the new data is close to the decision boundary. In [8], the KNN algorithm is applied directly onto those data vectors which are within the margin. However, in [8], it is claimed that previously proposed KSVM cannot reduce the generalization error.

Also, in studies such as in [9] and in [10], KNN idea is used in a different way combined with SVs. In [9], authors study the effects of using K nearest SVs by focusing on query time rather than improving the accuracy. They propose using a varying K value for each test data till they reach to a certain threshold. Thus they search for an appropriate K value for each given test data. In [10], instead of training the SVM only once, the authors propose using the K nearest data values to train SVM separately for each given test data.

Both of the papers [9] and [10] uses the KNN idea in a different way, while [9] requires to search for an appropriate K value for each single test data, the authors of [10] require to train SVM for each given new test data. Moreover, although these papers do not clearly indicate in them, they can perform better when used with Gaussian kernel because of the Gaussian kernel function's shape.

In this study, we propose more naive yet efficient way of using KNN SVs when used with Gaussian kernels for a given dataset. We train the SVM only once and after that we require only one K value to be found. Our approach is applicable to all new data points regardless of their distance to the decision boundary. In this approach, we use the entire training data to find the SVs. However after this point, instead of using the all SVs that have been found on the training step, we propose to use only the K nearest SVs. Since the Gaussian kernel is also using the Euclidian distance, there is not much computational cost to find distances to each SVs.

The classification with SVM, besides its high accuracy, also provides sparseness which is another advantage of SVM, thus we do not need to save all the training data. Therefore, in this study, we also propose using only the K highest Lagrange multipliers (α) instead of all the nonzero Lagrange multipliers found at the training step of SVM. Section 4 tests and investigates if all the SVs found by the classifier, are necessary to classify the new data. Experimental results show that, even if the non-zero α values has closer value to each other, there can be some redundancy where we can reduce the SV number by choosing only the K highest SVs and corresponding α values.

Consequently, the SV number can be reduced by using the method presented on this paper for a similar performance. Besides, we also show that it is possible to increase the efficiency of the SVM by using only a fraction of SV numbers. Preliminary test results provide us interesting results about SVs which we discuss at the Section 5.

2 Support Vector Machine

SVM searches for the optimal decision boundary between two classes [5], [6]. Although SVM is mainly designed as a linear binary classifier, it is widely being used for nonlinear data efficiently as well, by the use of kernel functions [5].

SVM uses the following formula for the classification, for a given new data vector **x**:

$$f(\mathbf{x}) = \mathrm{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \right) \tag{1}$$

where α is the Lagrange multiplier for each SV that needs to be found in the training step, m the support vector number, b the biasing term, y the class labels, $K(\mathbf{x},\mathbf{x_i})$ the kernel function, and $\mathbf{x_i}$ are the support vectors. The parameters b and α_i need to be found in the training step. The Lagrange multipliers, α_i, can be found by maximizing the following equation:

$$w(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \tag{2}$$

subject to $\sum_{i=1}^{n} \alpha_i y_i = 0$ and $\alpha_i \geq 0$, where n is the training sample number.

Thus the $\mathbf{x_i}$ input vectors with nonzero α_i values, are called support vectors (SV). Although several kernel functions have been proposed to be used with SVMs, as in [5], [11] and [12], the kernel function used in this study is the Gaussian kernel which is defined as:

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2} \right) \tag{3}$$

where σ is the kernel parameter that needs to be found for a satisfactory classification performance.

3 The Proposed Method

K-NN SVM: If the Gaussian kernel is being used, then SVM can be considered as a binary clustering algorithm. However, in contrast to the other clustering algorithms, instead of finding the centroids of the clusters, SVM uses the edge information of the clusters where the two clusters are the closest to each other.

Our assumption in this study is that for a given new data vector, we do not need to use all the support vectors as in the traditional SVM. That is because the hyperplane can be more linear in some regions of the whole data space, and can be highly nonlinear in other regions. Therefore using only the K nearest support vectors within the same local region can increase the performance. Let us re-arrange the equation (1) as follows:

$$f(\mathbf{x}) = \mathrm{sgn}\left(b + \sum_{i=1}^{h} \alpha_i K(\mathbf{x}, \mathbf{x}_i) - \sum_{j=1}^{g} \alpha_j K(\mathbf{x}, \mathbf{x}_j) \right) \tag{4}$$

where h is the number of support vectors for the (+1) zone and similarly g is the number of the support vectors for the (-1) zone.

When the Gaussian kernel is being used for SVM, the Equation (4) simply becomes a weighted subtraction of α values with a biasing term b, treating the $K(\mathbf{x},\mathbf{x}_i)$ values as weights. Here the weights $K(\mathbf{x},\mathbf{x}_i)$ are mapped to a value based on the Euclidian distance between the new data and the support vector.

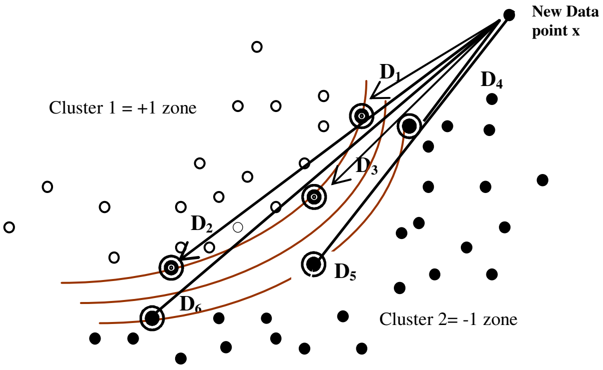


Fig. 1. Distances to all Support Vectors for a given new data

As shown in Equation (5), the Gaussian kernel maps the distance values between 0 and 1, where the closer distance is mapped to a higher value. Here, the kernel parameter σ decides after which value the mapping decays to 0 more faster. Thus, for a given test data vector, some of the α values in Equation (4) can vanish because of their weights goes to zero, then only the b value and the closest α values decide for the sign of the new test data. That means all SVs have some local effect on the whole decision boundary.

$$0 < K(\mathbf{x},\mathbf{x}_i) \leq 1 \tag{5}$$

As illustrated in Figure 1, for a given new data the distances to all support vectors are shown as D_1, D_2, D_3, D_4, D_5, D_6. For the classification of the new data point, D_2 and D_3 will be more effective than D_5 and D_6, as these distance values are smaller. This may yield an incorrect classification of the data. This situation can be more important as the test data gets closer to the decision boundary.

This can reduce the effect of the noise on forming the decision boundary. As the overfitting problem yields a complicated nonlinear decision surface, and usually requires more support vectors.

As a result, the classifier for a given new test data can be constructed as:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{k} \alpha_i y_i \exp\left(-\frac{\left\| \mathbf{x} - \mathbf{x}_i \right\|^2}{2\sigma^2} \right) + b \right) \tag{6}$$

where k is the nearest support vector number and $k \leq m$ for an improved accuracy of SVM.

4 Experimental Results

In this section we perform experiments to illustrate the performance of the proposed method.

For the experiments, we use the image segmentation dataset which has 7 different classes of certain images. Each instant is a 3 by 3 region and randomly chosen from a database of 7 outdoor images. Each image is hand segmented for classification purpose. The dataset is available at [13].

We use one against all rule for each 7 classes. The first 210 data are used as training dataset and the remaining 2100 data are used for testing. Each vector has 18 features. We calculate the highest performance values by finding the appropriate Gaussian kernel parameters. Before the training the SVM, we first normalized all the data between the range [-1,1]. For each class, we first find the best kernel parameter that gives the lowest generalization error, and then by using this kernel parameter, we find the support vectors and corresponding Lagrange multipliers. For the experiments, we used and modified the code available at [14].

Table 1 shows best classification results for the test data with the corresponding Gaussian kernel parameters and support vector numbers for each class. Then by keeping the same support vectors and the corresponding α values, we applied *K* nearest SVM technique on the same dataset and the results are shown on Table 2. The best classification percentages are obtained by using the lowest *K* nearest support vectors, and are shown on Table 2 where *K* is the nearest Support vector numbers.

On Table 3 we first sorted the *α* values in descending order and then have chosen the *K* highest α values with the corresponding support vectors. The remaining *α* values are set to zero. Therefore the number of SVs is reduced in each test.

Table 1. The SVM training and best kernel parameters with SV numbers for the best classification results

Traditional SVM Test Results

Class name:	cement	brickface	Grass	foliage	sky	path	window
Best %:	96.95	99.48	99.86	96.71	100	99.71	94.57
Parameter:	0.53	0.44	0.5	1.43	1.45	0.43	0.40
SV Number	84	93	94	28	18	106	122

Table 2. K nearest SVM classification results for the image segmentation dataset

K Nearest SVM Test Results

Class name:	cement	brickface	Grass	foliage	Sky	Path	Window
Best %:	97	99.52	99.86	96.71	100	99.95	94.62
σ value:	0.53	0.44	0.5	1.43	1.45	0.43	0.4
K	37	29	7	28	3	9	25

Table 3. Using only the highest K number of α values and its results for different classes

Using only the K highest α values for SVM

Class name:	cement	brickface	Grass	foliage	Sky	Path	Window
Best %:	97	99.52	99.86	96.71	100	99.86	94.57
σ value:	0.53	0.44	0.5	1.43	1.45	0.43	0.4
SV Number	41	23	8	28	3	23	51

Table 4. Showing the maximum and minimum α values that are used and discarded in the "K highest α values for SVM" experiment

The Maximum and Minimum α values used in Table 3

Class name:	cement	brickface	Grass	foliage	Sky	Path	Window
Used max α	12.16	15.31	1.42	464.3	1.27	1.61	14.02
Used min α	0.36	0.38	0.57	1.13	1.17	0.28	0.21
Nonused max	0.34	0.21	0.38	0	0.99	0.28	0.21
Nonused min	0.002	0.001	0.001	0	0.02	0	0.004

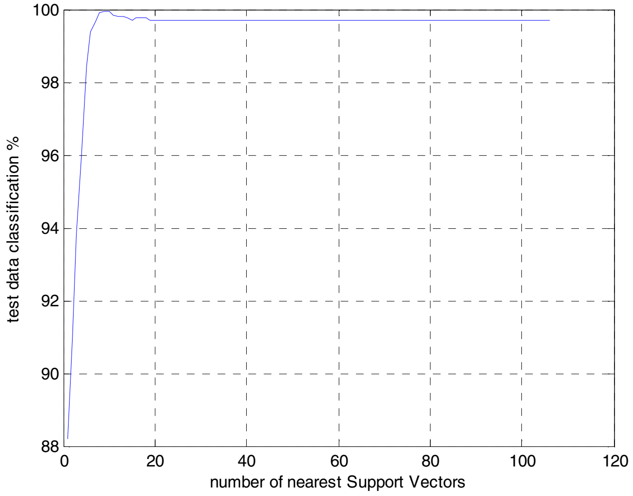


Fig. 2. For the Path class the K nearest SV number vs test classification percentage plot

The classification percentage with the same kernel parameters are shown on Table 3 for each class separately with the best *K* values. The maximum and minimum α values that are used and discarded for each class are shown on Table 4.

In Figure 2, we plotted the change on classification percentage versus the nearest support vector numbers used in Equation (7) when the kernel parameter is kept as 0.43. It can clearly be seen that the best classification result is not obtained by using all the support vectors. The peak value for the plot is obtained when the K is chosen as 9 as it is shown on the plot.

Comparing Table 1, Table 2 and Table 3, we can see that K nearest SVM gives the best results for Path and Window classes when the same kernel parameters are used. Cement and Brickface classes show the same improved performance on Table 2 and Table 3. For Foliage, Grass and Sky classes we find the same results as in the regular SVM case. However for the Grass and Sky classes the same percentage values are obtained by using lower support vector numbers on experiments.

5 Conclusion and Discussion

In this paper, as an alternative SVM tuning method, we propose using the KNN idea to decrease generalization error, when the optimum kernel parameter is used with the Gaussian kernel. Moreover, we also show that the SV number can be reduced gradually by using only the highest K number of α values for the same or an increased performance for many applications.

Based on the experimental results on Table 1, and Table 2 we can conclude that, on SVM generalization, learning with the lowest Support vector numbers is not always the best way of learning the training data when the accuracy is the main concern. Although SVM is called a “sparse learning algorithm”, it is better to keep sparseness at an optimum value (which is not the minimum value always) so that, it does not reduce the generalization ability of the SVM. Especially for highly nonlinear data structures, it is safer and better to learn with more support vectors. And then by using K nearest SVM technique, the generalization error can be decreased.

As shown on Table 1 and Table 2, the preliminary experimental results indicate us that, if the training step is completed with a small number of support vectors, then the generalization error may not be decreased with K nearest SVM as the support vectors are not close enough to form a proper smaller clusters to smoothen the decision boundary, thus we may not capture the nonlinearity of the space in a better way by using less support vectors.

Table 3 shows that, there may be some redundancy on support vector number which can be further reduced for SVM classification with the Gaussian kernel. Even the α values may have similar values (not closer to zero), by choosing only the K highest α values, and setting all the remaining ones to zero, we can obtain the same generalization performance. Finding this K value is an important step and it can be found heuristically. This result can be quite useful where the SV number is more important such as in feature selection and feature extraction applications. Less support vector also means less computation time for a given test data.

The information we obtain from this study when combined with previous similar works, shows us that there are interesting properties with the Gaussian kernel, and there is a relation between the decision boundary and the kernel parameter as well as the K value. We will use these preliminary results in our next study to obtain a novel method that finds its own parameters automatically during the training step.

References

[1] El-Naqa, I., Yang, Y., Wernick, M.N., Galatsanos, N.P., Nishikawa, R.M.: A support vector machine approach for detection of microcalcifications. IEEE Trans. on Medical Imaging 21(12), 1552–1563 (2002)

[2] Artan, Y., Huang, X.: Combining multiple 2ν-SVM classifiers for tissue segmentation. In: Proc. of ISBI 2008, pp. 488–491 (2008)

[3] Lucey, S.: Enforcing non-positive weights for stable support vector tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8 (2008)

[4] Ozer, S., Haider, M.A., Langer, D.L., van der Kwast, T.H., Evans, A.J., Wernick, M.N., Trachtenberg, J., Yetik, I.S.: Prostate Cancer Localization with Multispectral MRI Based on Relevance Vector Machines. In: ISBI 2009, pp. 73–76 (2009)

[5] Vapnik, V.N.: Statistical Learning Theory. John Wiley & Sons, Chichester (1998) ISBN: 0-471-03003-1

[6] Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)

[7] Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13(1), 21–27 (1967)

[8] Ming, T., Yi, Z., Songcan, C.: Improving support vector machine classifier by combining it with k nearest neighbor principle based on the best distance measurement. IEEE Intelligent Transportation Systems 1, 373–378 (2003)

[9] De Coste, D., Mazzoni, D.: Fast query-optimized kernel machine classification via incremental approximate nearest support vectors. In: 20th International Conference on Machine, Learning - ICML, Washington, DC (2003)

[10] Zhang, H., Berg, A.C., Maire, M., Malik, J.: Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In: IEEE Conference on Computer Vision and Pattern Recognition CVPR (2006)

[11] Zhang, L., Zhou, W., Jiao, L.: Wavelet Support Vector Machine. IEEE Trans. On Systems, Man, and Cybernetics-Part B: Cybernetics 34(1), 34–39 (2004)

[12] Ozer, S., Chen, C.H.: Generalized Chebyshev Kernels for Support Vector Classification. In: 19th International Conference on Pattern Recognition, ICPR (2008)

[13] Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, `http://www.ics.uci.edu/~mlearn/MLRepository.html`

[14] Canu, S., Grandvalet, Y., Guigue, V., Rakotomamonjy, A.: SVM and Kernel Methods Matlab Toolbox, Perception Systèmes et Information, INSA de Rouen (2005) `http://asi.insa-rouen.fr/enseignants/~arakotom/toolbox/index.html`

Automatic Speech Segmentation Based on Acoustical Clustering*

Jon A. Gómez, Emilio Sanchis, and María J. Castro-Bleda

Departamento de Sistemas Informáticos y Computación,
Universidad Politécnica de Valencia, Spain
{jon,esanchis,mcastro}@dsic.upv.es
http://elirf.dsic.upv.es

Abstract. In this paper, we present an automatic speech segmentation system based on acoustical clustering plus dynamic time warping. Our system operates at three stages, the first one obtains a coarse segmentation as a starting point to the second one. The second stage fixes phoneme boundaries in an iterative process of progressive refinement. The third stage makes a finer adjustment by considering some acoustic parameters estimated at a higher subsampling rate around the boundary to be adjusted. No manually segmented utterances are used in any stage.

The results presented here demonstrate a good learning capability of the system, which only uses the phonetic transcription of each utterance. Our approach obtains similar results than the ones reported by previous related works on TIMIT database.

Keywords: automatic speech segmentation, phoneme boundaries detection, phoneme alignment.

1 Introduction

It is well known the usefulness of phonetically segmented speech corpora for several purposes. Lately, there is a special attention in the selection of phonetic units for Text-To-Speech (TTS) systems. However, the availability of segmented speech databases for training acoustic models continues being of interest in the construction of Automatic Speech Recognition (ASR) systems.

The manual segmentation of speech corpora is a hard work which implies many hours of human phonetic experts, and it does not avoid some deviations due to different human expert criteria. Some researchers have given the same speech database to different human experts to segment it. Then, they evaluated the difference between the manual segmentations obtained. In [1], 97% of the boundaries within a tolerance interval of 20 *ms* were found, and 93% in [2].

Our method does not need any subset of manually segmented sentences for bootstrapping. The input to our system are both the speech signal and the known phonetic sequence of each utterance. It fixes correctly 88% of boundaries

* Work supported by the Spanish MICINN under contract TIN2008-06856-C05-02.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 540–548, 2010.

within a 20 ms tolerance interval compared to a manual segmentation. Therefore, their output is suitable for training acoustic models in ASR systems, such as those based on Hidden Markov Models (HMM), Neural Networks or hybrid systems. This method was used on selecting phonetic units for TTS as part of a comparative study. It obtained better results than HMM in an objective test, but worse in a perceptual one [3].

A work for speech/music discrimination of radio recordings that uses similar techniques was presented in [4]. Their goal is to segment audio streams by classifying each segment as either speech or music, while our goal is the adjustment of phonetic boundaries. Both our and their system operate on three stages and use dynamic programming for optimal segmentation. Nevertheless, their dynamic programming algorithm discriminates speech against music using posterior class probabilities estimated by means of Bayesian Networks. Our dynamic time warping algorithm uses posterior phonetic probabilities computed as described below in Section 2.2.

Next section describes our automatic segmentation system. Section 3 explains the measures used for evaluating segmentation accuracy. Section 4 presents experimental results and discuses them, and Section 5 concludes.

2 The Speech Segmentation System

Our automatic speech segmentation system attempts to solve the problem in three stages. The first stage estimates a coarse segmentation which is used as a starting point to the second one. The second stage does a progressive refinement of phoneme boundaries by means of a dynamic time warping (DTW) algorithm which uses phonetic probabilities estimated at each frame. The third stage adjusts the boundaries in a more precise manner.

A previous version of our system was presented [5] and it worked in two stages. The coarse segmentation module fixed phoneme boundaries following a knowledge-based approach by using a set of language-dependent acoustic-phonetic rules. New coarse segmentation is language-independent and positions phoneme boundaries based on statistical analysis of acoustic parameters. The whole system has been adapted to be language-independent. Results for both English and Spanish languages are presented in Section 4.

2.1 Coarse Segmentation

Initial phoneme boundaries are positioned using classification techniques at different levels and doing four consecutive steps. First and second steps are applied to each sentence individually. In the first step, time marks are placed where acoustic changes are considered relevant. Every two consecutive time marks define an *acoustic segment*. In the second step, acoustic segments are associated with phonetic units following simple acoustic-phonetic rules. In the third step, the previous association is used to estimate a Gaussian Mixture Model (GMM) with several Gaussians per phonetic unit. A new association of acoustic segments with phonetic units is performed based on phonetic probabilities from the GMM, then, an iterative process refines the GMM until no changes in association are found. In the fourth step, time marks from step 1 are not used as reference, so phoneme boundaries are fixed by using only the phonetic probabilities. The first iteration of this step uses the phonetic probabilities provided by the GMM computed in the last iteration of previous step. The iteration stops when no changes on boundary positions are found. The algorithm follows:

1) Location of time marks in relevant acoustic changes. Time marks are placed between two consecutive frames which are classified in different classes. This strategy gives us a sequence of time marks where a phoneme boundary is located with high probability. Here, time marks are established in three levels. The first level does clustering with two Gaussian distributions using two parameters: energy (E) and the first cepstral coefficient ($CC1$). Then, frames are classified into one of the two classes, and a time mark is fixed when frame class changes. The next level performs clustering with three Gaussian distributions, using again E and $CC1$, and new time marks are obtained, most of them on the same location than the previous marks. With these clustering processes, boundaries between fricatives and no fricatives are found by the first level, the second level confirms them and find new boundaries between silence and no silence. The third clustering process works inside the acoustic segments delimited by existing time marks. They are clustered in 2, 3 and 4 classes, and the number of classes with the lower entropy is selected. New time marks are obtained. This step is repeated until no acoustic segments larger than 60 ms remains.

2) Association of acoustic segments with phonetic units. A DTW algorithm which takes into account the following acoustic-phonetic rules is used to associate acoustic segments with phonetic units:

Rule (a). The association of an acoustic segment with a silence is penalized proportionally to the value of E. The association of an acoustic segment with a fricative or stop plosive phoneme is penalized with low values of E and high values of $CC1$. Finally, the association of an acoustic segment with other phonemes is penalized with low values of E and with low values of $CC1$.

Rule (b). The length of one or more consecutive acoustic segments associated with a phoneme is used to penalize the association if it is too short, except for stop plosive consonants and silences.

Rule (c). The length of an acoustic segment associated with a stop plosive consonant is used to penalize the association if it is larger than 30 ms.

3) Association of segments with phonetic units using phonetic probabilities. The output of the previous association of acoustic segments with phonetic units is a primary segmentation used to estimate a GMM with several Gaussians per unit, typically 16. A DTW algorithm is also followed, but rule (a) is substituted by the use of phonetic probabilities from the GMM. Rules (b) and (c) are also used here with the same purpose. A new segmentation is obtained and a new GMM is estimated. This step is repeated until no changes in associations are found.

4) Forced alignment of phonetic units using phonetic probabilities. We follow the same strategy as before, but association of phonetic units with acoustic frames is not restricted by time marks. Rules (b) and (c) continue being applied. Actually, this step does the same alignment described in next subsection, but the phonetic probabilities used here are not as precise as the ones used in the progressive refinement.

2.2 Progressive Refinement

This is the core of our segmentation technique: Acoustical Clustering-Dynamic Time Warping (AC-DTW). It is based on unsupervised learning of acoustic classes and its association to phonemes by means of conditional probabilities. Each acoustic class represents a particular kind of acoustical manifestation and is modelled by a Gaussian distribution.

Phonetic boundaries are established by a DTW algorithm that uses the *a posteriori* probability of each phonetic unit given an acoustic frame. These *a posteriori* probabilities of phonemes are calculated by combining probabilities of acoustic classes, which are obtained from a clustering procedure on the acoustic feature space, and the conditional probabilities of each acoustic class with respect to each phonetic unit [5].

In the clustering procedure, it is assumed that acoustic classes can be modelled by means of Gaussian distributions. Parameters of each Gaussian distribution are estimated by using the unsupervised version of the Maximum Likelihood Estimation (MLE) procedure [6]. Thus, it is possible to estimate $\Pr(a|x_t)$, that is, the probability of each acoustic class a from the set A of acoustic classes, given an acoustic frame x at time t, x_t, from the GMM. Nevertheless, as we need the probability of each phonetic unit u from the set U of phonetic units, given an acoustic vector x_t, $\Pr(u|x_t)$, a set of conditional probabilities are estimated in order to calculate the phonetic probabilities from the acoustic ones.

The use of conditional probabilities allows us to compute the phonetic-conditional probability density $p(x_t|u)$ as follows [5]:

$$p(x_t|u) = \sum_{a \in A} p(x_t|a) \cdot \Pr(a|u) \tag{1}$$

for each $u \in U$, where $p(x_t|a)$ is the acoustic class-conditional probability density, computed as a Gaussian probability density function, and $\Pr(a|u)$ is the conditional probability that acoustic class a has been manifested when phonetic unit u has been uttered. Then, applying the Bayes formulation, we obtain the phonetic probabilities as:

$$\Pr(u|x_t) = \frac{\sum_{a \in A} p(x_t|a) \cdot \Pr(a|u)}{\sum_{v \in U} \left(\sum_{a \in A} p(x_t|a) \cdot \Pr(a|v) \right)} \tag{2}$$

for each $u \in U$. The DTW algorithm uses these *a posteriori* phonetic probabilities to align the frame sequence with the phonetic transcription.

The set of conditional probabilities $\Pr(a|u)$ for all $a \in A$ is initially computed from the coarse segmentation described in previous subsection. An iterative process updates the conditional probabilities until no improvements on segmentation are found.

2.3 Boundary Adjustment

A boundary adjustment is made from the segmentation previously obtained. This adjustment takes into account the values of several acoustic parameters to move phonetic boundaries. The parameters used at this stage are dE, the absolute value of first time derivative of Energy, dZ, the absolute value of first time derivative of zero crossing rate (Z), and $dEdZ = dE * dZ$. Energy and Z are computed every 2 ms using a 10 ms window.

Each phoneme boundary is adjusted using the gravity center formula with respect to a function inside a window centered in it. Both the function used and the window length depend on which phonetic units are related with the boundary:

- *A stop plosive consonant followed by any other phoneme*, gravity center of dE calculated within a 20 ms window.
- *A fricative consonant followed or preceded by any other phoneme*, gravity center of $dEdZ$ calculated within a 60 ms window.
- *Silence followed or preceded by any other phoneme*, gravity center of dE calculated within a 40 ms window.
- *Vowel followed by other vowel*, considered as a special case.
- *Any other pair of consecutive phonetic units*, gravity center of dE plus $dEdZ$ calculated within a 40 ms window.

Boundaries between consecutive vowels are adjusted by dividing the sequence of frames from concatenating the two vowel segments into three subsegments with same length. Then it begins an iterative process which reduces the central segment as follows: if the first frame of the central segment is closer to the left segment than the central one, then that frame belongs to the left segment; by the other hand, if the last frame of the central segment is closer to the right segment than the central one, then that frame belongs to the right segment. When the central segment disappears or becomes unchanged, loop ends. In this last case, the adjusted boundary is fixed as the mean of the central segment boundaries.

3 Segmentation Evaluation

The evaluation criteria most widely used in the literature is to measure agreement of the obtained segmentation with respect to a manual segmentation. Usually the percentage of boundaries whose error is within a tolerance is calculated for a range of tolerances [1,2,7].

As discussed in the introduction, some researchers have wondered whether or not a manual segmentation is a valid reference [1,2]. To evaluate it, they gave the same speech database to different human experts to segment it, and they evaluated the difference between them. In the study presented in [1], 97% of the boundaries within a tolerance of 20 *ms* were found and in [2] 93%. We interpret this agreement as the maximum accuracy for a segmentation system, since a system that reaches 100% compared with a manual segmentation will at least differ around 95% with another manual segmentation for the same speech database.

4 Experimental Results

4.1 Corpora

In order to carry out experiments for both Spanish and English, we used two speech databases: *Albayzin* [8] and TIMIT [9], respectively.

The phonetic subcorpus from *Albayzin* database was used for the Spanish experiments: 6,800 utterances (around six hours of speech) obtained by making groups from a set of 700 distinct sentences uttered by 40 different speakers. 1,200 sentences manually segmented and labelled were used for testing, the remaining 5,600 sentences were used for training. No intersection speakers between training and testing subcorpora exist.

The TIMIT database was used for the English experiments: 6,300 utterances (approximately five hours of speech) by making groups of 10 sentences spoken by 630 speakers from 8 different dialect divisions of the United States. Two sentences were uttered by all speakers, the other eight sentences were selected from two phonetically rich sets. We used the suggested training/test subdivision [9].

The same acoustic parameters were used on both databases. Each acoustic frame was formed by a 39-dimensional vector composed by the normalized energy, the first 12 Mel frequency cepstral coefficients, and their first and second time derivatives. An acoustic frame is obtained using a 20 *ms* Hamming window at two different subsampling rates: 100 Hz (one frame every 10 *ms*) and 200 Hz (one frame every 5 *ms*) in order to study the influence of subsampling rate on segmentation accuracy.

4.2 Coarse Segmentation

As explained in subsection 2.1, the coarse segmentation is done in several steps. First, time marks are fixed where relevant acoustic changes are detected by means of statistical analysis of some acoustic parameters. Following, time marks are used to define acoustic segments which must be associated with phonemes in the phonetic transcription of each sentence. This association is refined until no changes are found. Then, the alignment of the frame sequence with the phonetic transcription begins. This alignment also repeats until no boundaries changes are detected. Table 1 shows the percentage of correctly fixed phonetic boundaries for the coarse segmentation. A set of tolerance intervals are considered.

Table 1. Coarse segmentation. Percentage of correct phonetic boundaries within a set of tolerance intervals for the coarse segmentation.

Database	<5 *ms*	<10 *ms*	<15 *ms*	<20 *ms*	<30 *ms*
Albayzin	35.7 %	59.0 %	70.0 %	76.2 %	83.3 %
TIMIT	37.5 %	61.4 %	71.8 %	77.0 %	83.8 %

4.3 Boundary Adjustment

The progressive refinement estimates a set of conditionals probabilities in an iterative process. The conditional probabilities are combined with the acoustical ones to obtain the *a posteriori* phonetic probabilities (see subsection 2.2). These phonetic probabilities are used by a DTW algorithm to align the acoustic frame sequence with the phonetic transcription. A phoneme segmentation is obtained as output, then a final boundary adjustment is done in order to improve the location of phoneme boundaries with respect to speech signal (see subsection 2.3).

Table 2 shows the percentage of correctly fixed phonetic boundaries obtained before and after applying the boundary adjustment when 100 Hz subsampling rate was used. Table 3 shows the same results when 200 Hz subsampling rate was used. It can be observed a significant improvement when the boundary adjustment is applied, specially for shorter tolerance intervals. In contrast, improvement when using higher subsampling rates is not as significant, even, there is no appreciable differences when using *Albayzin* database.

Another aspect to point out is the difference between both databases. This difference could reveal that our system is biased in favour of Spanish language. However, our results working on TIMIT database are similar to the ones reported in [7]. Their segmentation accuracy within a tolerance interval of 20ms is 83.6%, our segmentation accuracy is 84.7%.

We made additional experiments using a set of manually segmented and labelled sentences as a starting point to the refinement process. Thus, we calculate the *a posteriori* probabilities using the best conditional probabilities we can obtain. Table 4 shows these segmentation results, which represent an upper bound of our segmentation technique, confirming that our system can learn without manually segmented and labelled sentences.

Table 2. Percentage of correct boundaries within a set of tolerances before and after the boundary adjustment. Subsampling rate 100 Hz.

Database	<5 *ms*	<10 *ms*	<15 *ms*	<20 *ms*	<30 *ms*
Albayzin (before)	41.0 %	67.2 %	80.4 %	87.3 %	94.0 %
Albayzin (after)	47.0 %	72.0 %	83.0 %	**88.8 %**	94.3 %
TIMIT (before)	35.3 %	60.5 %	74.4 %	81.6 %	90.1 %
TIMIT (after)	40.6 %	65.9 %	77.1 %	82.9 %	89.9 %

Table 3. Percentage of correct boundaries within a set of tolerances before and after the boundary adjustment. Subsampling rate 200 Hz.

Database	<5 ms	<10 ms	<15 ms	<20 ms	<30 ms
Albayzin (before)	40.5 %	65.5 %	79.7 %	87.6 %	94.2 %
Albayzin (after)	46.2 %	71.2 %	82.8 %	88.5 %	94.2 %
TIMIT (before)	37.9 %	62.7 %	76.0 %	83.3 %	91.6 %
TIMIT (after)	42.4 %	67.9 %	79.0 %	**84.7 %**	91.2 %

Table 4. Percentage of correct boundaries within a set of tolerances before and after the boundary adjustment when manually segmented and labelled sentences were used to estimate the conditional probabilities. Subsampling rate 200 Hz.

Database	<5 ms	<10 ms	<15 ms	<20 ms	<30 ms
Albayzin (before)	44.9 %	70.5 %	83.5 %	89.8 %	95.4 %
Albayzin (after)	50.2 %	74.8 %	85.6 %	90.7 %	95.5 %
TIMIT (before)	41.1 %	65.9 %	79.3 %	85.9 %	92.9 %
TIMIT (after)	46.1 %	71.2 %	81.7 %	86.8 %	92.6 %

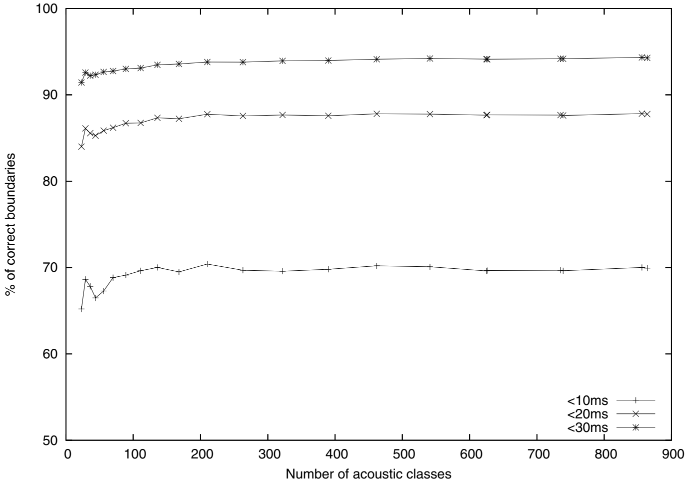


Fig. 1. Percentage of correct phonetic boundaries versus the number of acoustic classes using *Albayzin* database and 100 Hz subsampling rate. Tolerance intervals of 10, 20 and 30 ms are presented.

In order to study the influence of the number of acoustic classes an exploratory experiment was performed. The progressive refinement, explained in Section 2.2, was repeated for a set of GMM with different number of mixture components. Each GMM with a particular number of Gaussian distributions is the product of each intermediate step in the hiearchical clustering procedure applied to estimate the "natural" acoustic classes. Figure 1 shows the segmentation accuracy obtained for different values of the number of acoustic classes. There is no significant improvement from 200 acoustic classes.

5 Conclusions

In this work, we have presented a fully automatic system to segment speech databases without the need for a manually segmented subset. This task is important in order to obtain segmented databases for training phoneme-based speech recognizers or selecting phonetic units in TTS systems.

The improvement in coarse segmentation stage has impact in the final segmentation. The results obtained using the conditional probabilities estimated from a set of manually segmented and labelled sentences represent an upper bound of our technique. The small difference with respect to the automatic system validates our technique, which no uses manually segmented and labelled sentences at all. The segmentation accuracy obtained here for TIMIT database is similar to the results presented in other works with the same database using more complex methods for the final adjustment.

References

1. Toledano, D.T., Hernández Gómez, L., Villarrubia Grande, L.: Automatic Phonetic Segmentation. IEEE Transactions on Speech and Audio Processing 11(6), 617–625 (2003)
2. Kipp, A., Wesenick, M.B., Schiel, F.: Pronunciation modelling applied to automatic segmentation of spontaneous speech. In: Proceedings of Eurospeech, Rhodes, Greece, pp. 2013–2026 (1997)
3. Adell, J., Bonafonte, A., Gómez, J.A., Castro, M.J.: Comparative study of automatic phone segmentation methods for TTS. In: IEEE ICASSP, Philadelphia, USA, vol. 1, pp. 309–312 (2005)
4. Pikrakis, A., Giannakipoulos, T., Theodoridis, S.: A Speech/Music Discriminator of Radio Recordings Based on Dynamic Programming and Bayesian Networs. IEEE Trans. on Multimedia 10, 846–857 (2008)
5. Gómez, J.A., Castro, M.J.: Automatic Segmentation of Speech at the Phonetic Level. In: Caelli, T.M., Amin, A., Duin, R.P.W., Kamel, M.S., de Ridder, D. (eds.) SPR 2002 and SSPR 2002. LNCS, vol. 2396, pp. 672–680. Springer, Heidelberg (2002)
6. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley & Sons, Chichester (2001)
7. Mporas, I., Ganchev, T., Fakotakis, N.: A Hybrid Architecture for Automatic Segmentation of Speech Waveforms. In: IEEE ICASSP 2008, Las Vegas, USA, pp. 4457–4460 (2008)
8. Moreno, A., Poch, D., Bonafonte, A., Lleida, E., Llisterri, J., Mariño, J.B., Nadeu, C.: Albayzin Speech Database: Design of the Phonetic Corpus. In: Eurospeech 1993, Berlin, Germany, September 1993, vol. 1, pp. 653–656 (1993)
9. TIMIT Acoustic-Phonetic Continuous Speech Corpus, National Institute of Standards and Technology Speech Disc 1-1.1, NTIS Order No. PB91-5050651996 (October 1990)

An Efficient Iris and Eye Corners Extraction Method

Nesli Erdogmus and Jean-Luc Dugelay

Eurecom, Multimedia Communications Department
2229 Routes des Crêtes, 06904 Sophia Antipolis, France
{nesli.erdogmus,jean-luc.dugelay}@eurecom.fr
http://image.eurecom.fr

Abstract. Eye features are one of the most important clues for many computer vision applications. In this paper, an efficient method to automatically extract eye features is presented. The extraction is highly based on the usage of the common knowledge about face and eye structure. With the assumption of frontal face images, firstly coarse eye regions are extracted by removing skin pixels in the upper part of the face. Then, iris circle position and radius are detected by using Hough transform in a coarse-to-fine fashion. In the final step, edges created by upper and lower eyelids are detected and polynomials are fitted to those edges so that their intersection points are labeled as eye corners. The algorithm is experimented on the Bosphorus database and the obtained results demonstrate that it can locate eye features very accurately. The strength of the proposed method stems from its reproducibility due to the utilization of simple and efficient image processing methods while achieving remarkable results without any need of training.

Keywords: Eye features extraction, iris, eye corners and eyelids.

1 Introduction

Facial features have a crucial role in many computer vision applications such as face normalization, facial expression recognition or model-based human face coding. For this reason, automation of their extraction has a wide range of usage. Among those features, eyes have the highest importance with their higher prominence and stability compared to other facial features. In [1], it is proven that the eyes can improve the recognition performance as compared to the nose and mouth. The eyes features include iris center (or pupil center) and radius, eyelid contours and eye corners which are located at the intersection of the upper and lower eyelids.

In most cases, firstly the eye region is extracted. Many different methods have been proposed for this task, such as extracting contrasted components by morphological operations [4], using eye filters to detect eye candidates [5] or by projecting the facial edge map vertically and horizontally, where the maximum points of the projection curves are associated to the eye positions [6].

For iris, there are numerous approaches where the center and the radius are searched separately [2] or together [3]. In [2], the physiological property of the pupil is used to detect the center. Due to the pupil's response to the light, it is the brightest

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 549–558, 2010.

region in H channel of the HSV color space. After the detection of the center, iris radius is estimated so that the mean gray level of the pixels in the circle is the lowest. On the other hand in [3] circular Hough transform is used to detect the iris border where both center and radius are estimated simultaneously. In some approaches, the iris radius is supposed to be known [8] or limited to a set of expected values [7].

In order to locate eye corners, one general approach is projection functions [9]. The weakness of Integral Projection Function (IPF) to reflect well the variation in the image Variance Projection Function (VPF) is proposed [10]. Later this approach is diversified as General Projection Function (GPF) and Hybrid Projection Function (HBF) which combine IPF and VPF and Weighted Variance Projection function (WVPF) [11] in which pixels are assigned weights according to their Harris corner response. Utilization of deformable templates [7] is another common approach to detect eye corner positions which often requires a good initialization in order to avoid incorrect results. Additionally, in [2] eye-corner filter using Gabor feature space is proposed for eye-corner detection and in [12], two semantic features for eye corners are introduced which are further fused by logistic regression classifier to determine their accurate locations.

Lastly, for eyelids the proposed methods can be classified under two groups: using deformable contour models [13], curve fitting [7, 14]. As mentioned before, for deformable contour models initialization is crucial. Additionally, the energy term should be formulated carefully to reach an accurate result. In curve fitting approach, usually the eyelid contours are extracted after the detection of the eye corners. In [7, 14], parabolic sections with parameters controlling its curvature, position and rotation is fitted to a set of points including the corner points and the detected edges in the eye region whereas in [15] edges are replaced by four control points where iris border and the eyelids intersect.

In this paper, the facial region in the image is assumed to be known and the eye region is taken to be the non-skin region in the upper half of the facial image with the assumption of frontal face with the nose being vertical. Firstly, a coarse localization of the irises is performed in the estimated eye region by circle detection using Hough transform. The detected circles are subjected to elimination with the help of a priori knowledge about relative size and position of irises. Afterwards, the color images of the eye regions (window around the coarsely detected iris centers) are further processed to refine the iris radius and location. Finally, the cropped eye images are segmented into three color regions and contrary to previous works, the eyelid contours are estimated first to obtain the eye corners on their intersection points.

The rest of this paper is organized as follows: In section 2, the method for detection of the eye regions by extracting the non-skin part of the face is presented in detail. The coarse estimation of the iris centers and radiuses and afterwards the refining of these results are explained in section 3. Section 4 is on eye corners detection method. Finally, section 5 is where the conducted tests and their results are represented, followed by the conclusions in section 6.

2 Eye Region Extraction

The eye region in the facial image is extracted under the assumptions that the face is frontal with the line connecting the eye centers close to horizontal. Hence, the upper half of the face is taken to be analyzed. The non-skin region is found by removing the pixels with the most frequent (C_b, C_r) values present in the image, using YC_bC_r space. For this purpose, firstly the histogram is calculated for distribution analysis. Even though the face image is cropped into its upper half where the eyes are located, still the skin pixels constitute the majority. Taking the histogram into account, a threshold is set according to the maximum count and the image size. Afterwards, the pixels with higher value than this threshold is eliminated as skin pixels. Lastly, the small islands in the obtained binary mask are removed. In Fig. 1, an example set of images is given to demonstrate the process (in which the forehead is not shown to have better view of the eye region).

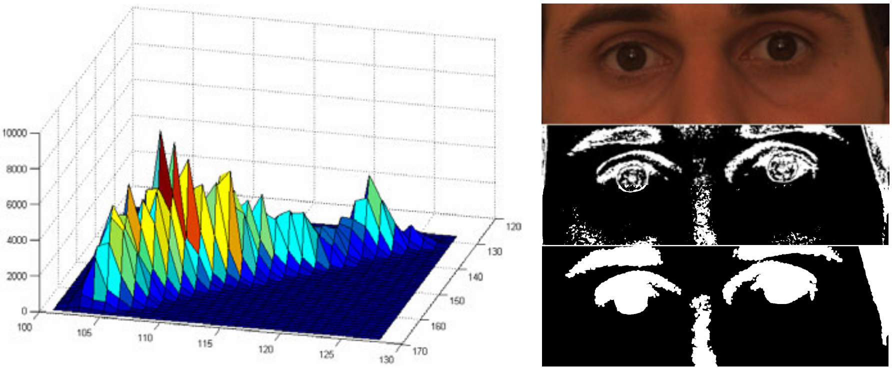


Fig. 1. A (C_b, C_r) histogram and the resulting mask after thresholding. As you can see, the non-skin region which includes eyes is clearly separated from the rest.

Since the algorithm proposed in this paper is stepwise, the iris detection results affect the rest. Hence, this part is added to the system as a supportive module to improve the iris extraction by removing other possible circular edges as much as possible. The improvement due to this addition can be seen in “Tests and Results” section.

3 Iris Extraction

After obtaining the eye regions, firstly edge maps are constructed by Canny edge detector. The drawback of this edge detection method is that it requires a good adjustment of the threshold. In order to overcome this issue, we propose to use the edge detector iteratively, by tuning the threshold parameter until a descriptive edge map is obtained. Afterwards, Hough transform is applied to the edge map to detect circles. For each detected circle, an overlapping score is calculated by the ratio of the detected portion of the circle to the whole circle parameter. Here, we define the “descriptiveness” of an edge map by the number of the edge pixels in the image and the number of circles that can be detected using these edges. For circle detection, minimal and maximal radius values are defined to speed up the process.

Subsequently, the detected circles (iris candidates) are grouped into two classes according to their position: right side and left side. Then, for all possible pairs of right and left circles, those criteria are applied:

- Vertical distance of the centers
- Horizontal distance of the centers
- Difference between radiuses

Among the compatible pairs, the one with maximum total overlapping score is chosen to be the two irises. In Fig. 2, the procedure to roughly obtain the iris positions and dimensions is depicted with examples.

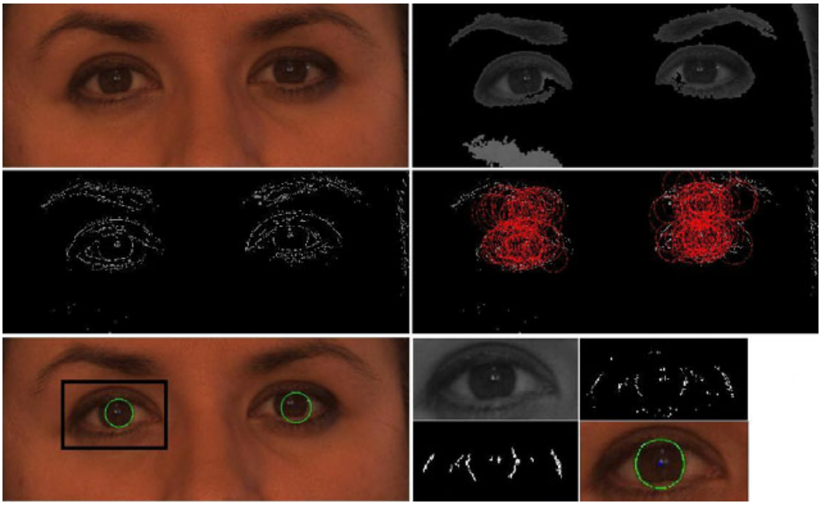


Fig. 2. From left to right, top to bottom: a. Input image b. Masked image after skin region removal c. Detected edges d. Detected circles e. New eye region window f. Refining of the iris position and radius after detecting best circle to detect vertical edges

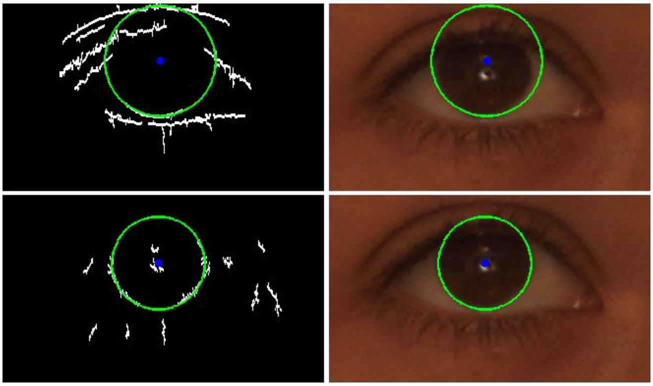


Fig. 3. The positive effect of using vertical edges only can be observed when the two detected iris circles are compared

Once the approximate positions of the irises are obtained, rectangular windows centered at the detected iris centers are extracted and analyzed separately. Firstly, an averaging filter is applied with a rectangular kernel, where the noise and horizontal edges are suppressed and vertical edges are preserved to some point. Then, the vertical edges are detected by using the Sobel operator. As also explained in [7], the upper and the lower parts of the iris border are mostly occluded by eyelids. This leads to incorrect hints for circle detection (Fig. 3). Therefore, only the vertical edges are detected. Then, the obtained edge map is cleaned with the help of morphological operations where only the connected components which are larger than a threshold, are preserved. Since edges detected by the Sobel operator are often broken, vertical dilation is applied before the removal of small islands.

Using this edge image, similar to the previous approach, circles are again detected by using the Hough transform method. The circle with the maximum score provides us the center and the radius of the iris.

4 Eye Corners Extraction

For this part, the eye images are further cropped since now the accurate iris centers and radiuses are known. In this approach, firstly the eyelids contours are aimed to be detected which can be used to determine the eye corners. For this purpose, the edges created by the eyelids are searched for. The edge detection is done in two ways:

- On the color segmented image
- On the grayscale image

The details are given in the following sections:

4.1 Eyelid Detection on the Color Segmented Eye Images

Firstly, the color eye image is segmented into 3 regions: dark regions like iris and eye lashes, skin regions and sclera (white part of the eye ball). In this segmentation, at the beginning the input eye image is coarsely represented using 10 bins. For this coarse representation, spatial information from a Histogram based windowing process is used [16]. Next, k-means is used to cluster the coarse image data. The cluster centroid locations are initialized with the mean value of the 70 manually collected colors for each region.

After clustering, the resulting segmented image is convolved with horizontal and vertical Sobel operators to detect the corresponding edges and for each edge its angle is calculated. In view of the fact that eyelids are mostly closer to horizontal, only the edges with less than 45 degrees are taken into account. Additionally, similar to the processing in section 3 for iris extraction, horizontal dilation is applied to connect broken edges and then small sections are removed.

4.2 Eyelid Detection on the Grayscale Eye Images

In a similar manner to the segmented ones, in this part grayscale eye images are processed to detect the eyelids. Horizontal edges are detected again using Sobel operators. Since the edges are not as well-defined, small parts of the iris border are also detected as horizontal. In order to solve this, the edges detected in the close neighborhood of the previously detected iris contour are removed. Lastly, morphological thinning operation is applied on the resulting edge map.

4.3 Final Edge Map and Eye Corners Detection

After the two detected edge maps are superimposed, the following method is applied to remove outliers:

Observing that the eye corners are mostly located lower than the iris center, two lines are created, which are imagined to be approximately connecting the iris center and the corners. The slope of both lines is empirically chosen to be 1/3. Afterwards, only the closest edges that are below and above these lines are labeled as upper and lower eyelids. This method is illustrated in Fig. 4.

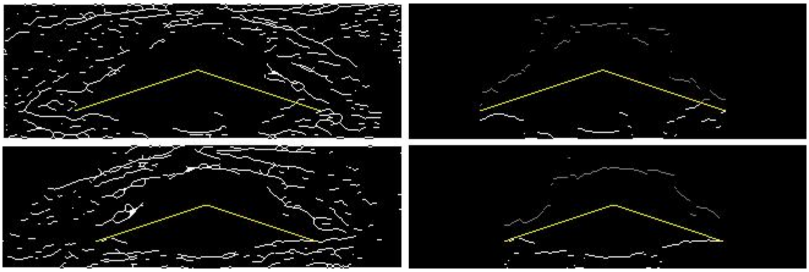


Fig. 4. Two example edge maps before and after the method is applied

In the final step, 2nd and 3rd degree polynomials are fitted for lower and upper eyelids edges respectively, in a least squares sense. The fitting is repeated once more with only the edges close to the first estimation, to further remove the outliers that still exist. The inner (near the nose bridge) and outer eye corners are determined as the intersection points of the two fitted polynomial. In Fig. 5, a set of sample images is given to illustrate each step of this section.

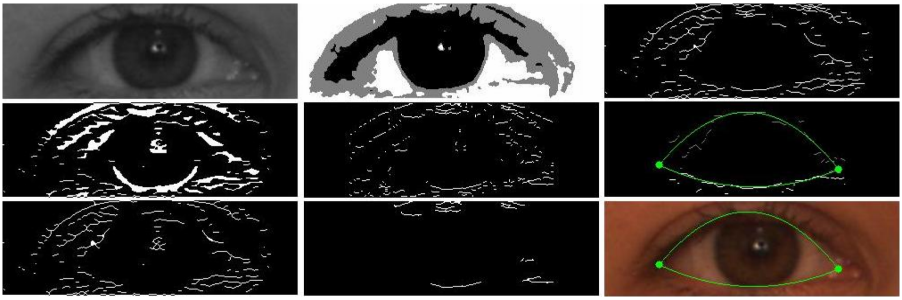


Fig. 5. Each column from left to right: a. Edge detection using grayscale image and removal of the edges close to the iris contour b. Detection of edges with less than 45° using segmented image and removal of small sections c. Fusing the two edge maps and curve fitting after eliminating edges that are not related to eyelids

5 Tests and Results

The method proposed in this paper is tested on the Bosphorus Database [17] which is actually a database of 3D faces but also supplies 2D high resolution color images which are cropped for face regions. The presented eye features extraction approach is applied to the neutral and frontal images of 105 subjects in the database. The image sizes are not fixed and they change between (936-1404) × (1218-1740) pixels. The computational time in Pentium(R) Dual-Core CPU 2.49GHz, using MATLAB is less than 13 seconds for a single face.

For the evaluation of the iris center position and its radius, it is not very easy to determine the ground truth because first of all, even for a simple perfect circle it is not easy to find the accurate center manually. Measuring the "width" of the iris to find the radius is also defective since the diameter should be measured through the exact center. Hence, manual marking or measurement of these features can yield to incorrect evaluations. For this reasons, the results are examined visually for iris extraction which are given in Table 1.

Table 1. Success rates for iris localization

Method	Threshold	Success rates
[6]	-	94.82%
[7]	-	94%
Our method –without the eye region extraction module	5 pixels	95.23%
Our method –with the eye region extraction module	5 pixels	100%

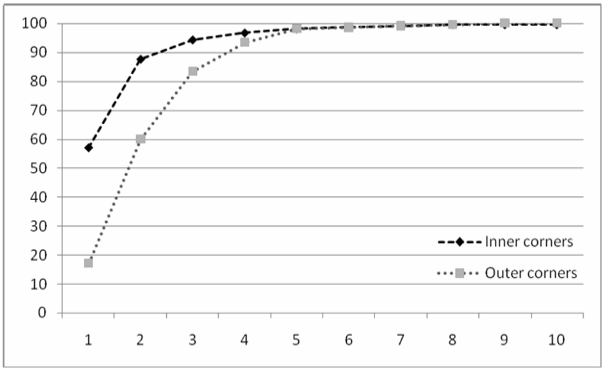


Fig. 6. Eye corner detection rates for different thresholds – Approach #1

On the other hand, since the eye corners are not very well defined, visual inspection is not an option. Hence, the corners are marked manually to constitute the ground truth. The error for the eye corners are calculated in two different manners. Firstly, the error is taken to be the Euclidean distance between the detected and the manually labeled corner. Secondly, as suggested in [11], the error is defined as this distance divided by the standard Euclidean distance between two inner eye corners. But since this "standard" distance is not revealed in [11], it is taken to be:

- the distance between two inner eye corners in that image
- the mean distance between two inner eye corners of all images

For the first error definition, the results are given in Fig. 6, where the threshold considered for accurate detection is defined from 1 to 10 pixels. This error is calculated after the images are scaled according to the distance between two iris centers, to be comparable with [12], in which this distance is fixed to 60 pixels. In Table 2, the results are listed.

For the second one, both approaches are evaluated. According to these results, it is revealed that using a constant to scale the error can be misleading. As can be seen in Fig. 7, the success rates are seemed to be higher when the mean distance between two inner eye corners is used instead of the real distance for each image itself. The second approach is much more informative and hence presented here to be used in further comparisons. In the graph, the error threshold is scaled from 1% to 15%.

Table 2. Success rates – Approach #1

Method	Threshold	Success rates for inner corners	Success rates for outer corners
[12]	4 pixels	96.89%	94.89%
Our method	4 pixels	96.67%	93.33%

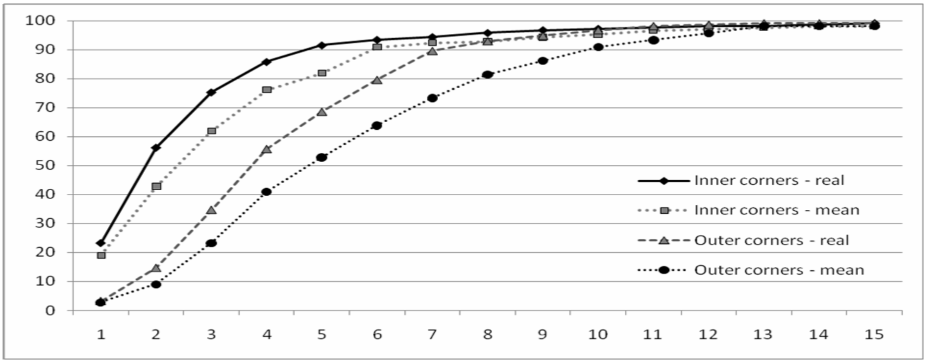


Fig. 7. Eye corner detection rates for different thresholds – Approach #2

Table 3. Success rates – Approach #2

Method	Threshold	Success rates for inner corners	Success rates for outer corners
[11]	5%	95.7%	93%
Our method (using mean)	5%	91.43%	68.57%
Our method (using real)	5%	81.90%	52.86%
Our method (using mean)	10%	97.14%	96.67%
Our method (using real)	10%	95.24$	90.95%

In both ways, it is shown that the algorithm performs better for the inner eye corners. This is because closer to the inner eye corners, the eyelid contours are more prominent than the ones around the outer eye corners.

6 Conclusion

In this paper, an accurate method for automatic detection of eye features is presented. Firstly, the iris position and radius is extracted by using the edges in the non-skin region in the upper half of the face and then, refined by using vertical edges only in a smaller window. Afterwards, also with the help of the previously extracted iris, edges which are formed by the eyelids are detected and two curves for upper and lower eyelids are fitted. Finally, the intersections of these two curves are labeled as the eye corners. Experimental results demonstrated that high accuracy can be achieved with the proposed algorithm. In Fig. 8, some extraction results are illustrated.

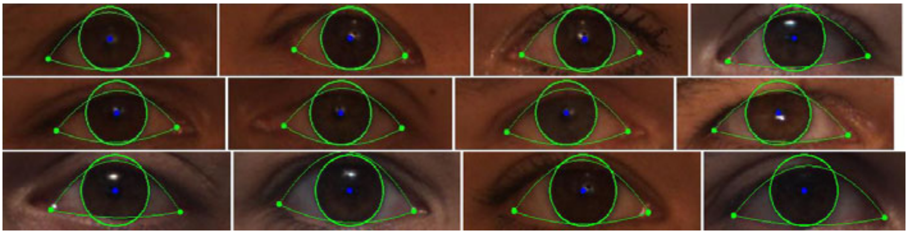


Fig. 8. Some extraction results for iris center and border, eyelid contours and eye corners

References

1. Brunelli, R., Poggio, T.: Face recognition: features versus templates. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(10), 1042–1052 (1993)
2. Zheng, Z., Yang, J., Yang, L.: A robust method for eye features extraction on color image. Pattern Recognition Letters 26(14), 167–8655 (2005) ISSN 0167-8655
3. Khairosfaizal, W.M.K.W.M., Nor'aini, A.J.: Eyes detection in facial images using Circular Hough Transform. In: 5th International Colloquium on Signal Processing & Its Applications, CSPA 2009, March 6-8, pp. 238–242 (2009)
4. Pardas, M.: Extraction and tracking of the eyelids. In: Proceedings of 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP 2000, vol. 6, 4 pp. 2357–2360 (2000)
5. Park, C.W., Kwak, J.M., Park, H., Moon, Y.S.: An Effective Method for Eye Detection Based on Texture Information. In: International Conference on Convergence Information Technology 2007, November 21-23, pp. 586–589 (2007)
6. Guan, Y.: Robust Eye Detection from Facial Image based on Multi-cue Facial Information. In: IEEE International Conference on Control and Automation, ICCA 2007, May 30-June 1, pp. 1775–1778 (2007)
7. Kuo, P., Hannah, J.: An improved eye feature extraction algorithm based on deformable templates. In: IEEE International Conference on Image Processing, ICIP 2005, September 11-14, vol. 2, p. II-1206-9 (2005)

8. Wang, P., Green, M.B., Ji, Q., Wayman, J.: Automatic Eye Detection and Its Validation. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPR Workshops, June 25, pp. 164–164 (2005)
9. Zhou, Z.-H., Geng, X.: Projection functions for eye detection. Pattern Recognition 37(5), 1049–1056 (2004) ISSN 0031-3203
10. Feng, G.: Variance projection function and its application to eye detection for human face recognition. Pattern Recognition Letters 19(9), 899–906 (1998)
11. Haiying, X., Guoping, Y.: A Novel Method for Eye Corner Detection Based on Weighted Variance Projection Function. In: 2nd International Congress on Image and Signal Processing, CISP 2009, October 17-19, pp. 1–4 (2009)
12. Xu, C., Zheng, Y., Wang, Z.: Semantic feature extraction for accurate eye corner detection. In: 19th International Conference on Pattern Recognition, ICPR 2008, December 8-11, pp. 1–4 (2008)
13. Yin, L., Basu, A.: Integrating active face tracking with model based coding. Pattern Recognition Letters 20(6), 651–657 (1999) ISSN 0167-8655
14. Vezhnevets, V., Degtiareva, A.: Robust and Accurate Eye Contour Extraction. In: Proc. Graphicon 2003, Moscow, Russia, September 2003, pp. 81–84 (2003)
15. Tse, K.W., Lau, W.H., Leung, S.H., Liew, A.W.C.: Eye extraction using spatial fuzzy clustering method. In: Proceedings of 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering TENCON 2002, October 2002, vol. 1, pp. 515–518 (2002)
16. MATLAB Central Program for Color Image Segmentation – Athi Narayanan S, K.S.R. College of Engineering, Erode, Tamil Nadu, India, `http://www.mathworks.com/matlabcentral/`
17. Savran, N., Alyüz, H., Dibeklioğlu, O., Çeliktutan, B., Gökberk, B., Sankur, L.A.: Bosphorus Database for 3D Face Analysis. In: The First COST 2101 Workshop on Biometrics and Identity Management (BIOID 2008), May 2008, Roskilde University, Denmark (2008)

An Empirical Comparison of Kernel-Based and Dissimilarity-Based Feature Spaces*

Sang-Woon Kim[1] and Robert P. W. Duin[2]

[1] Dept. of Computer Science and Engineering,
Myongji University, Yongin, 449-728 South Korea
kimsw@mju.ac.kr

[2] Faculty of Electrical Engineering, Mathematics and Computer Science,
Delft University of Technology, The Netherlands
r.p.w.duin@tudelft.nl

Abstract. The aim of this paper is to find an answer to the question: What is the difference between dissimilarity-based classifications(DBCs) and other kernel-based classifications(KBCs)? In DBCs [11], classifiers are defined among classes; they are not based on the feature measurements of individual objects, but rather on a suitable dissimilarity measure among them. In KBCs [15], on the other hand, classifiers are designed in a high-dimensional feature space transformed from the original input feature space through kernels, such as a Mercer kernel. Thus, the difference that exists between the two approaches can be summarized as follows: The *distance* kernel of DBCs represents the discriminative information in a relative manner, i.e. through pairwise dissimilarity relations between two objects, while the *mapping* kernel of KBCs represents the discriminative information uniformly in a fixed way for all objects. In this paper, we report on an empirical evaluation of some classifiers built in the two different representation spaces: the dissimilarity space and the kernel space. Our experimental results, obtained with well-known benchmark databases, demonstrate that when the kernel parameters have not been appropriately chosen, DBCs always achieve better results than KBCs in terms of classification accuracies.

Keywords: kernel-based classifications (KBCs), dissimilarity-based classifications (DBCs), representation spaces, classification accuracies.

1 Introduction

Various kernel methods have been successfully used in the last decade to tackle complicated classification problems by a nonlinear mapping from the original input space to a kernel feature space [15]. Every learning algorithm that only makes use of inner products between data vectors can be transformed into a kernel method by means of replacing the inner product with an arbitrary kernel function [6]. The kernel function

* The work of the first author was partially done while visiting at Delft University of Technology, The Netherlands. We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). This work was supported by the National Research Foundation of Korea funded by the Korean Government (NRF-2009-0071283).

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 559–568, 2010.

is typically viewed as providing an implicit mapping of sample points into a high-dimensional space, with the ability to gain much of the power of that space without paying the computational penalty[1]. Formally, let $\mathcal{X}$ denote the original pattern space and $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a function mapping pairs of patterns to real numbers. If the function k satisfies the condition of positive definiteness, there exists a vector space $\mathcal{F}$ and a mapping from $\mathcal{X}$ to $\mathcal{F}$, such that k acts as a dot product in $\mathcal{F}$ [15]. Such functions, k, are commonly called *kernel functions*.

The most popular representatives of kernel methods are support vector machines (SVMs) for classification problems [15]. SVMs are hyperplane classifiers in implicitly defined Euclidean feature spaces. A large number of applications reported in the literature indicate that SVMs are able to generalize well from unseen data and are not prone to overfitting. Other kernel methods for solving feature extraction and classification include principal component analysis [13], Fisher discriminant analysis [3], CLAFIC (CLAss Featuring Information Compression) [1], Gaussian mixture modeling [17], canonical correlation analysis [7], subspace discriminant analysis [4], locally linear embedding [16], and many others [15]. In the interest of brevity, the details of these kernel methods are omitted here, but can be found in the corresponding literature.

On the other hand, Duin and his co-authors [11], [12] proposed an alternative object representation system based on dissimilarities between objects using a generalized kernel approach. The concept of dissimilarity-based classifications is a way of defining classifiers between the classes, which are not based on the feature measurements of the individual patterns, but rather on a suitable *dissimilarity measure* between them. Here, the dissimilarity measure can be defined for not only vectorial inputs, but also arbitrary non-vectorial patterns, such as strings, graphs, shapes, probabilistic models, etc. [9] Thus, this methodology can be considered a unified approach to statistical and structural pattern recognition [5], [9]. Furthermore, the advantage of such a paradigm is that it does not have to confront the problems associated with feature spaces, such as the *curse of dimensionality* and the issue of estimating a number of parameters [8].

In general, the kernels are understood as symmetric, positive definite functions of two variables, and, thereby, they express similarity between two objects represented in a feature space [15]. From this perspective, it is possible to regard a kernel as defining a *similarity measure* between the two variables. On the other hand, in [11], the kernels are addressed in a more general way, i.e., as a proximity measure. The important difference between these two types of kernels is summarized as follows: The *distance* dissimilarity kernel represents the information in a relative manner, i.e., through pairwise dissimilarity relations between the two objects; the *mapping* similarity kernel represents the information uniformly in a fixed way for all of the available objects.

Although classifications based on similarity kernels (which are referred to as kernel based classifications or KBCs) and classifications based on dissimilarity kernels (dissimilarity based classifications or DBCs) have been explored separately by many researchers, not much analysis has been done comparing the two. Therefore, the aim of this paper is to find an answer to the question: What is the difference between KBCs

[1] In the contrary of mapping objects into a high-dimensional space, a kernel function can also be viewed as a mapping to a low-dimensional space. The details of this kind of kernel method are omitted here, but can be found in [2].

and DBCs? or, more specifically, How different are these systems in their classification accuracies?

In this paper, we report an empirical comparison of KBCs and DBCs, which are built in two different representation feature spaces, respectively: dissimilarity-based feature spaces and kernel-based feature spaces[2]. Although it is hard quantitatively to evaluate the various KBC and DBC schemes, we have attempted to do exactly this. To achieve this goal, we have done a number of experiments with different methods to render this comparative study more complete. In KBCs, all samples are mapped to a higher-dimensional feature space using a kernel function; traditional classifications are then performed in the transformed feature space. In DBCs, on the other hand, dissimilarity-based feature spaces are directly obtained from all of the available objects; the same classifications are then done in the transformed feature space. Our experimental results obtained with well-known benchmark databases demonstrate that the classification performances obtained with KBCs and DBCs are almost the same. However, when the kernel parameters have not been appropriately chosen, it seems that DBCs are better than KBCs in terms of classification accuracy.

The main contribution of this paper is to demonstrate that the discriminative information of the dissimilarity-based feature space is less sensitive than that of the kernel-based feature space in choosing function parameters. This realization has been gained by executing classifications in the two feature spaces obtained with the training data sets and by comparing their strengths in terms of classification accuracy. Although many researchers have investigated the fact that SVMs are vulnerable to function parameters, to the best of our knowledge there is currently no reported empirical comparison of kernel-based and dissimilarity-based feature spaces.

2 Related Work

Kernel-Based Classifications (KBCs): In the implementation of kernel methods, the data is processed using a kernel to create a kernel matrix, which in turn is processed by a learning algorithm to produce a pattern function. This function is used to recognize unseen examples. Here, it is interesting to note that the resulting systems are modular: any kernel can be combined with any learning algorithm and vice versa [15].

Consider an embedding map $\phi : x \in \mathbb{R}^d \longmapsto \phi(x) \in \mathcal{F}$, where the choice of the map, ϕ, aims to convert the nonlinear relations into linear ones. Given a kernel and a training set, we can form a matrix known as a kernel matrix or Gram matrix, a matrix containing the evaluation of the kernel function on all pairs of data points [15]. To put it concretely, given a set of vectors $T = \{x_i\}_{i=1}^{n}, x_i \in \mathbb{R}^d$, the kernel matrix, K, is defined as the $n \times n$ matrix whose entries are $K_{ij} =< x_i, x_j >$. If we are using a kernel function, k, to evaluate the inner products in a feature space, $\mathcal{F}$, with a feature map, ϕ, the associated Gram matrix has entries: $K_{ij} =< \phi(x_i), \phi(x_j) >= k(x_i, x_j)$. Here, the Gram matrix, which is defined as a *kernel-based feature space*, is *positive semi-definite* (for details, see Proposition (3.7) of [15]).

[2] In this paper, we use the term 'feature space' for what we have called a vector space in pattern recognition unless otherwise mentioned.

The overall procedure for KBCs is summarized as follows:

1. Compute a kernel matrix, K, using a given training data set, $T = \{x_i\}_{i=1}^{n}$, and a kernel function, $k(\cdot, \cdot)$;

2. Compute the normalized eigenvectors of $K \in \mathbb{R}^{n\times n}$ in $\mathcal{F}$, and select a subspace dimension, q, to generate a transformation matrix, $A \in \mathbb{R}^{n\times q}$;

3. For a testing object, we compute a projection of the object onto the subspace using the transformation matrix A;

4. Achieve the classification through invoking a classifier built in the transformed subspace obtained with A and operating on the projected vector.

In the above algorithm, the kernel functions, $k(x_i, x_j)$, for example, such as *Polynomial*, *Radial basis*, or *Minkowski* function, can be defined as follows: $(x_i^T x_j + 1)^p$, $exp\left(-||x_i - x_j||^2)/p^2\right)$, or $\left(\sum |x_i - x_j|^p\right)^{1/p}$. Here, p's are the function parameters, such as function degree (d), standard deviation (σ), and degree order ($p \geq 1$), respectively. Among these kernels, the *Radial basis* function is the most widely used and has been extensively studied in this field. The parameter σ controls the flexibility of the kernel in a way similar to that of the degree d in the *Polynomial* kernel.

Dissimilarity-Based Classifications (DBCs): A dissimilarity representation of a set of samples, $T = \{x_i\}_{i=1}^{n} \in \mathbb{R}^{d\times n}$, is based on pairwise comparisons and is expressed, for example, as an $n \times m$ dissimilarity matrix $D_{T,Y}[\cdot, \cdot]$, where $Y = \{y_j\}_{j=1}^{m} \in \mathbb{R}^{d\times m}$, a prototype set, is extracted from T, and the subscripts of D represent the set of elements on which the dissimilarities are evaluated. Thus, each entry, $D_{T,Y}[i, j]$, corresponds to the dissimilarity between the pairs of objects, $\langle x_i, y_j\rangle$, where $x_i \in T$ and $y_j \in Y$.

Here, the dissimilarity matrix, $D_{T,Y}[\cdot, \cdot] \in \mathbb{R}^{n\times m}$, is defined as a *dissimilarity-based feature space*, on which the d-dimensional object, x, given in the feature space, is represented as an m-dimensional vector $\delta(x, Y)$, where if $x = x_i$, $\delta(x_i, Y)$ is the i-th row of $D_{T,Y}[\cdot, \cdot]$. In this paper, the dissimilarity matrix $D_{T,Y}[\cdot, \cdot]$ and the column vector $\delta(x, Y)$ are simply denoted by $D(T, Y)$ and $\delta_Y(x)$ (or $D(x, Y)$), respectively. Here $\delta_Y(x)$ is an m-dimensional vector, while x is d-dimensional.

A conventional algorithm for DBCs is summarized in the following:

1. Select the representative set Y from the training set T by resorting to a selection method, such as *Random*, *RandomC*, or *KCentres* algorithm, as described in [11];

2. Compute the matrix $D(T, Y)$, using T, by employing a measuring system, such as the Euclidean distance, $d_E = ((x - y)^T(x - y))^{1/2}$, for all $x \in T$ and $y \in Y$;

3. For a testing sample z, compute a dissimilarity column vector, $\delta_Y(z)$, by using the same measure used in Step 2;

4. Achieve the classification through invoking a classifier built in the dissimilarity space and operating on the dissimilarity vector $\delta_Y(z)$.

In the above two algorithms, the dimensions of the two classification spaces can be reduced with the cardinality of the representation set and the number of the chosen eigenvectors, respectively. However, to reduce the computational complexity of this experiment, we first construct the dissimilarity matrix D and the kernel matrix K with respect to *all* the training samples. Then, we reduce the dimensionality of the spaces by performing a principal component analysis (PCA).

Kernel Matrix Versus Dissimilarity Matrix: Assume a training set T of n samples, a prototype set Y of m samples, and a nonnegative dissimilarity measure d. Then, an object, x, is represented as a dissimilarity vector of $D(x, Y) = [d(x, y_1), \cdots, d(x, y_m)]^T$. If a similarity measure k is used instead, we will get a similarity representation defined by similarity vectors of $K(x, Y) = [k(x, y_1), \cdots, k(x, y_m)]^T$. Here, if $|T| = |Y|$ and k is *positive semi-definite*, then K is a kernel matrix [12].

If the dissimilarity d is designed first, then k is defined as follows: $k(x_i, y_j) = \frac{1}{2}\left(d^2(x_i, 0) + d^2(0, y_j) - d^2(x_i, y_j)\right)$, where 0 represents a specific element that acts as a reference. On the other hand, if the similarity k is defined first, then d is computed as follows: $d^2(x_i, y_j) = k(x_i, x_i) + k(y_j, y_j) - 2k(x_i, y_j)$. In the interest of compactness, the details of the derivation are omitted here, but can be found in the literature [6],[15].

Kernel methods are powerful, but often cannot handle arbitrary proximities without incorporating necessary corrections, such as Euclidean corrections [12]. For example, a symmetric dissimilarity matrix $D(T, T) \in \mathbb{R}^{n \times n}$ can be embedded in a pseudo-Euclidean space by an isometric mapping [12]. The pseudo Euclidean space $\mathcal{E}(= \mathbb{R}^{(p,q)} = \mathbb{R}^{(p)} \oplus \mathbb{R}^{(q)})$ is denoted with signature (p, q), where the bilinear, but not necessarily positive definite, inner product is defined as $< z, z' >_{pE} := z^T M_{pq} z'$, where M_{pq} is $diag(\mathbf{1}_p, -\mathbf{1}_q)$ and $\mathbf{1}_n$ is an n-element vector of 1's. Also, the squared dissimilarity distance, $\|z - z'\|^2_{pE}$, may not define a metric, as it can violate the triangle inequality. That is, the squared norm and the squared distance can be negative in contrast to the Euclidean case. The details of determining the pseudo-Euclidean space to refine the dissimilarity representation are omitted here, but can be found in the literature, including [11] and [12].

3 Experimental Results

Experimental Data: The two classifying approaches, DBCs and KBCs, have been implemented and compared. This was done by performing experiments on three well-known benchmark image databases, namely Nist38, RoadSign, and Kimia2. The data set captioned "Nist38", chosen from the NIST database [18], consists of two kinds of digits, 3 and 8, for a total of 1000 binary images. The size of each image is 32×32 pixels, for a total dimensionality of 1024 pixels. The data set described as "RoadSign" consists of gray-level images of circular road signs: Three hundred road signs and the same number of outlier images [10], in which each image is 32×32 pixels, for a total dimensionality of 1024 pixels. The data set named "Kimia2" consists of two groups of images, each of 9 categories of 12 objects, obtained from the Kimia database [14]. The size of each image is 64×64 pixels, for a total dimensionality of 4096 pixels.

Experimental Method: In this experiment, first, data sets are split into training sets and test sets in the ratio of 75 : 25. Then, the training and testing procedures are repeated 30 times and the results obtained are averaged. Also, in contrast with many other papers on dissimilarities, we start by a feature representation and not with given dissimilarities between raw objects. That is because we want to make a comparison with kernels that also start in the feature space.

To evaluate DBCs and KBCs, different classifiers, such as k-nearest neighbor classifiers, linear Bayes normal classifier, quadratic Bayes normal classifier, and support vector classifier, are employed and implemented with PRTools[3], and will be denoted as *knnc*, *ldc*, *qdc*, and *svc*, respectively, in subsequent sections.

In DBCs, the Euclidean distance between two samples is computed to measure their dissimilarity. Also, in KBCs, three mapping functions, *Polynomial*, *Radial basis*, and *Minkowski*, are employed as kernel functions. However, it is well known that selecting a proper kernel parameter with good class separability plays a significant role in kernel-based algorithms. In this experiment, therefore, to find optimal or near-optimal kernel parameters, in the case of the polynomial function, five function degrees, $p = \{s | s = 1, 2, \cdots, 5\}$, are tested. Then, in the case of the Minkowski function, five l_p distances, $p = \left\{2^{(s-1)} | s = 1, 2, \cdots, 5\right\}$, are examined. Finally, for the radial basis function, five deviation values, $p = \{\sigma_o(1.2 - 0.2s) | s = 1, 2, \cdots, 5\}$, are investigated. Here σ_o is determined after estimating the performance of the classifiers through cross-validation.

Experimental Results: The run-time characteristics of the DBC and KBC schemes for the experimental databases are reported below. First, the experimental results obtained with *qdc* and *ldc* trained in the dissimilarity space (shortly D) and the polynomial kernel space (shortly K) were probed into. Fig. 1 shows a 3-dimensional comparison of the error rates of *qdc* trained in the D and K spaces for Nist38. Here, x, y, and z axes are those of dimensions (which are obtained with PCA), kernel parameters (the degrees of the polynomial function), and the estimated error rates, respectively.

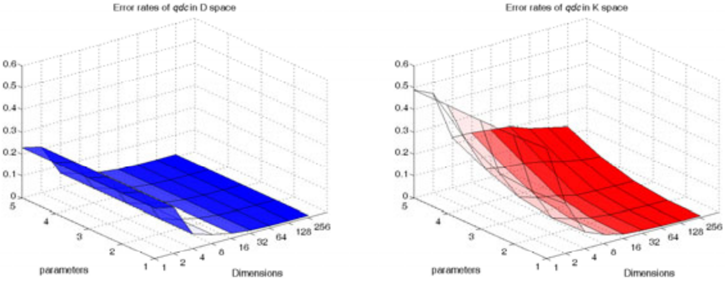


Fig. 1. A 3D comparison of the error rates of *qdc* for Nist38: (a) left and (b)right; (a) and (b) are obtained in D and K spaces, respectively, with different degrees of the polynomial function

From the figure, it can be observed that the two error rates obtained in D and K spaces are different, which implies that selecting an appropriate kernel parameter is essential for KBCs. This characteristic can be observed again in a subsequent experiment.

In principle, the quadratic Bayesian classifier could be better than the linear Bayesian classifier, but it requires far more training samples for estimation of the class covariance matrices. It is also well known that for 2-class problems with equally distributed samples, the quadratic classifier is equivalent to the linear one. Fig. 2 shows a comparison of the error rates of *qdc* and *ldc* trained in D and K spaces for Nist38.

[3] PRTools is a Matlab toolbox for pattern recognition(refer to http://prtools.org/).

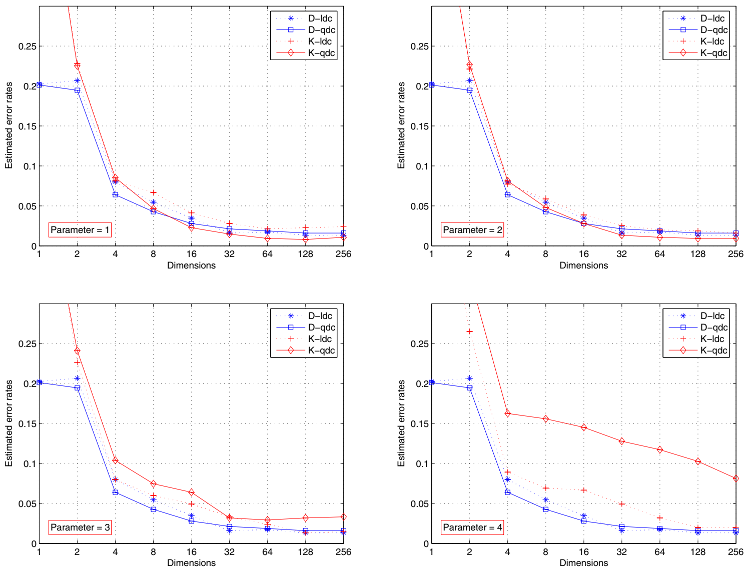


Fig. 2. A comparison of the error rates of *ldc* and *qdc* for Nist38: (a) top left, (b) top right, (c) bottom left, and (d) bottom right; (a) - (d) are obtained in D and K spaces with the four polynomial kernel parameters (degrees) of $s = 1$, 2, 3, and 4, respectively

In the figure, it should be pointed out that the difference in the estimated error rates between *qdc* and *ldc* for Nist38 increases as the value of the parameter increases. This is clearly shown in the error rates represented with two red lines (dashed and solid) in the four pictures of Fig. 2. This comparison shows that the classification accuracy of *qdc* is marginally higher than that of *ldc* when the appropriate parameter is present (refer to Fig. 2 (a) and (b)). However, the situation changes when an inappropriate parameter is chosen (refer to Fig. 2 (d)). From this consideration, the reader should again observe that choosing an appropriate kernel parameter plays an important role in KBCs. The same characteristic could also be seen in the other databases, such as RoadSign and Kimia2. The details for the results of these databases are omitted here to avoid repetition.

Second, as the main result, to investigate the difference of DBCs and KBCs further, the experiment (of estimating error rates) was repeated in other kernel spaces, such as *Polynomial*, *Radial basis*, and *Minkowski* spaces (which are shortly referred to as K_p, K_r, and K_m, respectively). Graphical comparisons of the error rates of the four classifiers trained in the dissimilarity based and the kernel based feature spaces are continually presented. Fig. 3 shows a comparison of the error rates of *knnc*, *ldc*, *qdc*, and *svc*, respectively, for Nist38.

The observations obtained from the figures are the following: (1) In general, the error rates of the classifiers trained in D space decrease constantly as the dimension increases, while those of the classifiers trained in $3K$'s spaces strongly depend on the kernel parameters. (2) As can be observed in the pictures in the left column of Fig. 3, when choosing an appropriate function parameter, all of the classifiers built in D and

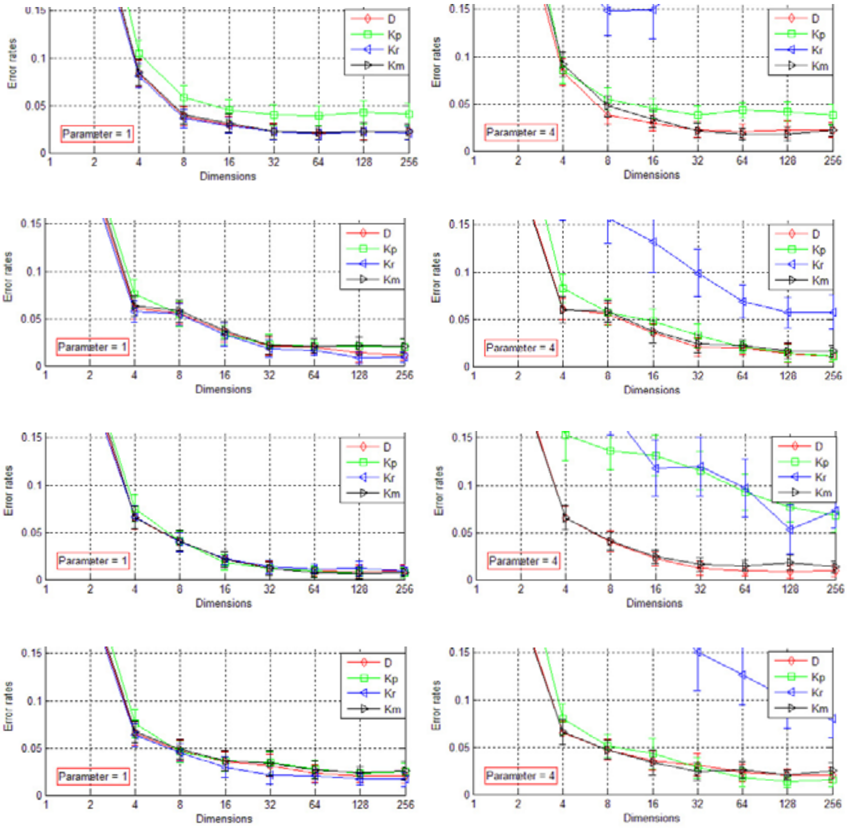


Fig. 3. A comparison of the error rates of *knnc*, *ldc*, *qdc*, and *svc* built in D and $3K$'s spaces with the kernel parameters of 1 and 4 for Nist38: (a) top left, (b) top right, $\cdots$, (g) bottom left, and (h) bottom right; (a) - (b) are of *knnc*, (c) - (d) are of *ldc*, (e) - (f) are of *qdc*, and (g) - (h) are of *svc*

$3K$'s have *almost* the same classification accuracies. (3) Specifically, the classification accuracy of *svc* is the best one obtained in K_r space. However, the classifier does not work satisfactorily in the kernel-based feature space with a wrong parameter, i.e., $s = 4$. (4) When the chosen parameters are far from optimal, the ranking of the discriminative power of the kernel-based feature space is K_m, K_p, and K_r. That is, the best discriminative power is that of K_m, while the worst one is that of K_r. The same characteristic could also be observed in the other databases, such as RoadSign and Kimia2. The details for the results of these databases are omitted here again in the interest of compactness.

Finally, it is an interesting issue to observe how robust to noise the classifiers trained in D and $3K$'s spaces are. To find reason for this phenomenon, we assume that the sample x_i is obtained by a noisy perturbation on the sample. This perturbation can be perceived as the inclusion of some additional noise θ[4], and, thus, we write: $x_i \leftarrow x_i + \theta$.

[4] $\theta(\cdot)$ refers to the noise generation random variables.

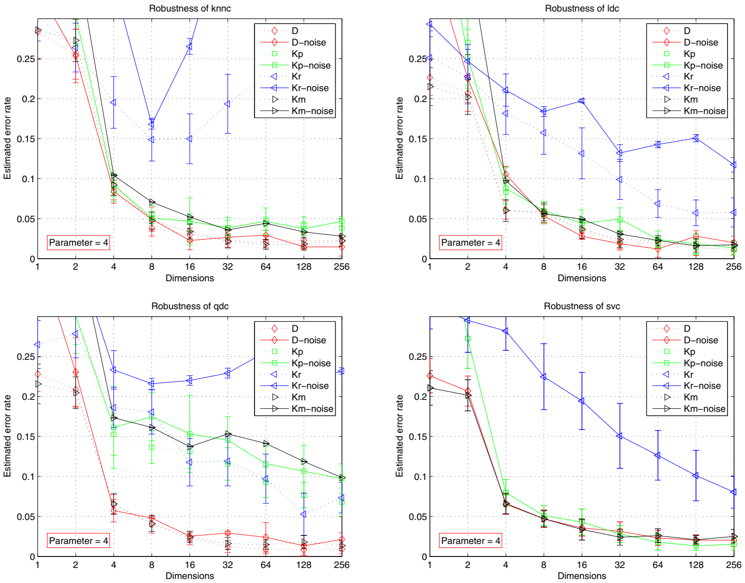


Fig. 4. A comparison of the error rates of *knnc*, *ldc*, *qdc*, and *svc* for the noisy Nist38: (a) top left, (b) top right, (c) bottom left, and (d) bottom right; (a) - (d) are obtained in D and $3K$'s spaces with kernel parameter "4"

For example, the noisy data can be obtained as: $x_i \leftarrow x_i * (1 + \epsilon * rand)$; Here, the function *rand* is to generate an array of random numbers whose elements are normally distributed with mean 0 and variance 1; ϵ is an experimental constant. Fig. 4 shows a comparison of the error rates of *knnc*, *ldc*, *qdc*, and *svc* trained in D and $3K$'s for the noisy Nist38. Here, $\epsilon = 0.3$.

From the figure, it should be observed that the differences in the estimated error rates of DBCs and KBCs obtained from the originally transformed feature space and their noisy perturbation spaces are different. This is clearly shown in the error rates of *qdc* represented with two red lines (dashed and solid lines of $\diamond$ marker) and two blue lines (dashed and solid lines of $\triangleleft$ marker) in Fig. 2(c). From this consideration, the reader should observe that the robustness of DBCs is higher than that of KBCs when there is a badly chosen parameter.

4 Conclusions

In this paper, we performed an empirical comparison of kernel-based classifications (KBCs) and dissimilarity-based classifications (DBCs). A number of classifiers designed in the two feature spaces were tested on well-known benchmark databases, and the classification accuracies obtained were compared. Our experimental results demonstrated that the classification accuracies obtained with KBCs and DBCs were almost the same when there was an appropriate kernel parameter. However, when the parameter was not chosen appropriately, it seemed that the accuracies of DBCs were better than those of the KBCs. Especially, the results demonstrated that support vector classifiers of KBCs were vulnerable to function parameters. Despite this success, problems remain to be addressed. First, in this comparison we employed only three real life databases, in which each feature component of all objects was uniformly distributed in a fixed manner. Thus, evaluating the dissimilarity relations represented in a relative way is an avenue for future work. Next, to improve the internal consistency of the representation matrices, we could correct the matrices using pseudo-Euclidean embedding algorithms. Therefore, the problem of investigating the embedding algorithms developed for KBCs and DBCs remains to be done. Future research will address these concerns.

References

1. Balachander, T., Kothari, R.: Kernel based subspace pattern recognition. In: Proc. of Int'l Joint Conference on Neural Networks, Washington DC, USA, vol. 5, pp. 3119–3122 (1999)
2. Balcan, M.-F., Blum, A., Vempala, S.: Kernels as features: On kernels, margins, and low-dimensional mappings. Machine Learning 65, 79–94 (2006)
3. Baudat, G., Anouar, F.: Generalized discriminant analysis using a kernel approach. Neural Comput. 12(10), 2385–2404 (2000)
4. Chen, B., Yuan, L., Liu, H., Bao, Z.: Kernel subclass discriminant analysis. Neurocomputing 71, 455–458 (2007)
5. Goldfarb, L.: A unified approach to pattern recognition. Pattern Recogni. 17, 575–582 (1984)
6. Haasdonk, B.: Feature space interpretation of SVMs with indefinite kernels. IEEE Trans. Pattern Anal. and Machine Intell. 25(5), 482–492 (2005)
7. Hardoon, D.R., Szedmak, S., Shawe-Taylor, J.: Canonical correlation analysis: An overview with application to learning methods. Neural Comput. 16, 2639–2664 (2004)
8. Kim, S.-W., Oommen, B.J.: On using prototype reduction schemes to optimize dissimilarity-based classification. Pattern Recognition 40, 2946–2957 (2007)
9. Neuhaus, M., Bunke, H.: Edit distance-based kernel functions for structural pattern classification. Pattern Recognition 39, 1852–1863 (2006)
10. Paclik, P., Novovicova, J., Somol, P., Pudil, P.: Road sign classification using Laplace kernel classifier. Pattern Recognition Lett. 21(13-14), 1165–1173 (2000)
11. Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition: Foundations and Applications. World Scientific Publishing, Singapore (2005)
12. Pekalska, E., Duin, R.P.W.: Beyond traditional kernels: Classification in two dissimilarity-based representation spaces. IEEE Trans. Sys. Man, and Cybern(C) 38(6), 727–744 (2008)
13. Schölkopf, B., Smola, A.J., Müller, K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. 10, 1299–1319 (1998)
14. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing shock graphs. In: Proc. of 8th IEEE Int'l Conf. on Computer Vision, Vancouver, Canada, pp. 755–762 (2001)
15. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
16. Tsagaroulis, T., Hamza, A.B.: Kernel locally linear embedding algorithm for quality control. In: Sobh, T., Elleithy, K., Mahmood, A., Karim, M.A. (eds.) Novel Algorithms and Techniques in Telecommunications, Automation and Industrial Electronics, pp. 1–6. Springer, Heidelberg (2008)
17. Wang, J., Lee, J., Zhang, C.: Kernel Trick Embedded Gaussian Mixture Model. In: Gavaldá, R., Jantke, K.P., Takimoto, E. (eds.) ALT 2003. LNCS (LNAI), vol. 2842, pp. 159–174. Springer, Heidelberg (2003)
18. Wilson, C.L., Garris, M.D.: Handprinted Character Database 3, Technical report, National Institute of Standards and Technology, Gaithersburg, Maryland (1992)

The Dissimilarity Representation as a Tool for Three-Way Data Classification: A 2D Measure

Diana Porro-Muñoz[1,2], Robert P.W. Duin[2], Mauricio Orozco-Alzate[3],
Isneri Talavera[1], and John Makario Londoño-Bonilla[4]

[1] Advanced Technologies Application Center (Cenatav), Cuba
{dporro,italavera}@cenatav.co.cu
[2] Pattern Recognition Lab, TU Delft, The Netherlands
r.duin@ieee.org
[3] Universidad Nacional de Colombia Sede Manizales, Colombia
morozcoa@bt.unal.edu.co
[4] Instituto Colombiano de Geología y Minería (Ingeominas), Colombia
jmakario@ingeominas.gov.co

Abstract. The dissimilarity representation has demonstrated advantages in the solution of classification problems. Meanwhile, the representation of objects by multi-dimensional arrays is necessary in many research areas. However, the development of proper classification tools that take the multi-way structure into account is incipient. This paper introduces the use of the dissimilarity representation as a tool for classifying three-way data, as dissimilarities allow the representation of multi-dimensional objects in a natural way. As an example, the classification of three-way seismic volcanic data is used. A comparison is made between dissimilarity measures used in different representations of the three-way data. 2D dissimilarity measures for three-way data can be useful.

Keywords: Object representation, classification, multi-dimensional data, dissimilarity representation.

1 Introduction

In many research areas e.g. chemometrics, image analysis, signal analysis, objects obtained from measurement equipments are represented by multi-dimensional arrays instead of a vector of features. Consequently, the variables from one dimension of the array can be related and analyzed together with the variables of the other dimensions. The structure in which a set of objects with this representation is organized is called multi-way data.

Multi-way data analysis [1, 2] is the extension of multivariate analysis when the analyzed data is arranged in this multi-way structure. However, the most common is the three-dimensional array. The analysis of such data is often used for extracting specific information and exploring the interrelations in the data. It has been shown that this data may not be analyzed optimally by two-way analysis, because it does not respect the multi-way design. Nevertheless, most of the applications and methods for multi-way analysis are for exploratory and

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 569–578, 2010.

regression purposes. Classification has been studied much less. This might be caused by the lack of classification tools able to operate on multi-dimensional spaces and taking all the information available into account.

In recent studies [3, 4, 5], the advantage of learning from dissimilarities between the objects instead of traditional features has been shown, in what is known as the Dissimilarity Representation (DR) [3]. This representation was mainly designed for classification. It is based on the important role that pairwise dissimilarities between objects play. Classifiers may be built in the dissimilarity space generated by a representation set. In this way, the geometry and the structure of a class are determined by the user defined dissimilarity measure, in which application background information may be expressed. It is important to remark that, any traditional classifier that operates in feature spaces can also be used in the dissimilarity space.

In this paper, we introduce the use of the DR as a tool for classifying three-way data in such a way that, objects are analyzed in their 2D representation. Thus, the relations between the objects are analyzed in the dissimilarity space. Moreover, the relationship between the dimensions can be included if the proper dissimilarity measure is selected. The key in this process is to find the dissimilarity measure that takes into account the information embedded in the data. Information about the data that is missing in the actual representation e.g. shape and connectivity, can also be taken into account in the dissimilarity measure.

Traditionally, signals are analyzed in the time domain or by their spectrum in terms of energy spread over its frequency components (Fourier transform) [6]. Recent studies have also shown that training the classifiers in the dissimilarity space is a feasible and more reliable alternative for automatic classification of seismic signals than the frequency-based one [4]. Nevertheless, these representations alone may not be optimal for seismic signal analysis, since the changes of spectral energy in time are not considered. Due to this limitation, the use of a time-frequency representation like spectrograms or scalograms, may be advantageous. Although these types of 2D object representations are raising popularity for the analysis of seismic signals [7], they have not intensively been exploited as such in automatic classification systems [8].

Hence, although the proposed approach can be applied to any three-way data in the form ($objects \times variables \times variables$), we will based the demonstration of its feasibility on a problem of classifying three-way seismic volcanic signals. With the purpose of comparing how it works for three-way data with different characteristics and some suitable dissimilarity measures, two three-way seismic volcanic data will be generated from the spectrogram and scalogram techniques. A 2D dissimilarity measure is also proposed. Additionally, results are compared with 1D feature representation using the time integrated spectra to show the advantages of the proposed approach in this case.

2 Three-Way Volcanic Data

In several research areas, different multi-way array configurations can be found e.g. several sets of variables measured on different objects. These data would be appropriately represented by higher order generalization of vectors and matrices. However, the most common design would be defined as $\mathbf{Y} \in \mathbb{R}^{n \times m \times l}$. Each horizontal slice $(\mathbf{m} \times \mathbf{l})$ of the block represents the data of one object; each vertical slice $(\mathbf{n} \times \mathbf{l})$ holds for the data of a specific type of variable and the front to back slices $(\mathbf{n} \times \mathbf{m})$, variables of other type.

The two three-way data to be used in this paper correspond to seismic signals from the ice-capped Nevado del Ruiz volcano in the Colombian Andes. This volcano is currently studied by the Volcanological and Seismological Observatory at Manizales. Signals from the Olleta crater station were selected for the experiments. Signals were digitized at 100.16 Hz sampling frequency by using a 12 bit analog-to-digital converter. The a-priori classification of the signals is done by visual inspection. The dataset is composed of 12032-point signals of two classes of volcanic activities: 235 of Long-Period (LP) earthquakes, and 235 of Volcano- Tectonic (VT) earthquakes.

The differences in 1D spectral content of these signals allow the discrimination between the events [9]. That is why spectral-based classification is often used for this type of data. However, with this representation we are not able to analyze how the frequency content changes in time. An intuitive way to represent this time-frequency relationship for all the signals would be in a three-way array $\mathbf{Y}$ as defined above. In the seismic volcanic three-way array configuration ($signals \times time \times frequency$), the signals are organized in the vertical axis, time in the horizontal and frequency in the depth axis.

To obtain the 2D time-frequency representation of each signal we used two techniques. The first one is the Short-Time Fourier Transform (STFT) by which the spectrograms are obtained [6]. With this technique, the time localization can be obtained by windowing the data at different times and computing the Fourier transform on that part of the signal. Consequently, it can be known what frequency intervals are present in a time interval of the signal, but not with much precision. The narrower it is, the better the time resolution will be and the poorer frequency resolution.

Another way to obtain the time-frequency representation of the signals is by the Continuous Wavelet Transforms (CWT), with which the scalograms are computed. This technique is based on the computation of continuous wavelet transforms over the entire signal for different scales [6]. It was originally introduced as a time-scale representation, but it can also be interpreted as a time-frequency representation as scales and frequencies are inversely proportional. In the scalograms we have better time resolution and poorer frequency resolution at high frequencies, and better frequency resolution and poorer time resolution at low frequencies. Consequently, this technique could lead to a more accurate time-frequency description of signals with low and high frequencies, as is the case of the treated data.

3 Dissimilarity Representation from Three-Way Data: A 2D Measure

The Dissimilarity Representation (DR) [3], was proposed as a more flexible representation of the objects than the feature representation, with the purpose of having more information about the structure of the objects. It is seen as a link between the statistical and structural approaches, as both types of patterns can be described by the (dis)similarity measure. The DR is also based on the role that (dis)similarities play in a class composition, where objects from the same class should be similar and objects from different classes should be different (compactness property). Hence, it should be easier for the classifiers to discriminate between them.

Using the DR, classifiers are trained in the space of the proximities between objects, instead of the traditional feature space. Thus, in place of the feature matrix $\mathbf{X} \in \mathbb{R}^{n \times q}$, where $\mathbf{n}$ runs over the objects and $\mathbf{q}$ over the variables, the set of objects is represented by the matrix $\mathbf{D}(\mathbf{X}, \mathbf{R})$. This matrix contains the dissimilarity values $d(x_i, r_j)$ between each object x_i of $\mathbf{X}$ and the objects r_j of the representation set $\mathbf{R}(r_1, ..., r_h)$. We build from this matrix a dissimilarity space. Objects are represented in this space by the column vectors of the dissimilarity matrix. Each dimension corresponds to the dissimilarities with one of the representation objects.

For a t-dimensional array $\mathbf{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_t}$, the theory of the DR is the same. In fact, one of the advantages of the DR is that it can be generated from any representation of the objects e.g. vectors of numbers, graphs, as long as we have a proper dissimilarity measure. This applies also to the multi-way data. Originally, each object is represented by a (t-1)-dimensional array of numerical values and all the objects together conform the t-dimensional array. Hence, to obtain the dissimilarity space, a mapping $\phi(\cdot, R) : \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_{t-1}} \rightarrow \mathbb{R}^h$ is defined, such that for every object y, $\phi(y, R) = [d(y, r_1), d(y, r_2), ..., d(y, r_h)]$. Classifiers are then built in this space, as in any feature space.

The elements of $\mathbf{R}$ are called prototypes, and have preferably to be selected by a prototype selection method [3]. These prototypes are usually the most representative objects of each class, $\mathbf{R} \subseteq \mathbf{X}$ or $\mathbf{X}$ itself, resulting in a square dissimilarity $\mathbf{D}(\mathbf{X}, \mathbf{X})$. $\mathbf{R}$ and $\mathbf{X}$ can also be chosen as different sets. As dissimilarities are computed to $\mathbf{R}$, a dimensionality reduction is reached if a good, small set can be found, resulting in less computationally expensive classifiers.

The issue to be addressed in this problem is how to obtain the dissimilarities from the multi-way representation. Many ideas can arise to do this transformation. Focusing in three-way data we propose as a first approach, to take each object matrix y of $\mathbf{Y}$, and compute the dissimilarities between them by a 2D dissimilarity measure. Some 2D measures have been proposed in [10] for face and palm-print recognition. However, the selection of the suitable measure for the problem at hand is a very important aspect in the DR approach. To deepen in this task we will focus in our case of study on three-way representation of seismic volcanic signals by spectrograms and on scalograms. Thus, each object is represented by a matrix (2D). A comparison is made about the characteristics of each data and the dissimilarity measure to be used. A 2D dissimilarity measure is also proposed.

In many types of data e.g. spectral data, it is necessary to take into account the shape information and connectivity between the measure points. Such is the case of the time-frequency three-way representation where shape changes are present in the spectral (frequency) direction and connectivity in the time direction. When this representation is obtained by scalograms, the CWT already retrieves these functional characteristics from the data. The observations in the signal can be seen as continuous single entities, instead of sets of different variables. Based on the results obtained with the 2D assembled matrix distance (AMD) (See Eq. 1) proposed in [10], it seems to be a good option for this case. As the information to be included about the data can be already found in its representation by wavelets, it might be enough to use this measure.

$$d_{AMD}(y_a, y_b) = \left(\sum_{k=1}^{l} \left(\sum_{j=1}^{m} (y_{a,j,k} - y_{b,j,k})^2 \right)^{p/2} \right)^{1/p} \qquad (1)$$

The weight p is used to emphasize either small or large differences between the elements, in dependence of the problem at hand. If $p < 1$, all the differences will be reduced, thus the larger ones will not interfere much in the measure. On the other hand, if $p > 1$, the larger differences will be more pronounced, resulting in a heavy influence on the measure. However, when the information is not taken into account in the representation of the data, the dissimilarity measure has to take care of it. Thus, considering the results obtained with the Shape measure (manhattan distance on the first Gaussian derivatives) for simple spectra [5], we propose to make use of the derivatives into the AMD measure. In such a way, we can take the ordering information into account as well as the shape of the spectra. A principle of the DR approach is that instead of a single representation of a problem, one may also consider either a complex representation, built from many dissimilarity representations, where different aspects of the data are described in various ways [3]. Based on this and the previously stated, we define the 2DShape dissimilarity measure as follows:

1. Compute the matrix D^1

$$D^1_{a,b} = \left(\sum_{k=1}^{l} \left(\sum_{j=1}^{m} (y^{\sigma}_{a,j,k} - y^{\sigma}_{b,j,k})^2 \right)^{p/2} \right)^{1/p} , \quad y^{\sigma}_{i,j,\cdot} = \frac{\mathrm{d}}{\mathrm{d}_j} G(j, \sigma) * y_{i,j,\cdot}$$

2. Compute the matrix D^2

$$D^2_{a,b} = \left(\sum_{j=1}^{m} \left(\sum_{k=1}^{l} (y^{\sigma}_{a,j,k} - y^{\sigma}_{b,j,k})^2 \right)^{p/2} \right)^{1/p} , \quad y^{\sigma}_{i,\cdot,k} = \frac{\mathrm{d}}{\mathrm{d}_k} G(k, \sigma) * y_{i,\cdot,k}$$

3. Combine both dissimilarities matrices $D = \frac{1}{\omega_1} D^1 + \frac{1}{\omega_2} D^2$

The variables $y_{i,j,\cdot}$ and $y_{i,\cdot,k}$, stand for the k-th columns and the j-th rows of the i-th matrix (object); $\forall i = 1, 2, ..., n$. Their expression correspond to the computation of the first Gaussian derivatives of spectra, where $*$ denotes convolution and σ stands for a smoothing parameter [5]. The dissimilarities in step 1 and step 2 correspond to the first and second directions respectively, as indicated by the notation e.g. spectra and time. In the combination step, we included a weight for scaling. We defined $\omega_c = var(D^c)$, to scale each dissimilarity matrix by its columns (prototypes) variance. This measure can also be used in three-way data where there are no variations in shape in one of the directions. In this case, it is enough to use the AMD measure in step 1 or step 2, such that only the differences in area are compared.

A good example where the proposed measure can be applied is in the time-frequency representation obtained by spectrograms. The connectivity in the time direction is not taken into account as the Fourier transform is computed separately in the different parts of the windowed signal. Besides, instead of having continuous points in time, we have time intervals.

4 Experimental Results and Discussion

To show how the proposed approach works, we selected a data of seismic volcanic signals. We make a comparison between the results with the three-way data obtained by the spectrograms and scalograms. This comparison is not only made in terms of the dissimilarity measures, but in the information we get from the three-way representation. A comparison is also made between the classification on the dissimilarity spaces derived from 2D and the 1D spectral representation of the data. This way, we show the advantages of using the 2D representation over the 1D e.g. time-frequency (spectral) based classification over the spectral-based. The Average Classification Errors (ACE) for the DR on both spectral and three-way data from spectrograms and scalograms are shown, using different sizes of the representation set.

For the experiments, a dataset with 235 objects per class (VT and LP) is considered. For the 1D (spectral) representation we have computed the spectrum by using a 12032-point Fast Fourier Transform (FFT). Thus, the whole signal is analyzed in both 1D and 2D representations. To compute the spectrograms, trying to make a trade-off between time and frequency resolution, a 256-point (windows size) short time Fourier transform was calculated with 50% of overlap. The values for these parameters were selected empirically. However, it is important to determine the best combination for these parameters as they can influence the results. Further research studies should be done in optimizing these parameters and its influence in the solution of the problem. From this technique, we get a 470 × 129 × 93 three-way data.

For the scalograms computation we used the Morlet wavelet, based on the literature [7, 8] and an interactive Matlab tooolbox for the analysis of seismic volcanic data [11]. Taking into account the inversely proportional relation frequency-scales, we selected the scale values related to the major frequency components in the signals. In a previous study on signals of the same volcano and station (although they are not the same samples)[12], the authors concluded that most of the discriminative information is contained between 7.5 Hz and 25

Hz approximately. A narrow band around 40 Hz, associated to an always present peak, was also selected.

However, a 10-component PCA was also made on the spectral representation of the signals and analyzed the modeling power of all the frequencies present. From this analysis we arrived to the same range of frequency values selected in the reference. Nevertheless, some important peaks were also detected from 0.1 Hz to 7 Hz. Hence, range of scales=[1/(0.1:02:2,3:0.4:25,39:0.3:42)] was used to analyze those frequencies. A 470 × 72 × 12032 three-way array was obtained. Before computing the three representations, the raw signals were normalized to zero-mean and unit-variance.

A Fisher Linear classifier was computed in the dissimilarity space. Experiments were repeated 10 times. Training and test objects were randomly chosen from the total data set, in a 10-fold cross-validation process. Different sizes of the representation set $[10, 20, 50, 75, 100, 125, 200, 250, 300]$ were randomly selected. For the generation of the dissimilarity space, the Manhattan (MD), Euclidean (ED) and Shape measures were computed on the spectral representation. These measures have performed well for spectral data [4, 5]. In a 5-times 10-fold cross-validation from a range of values $[1 - 50]$, the best results were achieved with $\sigma = 15$. For the two measures analyzed in Sec. 3, we used values of $p = [0.5, 1, 2]$ so we can investigate the effect of parameter p (small or big differences) in our classification problem. For the spectral direction in the 2DShape measure, we selected $\sigma = 3$ and for the time direction $\sigma = 2$.

It can be observed in Fig. 1 that, the ACE on the dissimilarity space generated from the spectral data is around 25% and 30%. The error values for the Manhattan measure are slightly better than those of the Euclidean and Shape measures. Nevertheless, if the standard deviation is taken into account, the values for the three measures are very similar. The results with the Shape measure (derivative-based) are not as expected (based in previous works). Hence, these results could suggest that there is not more information to be captured from this representation. It is also possible that these measures are not robust enough for this problem, which somehow contradicts the previous studies [4, 5]. Further studies may be done to find a more proper measure for this type of data.

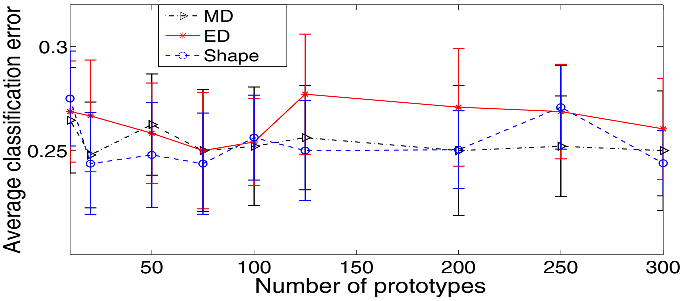


Fig. 1. ACE on the 1D representation for different numbers of prototypes

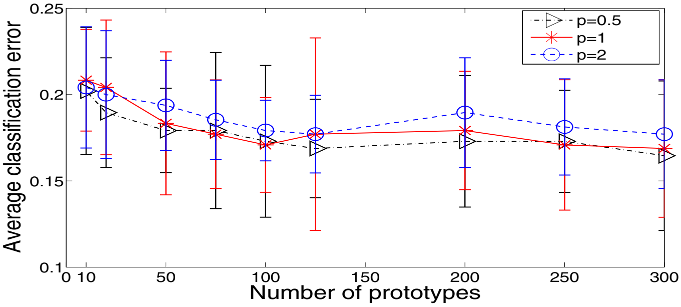


Fig. 2. ACE on the three-way data from spectrograms (2D) for different numbers of prototypes

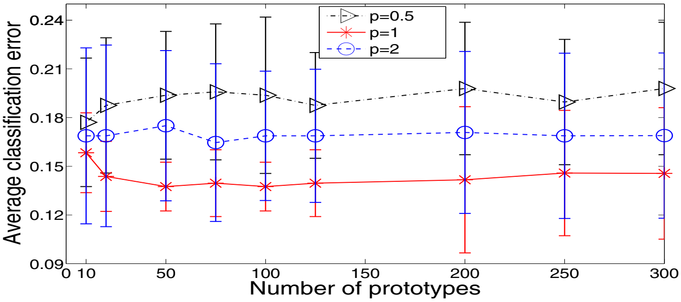


Fig. 3. ACE on the three-way data from scalograms (2D) for different numbers of prototypes

However, when we analyze the error of the DR from the three-way data we see a significant improvement in both Fig. 2 and Fig. 3. This ratifies the fact that the time-frequency relation is more discriminative than the spectra. In the case of the data obtained from the spectrogram, the ACE ranges from 15% to 20%. This also suggests that the proposed 2D measure is capable of capturing the information needed. Nevertheless, if we analyze the ACE of the three-way data obtained from the scalogram, it is slightly better. We can notice that it is also in a range of 13% to 20%, taking into account all the values of p. However, if we analyze only $p = 1$, the largest ACE value is around 15%. These results might be supported by the advantages of the CWT for analyzing this type of seismic volcanic signals. It is also evident that the AMD measure works well for this data, given that the shape and continuity information is already taken into account in the representation by wavelets. It might be possible to obtain better results if more precise scale values are chosen.

The selection of a dissimilarity measure for a representation depends on what we are looking for. In the case of the scalograms, the dissimilarity measure is very simple. However, the computation of the scalogram is really expensive in cases like this, where there are too many important frequency components and the signals are so large. Nevertheless, we cannot forget the advantages of using this technique for the analysis of frequencies in exact moments in time. On the other hand, if we analyze the computation of the spectrograms, it is less expensive than of the scalograms. But, due to the lack of some information in it, a more complex dissimilarity measure is required to include this information. Besides, for the signal analysis it is less precise than the scalograms, as we can only know what intervals of frequencies are present in an interval of time. Thus, it is up to the specialists to decide which of them to use in dependence of their priority.

If we analyze the ACE on the DR from the spectral data and three-way data from scalograms, we can see that from 50 or more prototypes it is approximately stable. The explanation we give to this phenomenon, is that there is no more discriminating information to be found in more prototypes. On the other hand, if we analyze the ACE on the DR from the three-way data from spectrograms, we can see that the behavior is different. While increasing the number of prototypes, the ACE decreases. The more prototypes we add, the more information we have to discriminate between the classes. Nevertheless, due to the so-called peaking phenomenon, when the number of prototypes starts reaching the size of the training set, the errors will increase.

5 Conclusions

We introduced the use of the Dissimilarity Representation as a tool for classifying three-way data. In this approach, objects are analyzed with a 2D representation. The relationship between the different dimensions is analyzed in the 2D dissimilarity measure. Besides, information about the data that is missing in the original representation e.g. shape, can be considered in it. The good performance of classifiers on the 2D representation of the objects, compared with the traditional 1D, shows that this approach can be a good solution for the classification of data with a three-way structure.

Two 2D dissimilarity measures were analyzed for three-way seismic volcanic data to evidence the importance of the selection of a suitable dissimilarity measure for the problem at hand. We developed a new 2D dissimilarity measure that allows taking into account the shape and continuity information in the directions of the three-way array. This measure demonstrated to work well in cases like the three-way seismic volcanic data generated by the spectrograms. In this data, the shape and continuity variation is not represented itself. Consequently, this type of measure is needed to make use of that information. Nevertheless, in cases where there is not discriminative information in both directions, we cannot ensure that this measure is effective. The combination of the matrices from both directions could be influenced if one of them is not good. Further investigations should be done on this issue. In cases like the three-way data obtained by scalograms, more simple dissimilarity measures can be used e.g. AMD. The discriminative information is already embedded in the representation by wavelets.

Although this paper was more focused on the solution for three-way data, it can be extended to multi-way. Further studies will be done on this aspect.

Acknowledgment

We acknowledge financial support from the FET programme within the EU FP7, under the project ”Similarity-based Pattern Analysis and Recognition - SIMBAD” (contract 213250). We would also like to thank to the project Cálculo científico para catacterización e identificación en problemas dinámicos (code Hermes 10722) granted by Universidad Nacional de Colombia.

References

[1] Porro-Muñoz, D., Talavera, I., Duin, R.P.W.: Multi-way data analysis. Technical report, CENATAV (2009)

[2] Kroonenberg, P.M.: Applied Multiway Data Analysis. John Wiley & Sons, Chichester (2008)

[3] Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation For Pattern Recognition. In: Foundations and Applications, World Scientific, Singapore (2005)

[4] Orozco-Alzate, M., García, M.E., Duin, R.P.W., Castellanos, C.G.: Dissimilarity-based classification of seismic signals at Nevado del Ruiz Volcano. Earth Sci. Res. J. 10(2), 57–65 (2006)

[5] Paclik, P., Duin, R.P.W.: Dissimilarity-based classification of spectra: computational issues. Real Time Imaging 9(4), 237–244 (2003)

[6] Benbrahim, M., Daoudi, A., Benjelloun, K., Ibenbrahim, A.: Discrimination of seismic signals using artificial neural networks. In: Ardil, C. (ed.) WEC (2), Enformatika, Çanakkale, Turkey, vol. (2), pp. 4–7 (2005)

[7] Lesage, P., Glangeau, F., Mars, J.: Applications of autoregressive models and time-frequency analysis to the study of volcanic tremor and long-period events. Journal of Volcanology and Geothermal Research 114, 391–417 (2002)

[8] Curilem, G., Vergara, J., Fuentealba, G., Acuña, G., Chacón, M.: Classification of seismic signals at Villarrica Volcano (Chile) using neural networks and genetic algorithms. Journal of Volcanology and Geothermal Research 180(1), 1–8 (2009)

[9] Zobin, V.M.: Introduction to Volcanic Seismology. Developments in Volcanology, vol. 6. Elsevier, Philadelphia (2003)

[10] Zuo, W., Zhang, D., Wang, K.: An assembled matrix distance metric for 2DPCA-based image recognition. Pattern Recognition Letters 27, 210–216 (2006)

[11] Lesage, P.: Interactive Matlab software for the analysis of seismic volcanic signals. Computers & Geosciences 114, 391–417 (2009)

[12] Orozco-Alzate, M., Skurichina, M., Duin, R.P.W.: Spectral characterization of volcanic earthquakes at Nevado del Ruiz Volcano using spectral band selection/extraction techniques. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 708–715. Springer, Heidelberg (2008)

Regularising the Ricci Flow Embedding

Weiping Xu, Edwin R. Hancock, and Richard C. Wilson

Dept. of Computer Science
University of York, UK
{elizaxu,erh,wilson}@cs.york.ac.uk

Abstract. This paper concerns the analysis of patterns that are specified in terms of non-Euclidean dissimilarity or proximity rather than ordinal values. In prior work we have reported a means of correcting or rectifying the similarities so that the non-Euclidean artifacts are minimized. This is achieved by representing the data using a graph, and evolving the manifold embedding of the graph using Ricci flow. Although the method provides encouraging results, it can prove to be unstable. In this paper we explore how this problem can be overcome using a graph regularisation technique. Specifically, by regularising the curvature of the manifold on which the graph is embedded, then we can improve both the stability and performance of the method. We demonstrate the utility of our method on the standard "Chicken pieces" dataset and show that we can transform the non-Euclidean distances into Euclidean space.

1 Introduction

Dissimilarity representations [1] provide a powerful and natural way of capturing the relationships between objects that are not characterised by ordinal measurements or feature vectors. The idea is to use a pairwise dissimilarity (or proximity) measure [2,3] to describe the properties of objects in terms of their attribute differences. Examples of such representations are provided by weighted proximity graphs. The advantages of such a representation are that if characterised in terms of a dissimilarity matrix, then pattern matching can be effected without the need for explicit alignment. However, the dissimilarities are quite frequently non-Euclidean and this prevents the use of many geometrically based learning techniques.

One way to overcome these problems is to represent the dissimilarity data using a weighted graph, and to embed the graph on a manifold. This produces a vectorial representation of the data by projecting dissimilarity data into a fixed-dimensional vector space. Examples of this approach include multidimensional scaling (MDS) [4], Isomap [5], locally linear embedding [6] and the Laplacian eigenmap embedding [7]. The common aim is to locate a low-dimensional representation. In order to apply non-Euclidean dissimilarity data with traditional geometric learning techniques, we must attempt to rectify the data so as to minimize the non-Euclidean artifacts. One route is to consider the positive definite subspace of the distances [8]. An alternative route adopted by Pekalska et al. [9] is to add a suitable constant to the squared off-diagonal elements of the dissimilarity matrix. It is equivalent to adding a certain constant to all eigenvalues of the related Gram matrix, and thus compensating for the effect of the negative eigenvalues, while maintaining the same eigenvector structure.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 579–588, 2010.

In prior work [10] we have shown how to correct the dissimilarity data, giving a set of new Euclidean distances. The method uses Ricci flow on a constant curvature Riemannian manifold to evolve the distance measures. This is effected by updating the curvatures on the edges of the graph representing the data. Unfortunately, the method can prove unstable due to local fluctuations in edge curvature. To overcome this problem, in this paper we show how to stabilise the method by regularising the curvatures of the embedded graph. To do this we use the heat kernel to smooth the curvatures on the edges. The result shows both improved numerical stability and lower classification error in the embedded space.

2 Embedding Non-Euclidean Data

In this paper we are concerned with embedded data represented in terms of pairwise dissimilarities or distances, and in particular the case where the data is non-Euclidean. Our overall aim is to rectify a given set of non-Euclidean dissimilarity data so as to make them more Euclidean. One way to gauge the degree to which a pairwise distance matrix contains non-Euclidean artefacts is to analyse the properties of its centralised Gram matrix. For an $N \times N$ symmetric pairwise dissimilarity matrix D with the pairwise distance as elements, the centered Gram matrix $G = -\frac{1}{2}JD^2J$,where D^2 is element-wise squaring of elements in D, $J = I - \frac{1}{N}11^T$ is the centering matrix and 1 is the all-ones vector of length N. The degree to which the distance matrix departs from being Euclidean can be measured by using the relative mass of negative eigenvalues or "negative eigenfraction" $F_{eigS} = \sum_{\lambda_i<0} |\lambda_i| / \sum_{i=1}^{N} |\lambda_i|$ [11]. This measure is zero for Euclidean distances and increases as the distance becomes increasingly non-Euclidean.

The kernel embedding is obtained from the centered Gram matrix using the factorisation $G = YY^T$, where Y is the $N \times N$ matrix with the embedded co-ordinates of the data as columns. To determine whether the Gram matrix is positive semi definite [11], we perform the eigendecomposition $G = \Phi\Lambda\Phi^T$ on the Gram matrix, where $\Lambda = diag(\lambda_1, ..., \lambda_N)$ is the diagonal matrix with the ordered eigenvalues as elements and $\Phi = (\phi_1|...|\phi_N)$ is the eigenvector matrix with the ordered eigenvectors ϕ_1, ..., ϕ_N as columns. In terms of the eigenvalues and eigenvectors, the matrix of embedded co-ordinates is given by $Y = \Phi\sqrt{\Lambda}$ where the eigenvalues Λ are positive. In Isomap embedding, the dimension and the number of nearest neighbors are estimated to be the optimal values by looking at the residue variances[5].

3 Ricci Flow

Our aim is to develop a method that can be used to rectify the non-Euclidean artefacts in such a dissimilarity matrix. The approach is as follows. Firstly, we consider the objects of interest to be represented by points on a manifold, and the given dissimilarities to be the geodesic distances on the manifold between these points (geodesic distances). For an arbitrary set of non-Euclidean similarities the manifold will be curved. By contrast, a Euclidean space will be flat and the geodesic and Euclidean distances will be identical. Our task is then to remove the curvature from the manifold to create a corrected set of Euclidean distances. We achieve this by evolving the manifold using Ricci flow.

The Ricci flow [12] evolves a manifold so that the rate of change of the metric tensor is controlled by the Ricci curvature. Essentially, this is an analogue of a diffusion process for a manifold. The geometric evolution equation is:

$$\frac{dg_{ij}}{dt} = -2R_{ij} \tag{1}$$

where g_{ij} is the metric tensor of the manifold and R_{ij} is the Ricci curvature.

We model the embedding manifold as consisting of a set of local patches with individual constant Ricci curvatures. These patches can be either elliptic (of positive sectional curvature) or hyperbolic (of negative sectional curvature). It is straightforward to re-express the Ricci flow in terms of the sectional curvature K:

$$\frac{dK}{dt} = \begin{cases} -2K^2 & \text{elliptic hypersphere,} \\ 2K^2 & \text{hyperbolic space.} \end{cases} \tag{2}$$

Under this evolution, the curvature moves towards zero for both types of patch, flattening the manifold. The solution of the differential equation is straightforward. Commencing with the initial conditions $K = K_0$ at time $t = 0$, then at time t we have

$$K = \frac{K_0}{1 \pm 2K_0 t} \tag{3}$$

with the positive sign for the elliptic space.

4 Curvature Computation

Our aim is to evolve a non-Euclidean dissimilarity measure into a Euclidean one using the Ricci flow described in the previous section. We commence by representing the dissimilarity data using a weighted graph $G = (V, E, D)$, where the node set V represents the set of objects and the edges E are weighted with the pairwise dissimilarities. We embed the graph onto a manifold so that the geodesic distance $d_g(u, v), (u, v) \in E$ between the positions of the nodes u and v is equal to the dissimilarity on the edges. Let $\boldsymbol{y}_u$ be the embedded co-ordinates of the node $u \in V$ and $Y = (\boldsymbol{y}_1|...|\boldsymbol{y}_{|V|})$ be the matrix with the embedded co-ordinates as columns. Under this embedding the edges acquire a curvature determined by the difference between geodesic distance (dissimilarity) $d_G(u, v)$ and Euclidean distance $d_E(u, v) = \sqrt{(\boldsymbol{y}_u - \boldsymbol{y}_v)^T(\boldsymbol{y}_u - \boldsymbol{y}_v)}$. The Ricci flow, modifies the Gaussian curvatures on the edges, so as to flatten the manifold. Adopted from [13] we use a Euclidean embedding of the points and use the difference between the geodesic distance d_G on the manifold (from the similarity or dissimilarity matrix) and the Euclidean distance in the embedded space d_E to compute the curvature. We compare experimental results for embeddings obtained with both Isomap [5] and the kernel embedding in Section 7. Lindman and Caelli [14] give the relationship between the two distances on elliptic, hyperbolic and Euclidean constant curvature manifolds as

$$d_E = \begin{cases} \frac{2}{K^{\frac{1}{2}}} \sin(\frac{K^{\frac{1}{2}}}{2} d_G) & \text{Elliptic,} \\ \frac{2}{|K|^{\frac{1}{2}}} \sinh(\frac{|K|^{\frac{1}{2}}}{2} d_G) & \text{Hyperbolic,} \\ d_G & \text{Euclidean.} \end{cases}$$

However, the adopted curvature approximations used only hold for small curvatures. In the data under study here, we find that the curvatures are too large for these approximations to hold. We therefore use it as the initialization and estimate curvature from Equation 4 using Newton's method. Taking the curvature in an elliptic space as an example, the Newton iteration is

$$K_{n+1}^{\frac{1}{2}} = K_n^{\frac{1}{2}} - \frac{K_n^{\frac{1}{2}} d_E - 2\sin\frac{K_n^{\frac{1}{2}}}{2} d_G}{d_E - d_G \cos\frac{K_n^{\frac{1}{2}}}{2} d_G} \tag{4}$$

Finally, we can compute new geodesic distances for the points based on the updated curvature. We keep the Euclidean distance between the points fixed, while updating the curvature.The updated geodesic distance under the new Gaussian curvature can be represented in terms of the old geodesic distance at the previous iteration. The update equation for the geodesic distance is

$$d_{G_{n+1}} = \begin{cases} \frac{2}{K_{n+1}^{\frac{1}{2}}} \operatorname{arcsin}\left(\frac{K_{n+1}^{\frac{1}{2}}}{K_n^{\frac{1}{2}}} \sin(\frac{K_n^{\frac{1}{2}}}{2} d_{G_n}) \right) & \text{elliptic hypersphere} \\ \frac{2}{|K_{n+1}|^{\frac{1}{2}}} \operatorname{arcsinh}\left(\frac{|K_{n+1}|^{\frac{1}{2}}}{|K_n|^{\frac{1}{2}}} \sinh(\frac{|K_n|^{\frac{1}{2}}}{2} d_{G_n}) \right) & \text{hyperbolic space} \end{cases} \tag{5}$$

This equation can be applied to each element of the dissimilarity matrix in turn.

5 The Algorithm

Given a set $X = \{x_1, \cdots, x_N\}$ of N objects and a dissimilarity measure d, a dissimilarity representation is an $N \times N$ matrix D_G. The following algorithm can be used to rectify the distance matrix from being non-Euclidean to Euclidean.

Begin with a pairwise distance matrix $D_G^{(0)}$,

1. Embed the objects in a Euclidean space using either Isomap or the kernel embedding. In the embedded space compute the Euclidean distances d_E.
2. From the geodesic distance d_G and Euclidean distance d_E, compute the constant curvature space with curvature K for a pair of objects using Equation 4.
3. Update the Gaussian curvature with a small time step using Equation 3.
4. Obtain the new geodesic distance $d_{G_{n+1}}$ from the previously available geodesic distance matrix together with the curvatures under a fixed Euclidean distance using Equation 5.
5. Obtain the updated distance matrix $D_G^{(1)}$ containing rectified geodesic distances between objects, and repeat from step 1 until D_G is Euclidean, that is its centered Gram matrix has no negative eigenvalues.

6 Regularizing Curvature

As posed above, the Ricci flow embedding updates the Gaussian curvature separately for each individual edge. This is because we use piecewise constant curvature manifolds for each edge. This places no constraint on the smoothness of the manifold, and this can lead to numerical instability in the embedding. Graph regularization provides a way to smooth data samples over a graph and overcome the numerical stability problems. One such regularization process is a graph diffusion. A diffusion process is analogous to the flow of heat, which flows from high to low concentrations, and over time creates a smooth distribution of heat. In a similar way, a diffusion of a function on the graph will create a smoother function. The diffusion is defined in terms of a random walk on the edges of the graph[15], and is represented by the diffusion (or heat) kernel:

$$H = \exp(-Lt) \tag{6}$$

The evolution of a function under this kernel is simply

$$f(t) = H(t)f(0) \tag{7}$$

The evolution is 'mass-preserving' in the sense that the sum of the values of the function over vertices is preserved.

We can use this process for smoothing curvatures before the application of the Ricci flow, to remove extreme values. However, our curvatures are defined pairs of objects and we therefore need to construct a graph which has vertices corresponding to object-pairs and edges describing a neighbourhood structure of these pairs. We construct this graph as follows. Firstly, we build the nearest-neighbours graph of the objects $G = \{V, E\}$. Each vertex represents an object u and an undirected edge E_{uv} exists if u is in the n nearest neighbours of v or v is in the n nearest neighbours of u. We then construct the dual of this graph $G_D = \{V_D, E_D\}$; each edge of the original graph becomes a vertex V_{uv} and an edge exist between two vertices if they share a common vertex from the original graph. In the dual graph, each vertex represents a pair of objects and the edges reflect the neighbourhood structure of the pairs. We can then define the curvature between object pairs as a function over the vertices of this graph and apply the diffusion kernel.

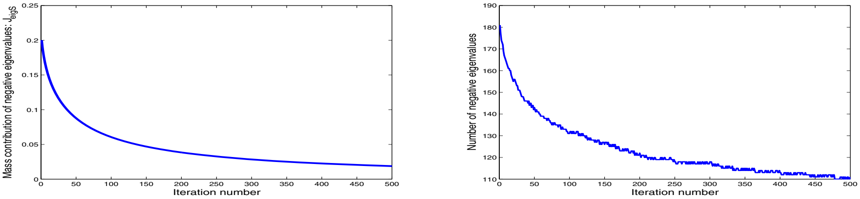
We therefore add an additional step in which we smooth the Gaussian curvatures over the dual of the nearest neighbor graph prior to performing the Ricci flow updating of the curvatures. All of the remaining steps of the algorithm remain as above. The following steps shows how to smooth Gaussian curvatures over the nearest neighbour edges.

Commence with initial Gaussian curvatures K from step 2 above,

1. Construct the n nearest neighbour graph over the available dissimilarity data. Node u and v are connected by an edge if u is among n nearest dissimilarity neighbors of v or v is among n nearest dissimilarity neighbors of u.
2. Construct the dual graph of the nearest neighbour graph. Each edge in the nearest neighbour graph is a vertex of the dual graph. If two edges in the nearest neighbour graph share a one common vertex, then the corresponding two vertices in the dual graph are connected by an edge.
3. Obtain the updated and regularised curvature K. Suppose that $\hat{L}$ is the normalized Laplacian of the dual nearest neighbour graph, then the heat-kernel of the dual graph is $\exp[-\hat{L}t]$. If V_D is the node-set of the dual graph, then we construct a

vector $\boldsymbol{K}$ of Gaussian curvatures $\boldsymbol{K} = (K_1, ..., K_{|V_D|})^T$. The vector of regularised Gaussian curvatures after heat kernel smoothing is $\boldsymbol{K}_{reg} = \exp[-\hat{L}t]\boldsymbol{K}$.

In summary, the above approach commences from a nearest neighbor graph over the dissimilarity matrix, and then constructs the dual graph where a node corresponds to an edge in the original graph. The heat kernel on the dual graph smooths the curvatures on the original nearest neighbour graph.



(a) J_{eigS}

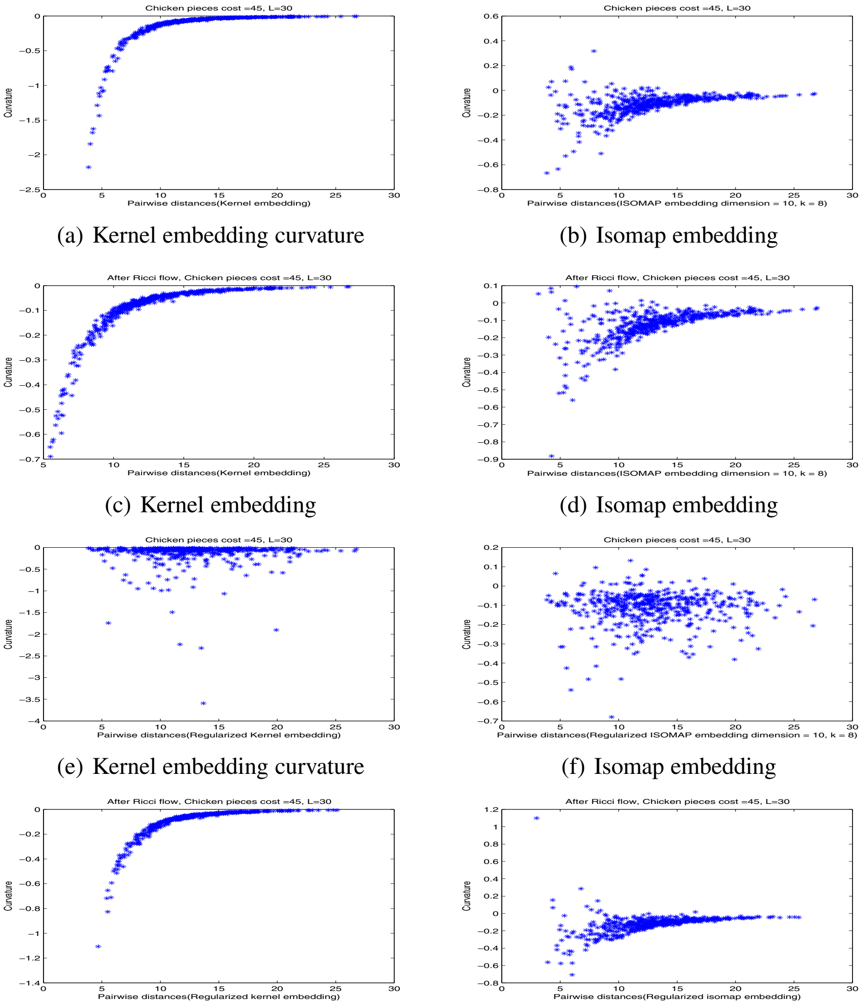
(b) number of negative eigenvalues

Fig. 1. (a) is the negative eigenfraction during iteration. (b) is the number of negative eigenvalues during iteration.

7 Experiments

We use the well known “Chicken pieces” dataset [8] for experimentation. The data-set concerns classifying binary images of a different types of chicken joint into shape-classes. It contains 446 binary images falling into five shape classes, namely a) breast (96 examples), b) back (76 examples), c) thigh and back (61 examples), d) wing (117 examples) and e) drumstick (96 examples). The data exists in the form of a set of non-Euclidean shape dissimilarity matrices, generated using different settings for the parameters in which, L is the length of straight line segments of chicken contours and C is the insertion and deletion costs for computing edit distances between boundary segments. Our experimental results are for the dissimilarity data with C = 45 and L = 5, 10, 15, 20, 25 and 30.

The negative eigenfraction for the Chicken Pieces data with $L = 5.0$, $C = 45$ is shown in Figure 1 as the manifold evolves with iteration number. As the curvatures are updated both the negative eigenfraction and the negative eigenvalues decrease, indicating that the dissimilarity measure becomes increasingly Euclidean. Figure 2 shows the curvatures as a function of distances obtained using the kernel embedding and Isomap embedding. It demonstrates how the Ricci flow process affects distances commencing from the two embedding methods with and without regularization. It indicates that the embedding method affects the magnitude of curvatures. The figure also shows that the Kernel embedding preserves the global distances. Here, the larger the distances, the smaller the curvatures. On the other hand, the Isomap embedding preserves some of the local distances. This maybe the due to the fact that the chicken pieces data does not reside on simple manifold such as Swiss roll. The embedding method determines



(a) Kernel embedding curvature

(b) Isomap embedding

(c) Kernel embedding

(d) Isomap embedding

(e) Kernel embedding curvature

(f) Isomap embedding

(g) Kernel embedding

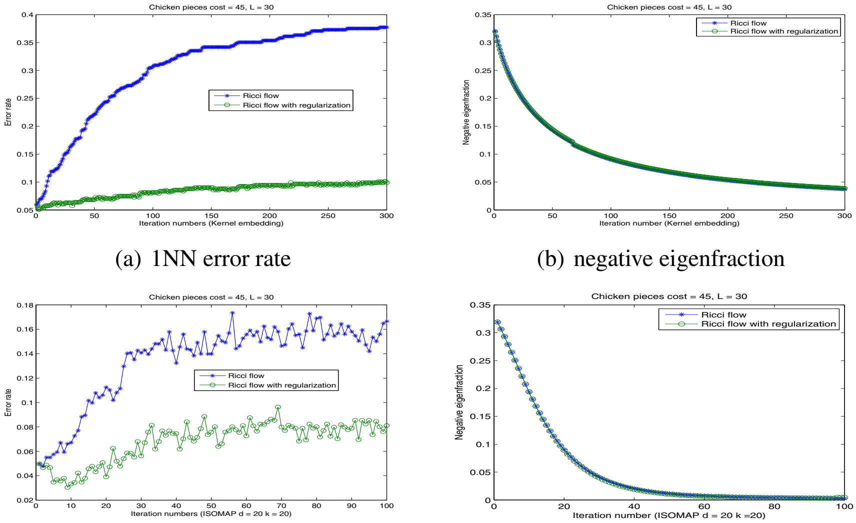
(h) Isomap embedding

Fig. 2. (a) and (b) are initial edge curvatures for the kernel and Isomap embeddings. (c) and (d) are edge curvatures after Ricci Flow for the kernel and Isomap embeddings.(e) and (f) are initial regularised edge curvatures for the kernel and Isomap embeddings. (g) and (h) are edge curvatures after Ricci Flow for the kernel and Isomap embeddings.

the magnitude of curvatures. From our Ricci flow curvature updating process, the larger the magnitude of the original curvatures, the larger the curvature reduction in the update process. As a result in the case of the kernel embedding, those locations associated with large curvature expand more rapidly than those associated with small curvatures. In other words, the initial smaller distances expand more rapidly than larger distances.

This effect can be observed from Figure 2(a) and Figure 2(c). As a result. it disrupts the local pattern of distances without influencing the larger ones.

During the regularization step, the curvatures are smoothed over nearest neighbour edges. The result is to reduce local curvature fluctuations, and this may reduce some locally large curvature values. Figure 2(e) shows that when regularisation is used, the curvatures are smoothed over local distance scales compared to the initial curvatures in Figure 2(a). Hence the local distance structure is preserved under the embedding, and this is demonstrated in Figure 2(g). As a result the regularization step preserves local distances and stabilizes the local structure.



(a) 1NN error rate

(b) negative eigenfraction

(c) 1NN error rate

(d) negative eigenfraction

Fig. 3. (a) is the 1NN error rate with and without the regularization step using the kernel embedding during iteration. (b)is the negative eigenfraction with and without the regularization step using the kernel embedding during iteration. (c) is the 1NN error rate with and without the regularization step using the Isomap embedding during iteration. (d)is the negative eigenfraction with and without the regularization step using the Isomap embedding during iteration.

Next, we turn our attention to the effect of regularisation and the choice of embedding on the results of classification. The classification results were obtained with the 1-NN classifier and 10-fold cross validation. In Figure 3 we compare the 1-NN error rates and the negative eigenfaction obtained with regularised and unregularised versions of Ricci flow on the two embedding schemes. The first point to note is that for both the kernel embedding and Isomap, we obtain better classification results when heat kernel regularisation is used. However, in each case the application of the Ricci flow scheme causes the classification error to increase with iteration number. However, in the case of the regularised kernel embedding, the effect is smallest. Finally, the choice of embedding scheme strongly affects the rate of decrease of the negative eigenfraction, with

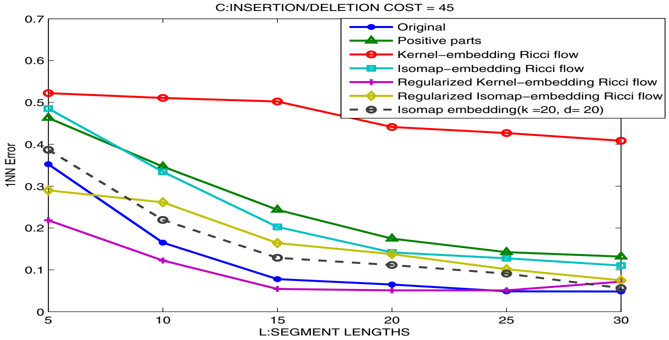


Fig. 4. Error rate from 1NN

Isomap giving a faster rate of decrease with iteration number than the kernel embedding. However, for both embedding schemes the use of regularisation has little effect on the rate of decrease.

Finally, we have compared our results with the known manifold embedding technique Isomap and those obtained using some alternative non-Euclidean distance rectification procedures. The methods explored were a) using the original distances, b) projecting onto the positive subspace and taking the distance here, unregularised Ricci flow on c) the kernel embedding and d) the Isomap embedding, regularised Ricci flow on e) the kernel embedding and f) the Isomap embedding. Figure 4 shows the 1-NN error rate as function of the shape parameter L (the segment length). The best results are obtained with Ricci flow on the regularised kernel embedding. All of the remaining methods give poorer results than applying the classifier to the original distance data.

8 Conclusion

In this paper we have explored how to evolve a non-Euclidean dissimilarity measure into a Euclidean one using Ricci flow. We commence by representing the dissimilarity data using a weighted graph, where the nodes represent objects and the edge weights dissimilarities between objects. We embed the graph onto a manifold so that the geodesic distance between nodes is equal to the dissimilarity on the edges. Under the embedding the edges acquire a curvature determined by the difference between geodesic distance (dissimilarity) and Euclidean distance. The Ricci flow, modifies the Gaussian curvatures on the edges, so as to flatten the manifold. We explore in depth the effect of stabilising this process by using heat-kernel regularisation to smooth the Gaussian curvatures prior to evolving the manifold.

We apply our method to the Chicken Pieces data. When applied without regularisation, although the distance measures can be transformed into a Euclidean space there is some loss of discriminating power and the classifier performance degrades. The loss of information is attributable to the effect of the Ricci evolution process which acts independently on each edge and ignores the local structure of the manifold. When heat kernel regularization is used the ranking of distance measures is preserved, and better performance is achieved. Although the method degraded the error obtained with a 1NN classifier, it does deliver data in a form where geometric classification methods can be applied to the data. The Ricci flow evolution minimise the curvatures, when the curvatures reach zero, then the geodesic and the Euclidean distances are equal and the negative eigenfraction is zero.

As the embedding methods affects the magnitude of curvatures a lot, one way to develop our work is to reduce the reliance on the embedding methods by using spherical embedding and tangent space projection. Another direction is to develop incremental learning, in which new points can be mapped on the manifold, as our current method is performed in a batch mode, i.e., all training points are processed simultaneously.

Acknowledgements

This work was supported by the EU FET project SIMBAD (213250). Edwin Hancock was supported by a Royal Society Wolfson Research Merit Award.

References

1. Pekalska, E., Duin, R.P.W.: Beyond traditional kernels: classification in two dissimilarity-based representation spaces. IEEE Transactions on Systems Man and Cybernetics-Part C 38(6) (November 2008)
2. Sanfeliu, A., Fu, K.S.: A distance measure between attributed relational graphs for pattern recognition. IEEE transactions on systems, man, and cybernetics 13(3), 353–362 (1983)
3. Bunke, H.: A graph distance metric based on the maximal common subgraph. Pattern Recognition Letters 19(3-4), 255–259 (1998)
4. Borg, I., Groenen, P.: Modern multidimensional scaling: Theory and applications. Springer, Heidelberg (2005)
5. Tenenbaum, J., Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
6. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding (2000)
7. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. Advances in neural information processing systems 1, 585–592 (2002)
8. Duin, R.P.W., Pekalska, E., Harol, A., Lee, W.J., Bunke, H.: On euclidean corrections for non-euclidean dissimilarities. In: SSPR/SPR, pp. 551–561 (2008)
9. Pekalska, E., Duin, R., Gunter, S., Bunke, H.: On not making dissimilarities euclidean. Lecture notes in computer science, pp. 1145–1154 (2004)
10. Xu, W., Hancock, E.R.W.: Rectifying non-euclidean similarity data using ricci flow embedding. In: To appear ICPR 2010 (August 2010)
11. Pekalska, E., Harol, A., Duin, R., Spillmann, B., Bunke, H.: Non-euclidean or non-metric measures can be informative, pp. 871–880 (2006)
12. Chow, B., Luo, F.: Combinatorial Ricci flows on surfaces. J. Differential Geom. 63(1), 97–129 (2003)
13. ElGhawalby, H., Hancock, E.R.: Measuring graph similarity using spectral geometry. In: ICIAR, pp. 517–526 (2008)
14. Lindman, H., Caelli, T.: Constant curvature riemannian scaling. Journal of Mathematical Psychology 17, 89–109 (1978)
15. Kondor, R., Lafferty, J.: Diffusion kernels on graphs and other discrete structures. In: Proceedings of the ICML, pp. 315–322 (2002)

Spherical Embedding and Classification*

Richard C. Wilson and Edwin R. Hancock

Department of Computer Science
University of York
Heslington, York, UK
wilson@cs.york.ac.uk

Abstract. Most problems in pattern recognition can be posed in terms of using the dissimilarities between the set of objects of interest. A vector-space representation of the objects can be obtained by embedding them as points in Euclidean space. However many dissimilarities are non-Euclidean and cannot be represented accurately in Euclidean space. This can lead to a loss of information and poor performance. In this paper, we approach this problem by embedding the points in a non-Euclidean curved space, the hypersphere. This is a metric but non-Euclidean space which allows us to define a geometry and therefore construct geometric classifiers. We develop a optimisation-based procedure for embedding objects on hyperspherical manifolds from a given set of dissimilarities. We use the Lie group representation of the hypersphere and its associated Lie algebra to define the exponential map between the manifold and its local tangent space. We can then solve the optimisation problem locally in Euclidean space. This process is efficient enough to allow us to embed large datasets. We also define the nearest mean classifier on the manifold and give results for the embedding accuracy, the nearest mean classifier and the nearest-neighbor classifier on a variety of indefinite datasets.

1 Introduction

Many pattern recognition problems can be posed in terms of measuring the dissimilarities between a set of objects. This is a very general approach, as it is a superset of the classic feature-based approach. Nearly all approaches to recognition involve measuring a dissimilarity or distance and classifying on this basis. One approach to this problem is to embed objects into a vector-space using techniques such as multidimensional scaling or IsoMap[1]. Once embedded in such a space then the objects can be characterised by their embedding co-ordinate vectors, and analysed in a conventional manner using Euclidean distance.

There are however some limits to this paradigm; Euclidean distances are always *definite* and are intrinsically unable to represent dissimilarities which are indefinite. We discuss the issue of indefinite dissimilarities in more detail in the

* We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). Edwin Hancock was also supported by a Royal Society Wolfson Research Merit Award.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 589–599, 2010.

next section. In practice, many dissimilarity measures are indefinite; examples include shape-similarities, and distance measures used in gesture interpretation and graph comparison, but there are many more. Any method of comparison which relies on local alignment or variable local control parameters has the potential to produce indefinite (non-Euclidean) dissimilarities.

One alternative is to 'correct' the data to remove the indefinite part. However, previous work[2] has shown that there is potentially useful information in the non-Euclidean part of the dissimilarities, and removing this can result in worse performance. Another alternative is to embed the data in a pseudo-Euclidean space, i.e. one where certain dimensions are characterised by negative eigenvalues and the squared-distance between objects has positive and negative components which sum together to give the total distance. A pseudo-Euclidean space is however non-metric, which makes it difficult to correctly compute the geometric quantities required by many classifiers. This is because locality is not preserved in this space; two points which are far apart can both be close to a third point.

A third alternative, which we explore here, is to use a non-Euclidean, but metric, embedding space. A Riemannian manifold is curved, and the geodesic distances are metric. However they can also be indefinite and so can represent indefinite dissimilarities. In this paper, we explore the embedding of objects onto the hypersphere with its associated spherical geometry. Non-Euclidean embeddings have been reported elsewhere in the literature. For example, Lindman and Caelli have studied both spherical and hyperbolic embeddings in the context of interpreting psychological data[3]. Cox and Cox[4] describe multidimensional scaling constrained to a spherical space and optimise the stress to find a good embedding. Shavitt and Tankel have used the hyperbolic embedding as a model of internet connectivity[5]. Hubert et al have investigated the use of unidimensional embeddings on circles[6]. Robles-Kelly and Hancock[7] preprocess the available similarity data so that it conforms either to elliptic or hyperbolic geometry. In practice the former corresponds to a scaling of the distance using a sine function, and the latter scaling the data using a hyperbolic sine function.

In this paper, we propose a optimisation-based procedure for embedding objects on hyperspherical manifolds. The purpose of this embedding is to faithfully represent the dissimilarities between objects in a metric space. A metric space is important because is allows us to compute statistics and define geometric constructs such as boundaries, in contrast to a non-metric space where non-locality is a problem. We also define the nearest mean classifier on the manifold and give results for the embedding accuracy, the nearest mean classifier and the nearest-neighbor classifier on a variety of indefinite datasets. The optimisation approach we use employs the Lie group representation of the hypersphere and its associated Lie algebra to define the exponential map between the manifold and its local tangent space. We can then solve the optimisation problem locally in Euclidean space. This process is efficient enough to allow us to embed datasets of several thousand objects.

2 Indefinite Spaces

We begin with the assumption that we have a set of objects of interest and have measured a set of dissimilarities or distances between all pairs of objects in our problem. This is denoted by the matrix $\mathbf{D}$, where D_{ij} is the distance between objects i and j. We can define an equivalent set of similarities by using the matrix of squared distances $\mathbf{D}'$, where $D'_{ij} = D^2_{ij}$. This is achieved by identifying the similarities as $-\frac{1}{2}\mathbf{D}'$ and centering the resulting matrix:

$$\mathbf{S} = -\frac{1}{2}(\mathbf{I} - \frac{1}{n}\mathbf{J})\mathbf{D}'(\mathbf{I} - \frac{1}{n}\mathbf{J}) \tag{1}$$

Here $\mathbf{J}$ is the matrix of all-ones, and n is the number of objects. In Euclidean space, this procedure gives exactly the inner-product or kernel matrix for the points.

If $\mathbf{S}$ is positive semi-definite, then the original distances are Euclidean and we can use the kernel embedding to locate positions $\mathbf{x}_i$ for the points in Euclidean space as follows

$$\mathbf{S} = \mathbf{U}_S\mathbf{\Lambda}_S\mathbf{U}_S^T = \mathbf{X}\mathbf{X}^T \tag{2}$$

$$\mathbf{X} = \mathbf{U}_S\mathbf{\Lambda}_S^{\frac{1}{2}} \tag{3}$$

where $\mathbf{U}_S$ and $\mathbf{\Lambda}_S$ are the eigenvector and eigenvalue matrices of $\mathbf{S}$, respectively. The position-vector $\mathbf{x}_i$ of the i^{th} point corresponds to the i^{th} row of $\mathbf{X}$.

If $\mathbf{S}$ is indefinite, which is often the case, then the objects cannot exist in Euclidean space with the given distances. This does not necessarily mean the the distances are non-metric; metricity is a separate issue. One measure of the deviation from definiteness which has proved useful is the negative eigenfraction (NEF) which measures the fractional weight of eigenvalues which are negative[8]:

$$\text{NEF} = \frac{\sum_{\lambda_i<0}|\lambda_i|}{\sum_i|\lambda_i|} \tag{4}$$

If NEF=0, then the data is definite and can be represented by points in Euclidean space. We can measure the *non-metricity* of the data by counting the number of violations of metric properties. It is very rare to have an initial distance measure which gives negative distance, so we will assume than the distances are all positive. The two measures of interest are then the fraction of triples which violate the triangle inequality (TV) and the degree of asymmetry of the distances (γ)[2]:

$$\gamma = \sum_{i\neq j}\frac{|\tilde{d}(i,j) - \tilde{d}(j,i)|}{|\tilde{d}(i,j) + \tilde{d}(j,i)|} \tag{5}$$

where $\tilde{d}(.,.)$ is the dissimilarity scaled so that the average dissimilarity is one.

If the data is metric (or, in practice, close to metric) but indefinite then we must use a curved space to embed the points.

3 Spherical Space

A spherical space is an example of a Riemannian manifold. On the manifold, distances are measured by geodesics (the shortest curve between points), and geodesic distances are metric. Spherical space is curved however, and so the distances are fundamentally non-Euclidean and in general the similarity matrix of points in spherical space will be indefinite. This makes it a potential choice for representing non-Euclidean datasets.

A manifold embedding is important because it allows the use of geometric and statistical tools on the embedded points. On a Riemannian manifold, distances are defined between any pair of points in the manifold in a consistent way (not just between the sample data-points). Geodesic distance is defined as the length of the shortest curve which joins two points (the curve is known as a geodesic), and is a metric. Geodesics are the equivalent of straight lines in Euclidean space, and allow us to construct a geometry in curved space. We can also compute statistics such as the mean in a way consistent with the normal Euclidean definition. This means that all the standard classifiers can be applied (at least in theory) to the data, but the exact formulation will differ from vector-space classifiers.

The spherical manifold in 2D is isomorphic to the 2D surface of a sphere embedding in 3D space, which has a well-known parametric form. Here r is the radius of the sphere, u is the azimuth angle and v is the zenith angle.

$$\mathbf{x} = (r \sin u \sin v, r \cos u \sin v, r \cos v)^T \tag{6}$$

This geometry generalises to an $n-1$ dimensional hypersphere embedded in an n-dimensional Euclidean space. The surface can be defined implicitly using the constraint

$$\sum_i x_i^2 = r^2 \tag{7}$$

where r is the radius of the hypersphere. This surface is curved and has a constant sectional curvature of $K = 1/r^2$ everywhere.

The geodesic distance between two points in curved space is the length of the shortest curve lying in the space and joining the two points. On the hypersphere, the geodesic is a great circle. The distance is the length of the arc of the great circle which joins the two points. If the angle subtended by two points at the centre of the hypersphere is θ_{ij}, then the distance between them is

$$d_{ij} = r\theta_{ij} \tag{8}$$

With the coordinate origin at the centre of the hypersphere, we can represent a point by a position vector $\mathbf{x}_i$ of length r. Since the inner product is $\langle \mathbf{x}_i, \mathbf{x}_j \rangle = r^2 \cos \theta_{ij}$ we can also write

$$d_{ij} = r \cos^{-1} \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{r^2} \tag{9}$$

4 The Exponential Map

Our procedure for embedding points on a sphere requires one important tool of Riemannian geometry, which is the exponential map. The exponential map is a map from points on the manifold to points on a tangent space of the manifold. As the tangent space is flat (i.e. Euclidean), we can calculate quantities in a straightforward way. The map has an origin, which defines the point at which we construct the tangent space of the manifold. The formal definition of the Exponential map is the map which connects the Lie algebra on the tangent space to the Lie group which defines the manifold. We will not concern ourselves with the technical details here, but the map has an important property which simplifies geometric computations; the geodesic distance between the origin of the map and a point on the manifold is the same as the Euclidean distance between the images of the two points on the tangent space. Formally, the definition of these properties as follows: Let T_M be the tangent space at some point M on the manifold, P be a point on the manifold and X a point on the tangent space. We have

$$X = \mathrm{Log}_M P \tag{10}$$

$$P = \mathrm{Exp}_M X \tag{11}$$

$$d_g(P, M) = d_e(X, M) \tag{12}$$

The Log and Exp notation defines a log-map from the manifold to the tangent space and an exp-map from the tangent space to the manifold. This is a formal notation and does not imply the normal log and exp functions - although they do co-incide for some types of data, they are not the same for the spherical space. M is the origin of the map and is mapped onto the origin of the tangent space. The distance $d_g(.,.)$ is the geodesic distance on the manifold and $d_e(.,.)$ the Euclidean distance on the tangent space.

For the spherical manifold, the exponential map is as follows. We define a point P on our manifold as a position vector $\mathbf{p}$ with length r (the origin is at the centre of the hypersphere). Similarly, the point M is represented by the vector $\mathbf{m}$, and M is the origin of the map. The maps are then

$$\mathbf{x} = \frac{\theta}{\sin\theta}(\mathbf{p} - \mathbf{m}\cos\theta) \tag{13}$$

$$\mathbf{p} = \mathbf{m}\cos\theta + \frac{\sin\theta}{\theta}\mathbf{x} \tag{14}$$

$$d_g(P, M) = d_e(X, M) = |\mathbf{x}| = r\theta \tag{15}$$

where $\theta = \cos^{-1}\langle \mathbf{p}, \mathbf{m}\rangle / r^2$. The vector $\mathbf{x}$ is the image of P in the tangent space, and the image of M is at the origin of the tangent space.

5 Spherical Embedding

Given a dissimilarity matrix $\mathbf{D}$, we want to find the embedding of a set of points on the surface of a hypersphere of radius r, such that the geodesic distances are as similar as possible to $\mathbf{D}$. Unfortunately, this appears to be a hard problem and therefore we use an approximate optimisation-based approach. We simplify the problem by considering just the distances to a single point at a time. Let the point of interest be $\mathbf{p}_i$; we then want to find a new position for this point on the hypersphere such that the geodesic distance to point j is d^*_{ij} where $*$ denotes that this is the target distance. We formulate the estimation of position as a least-squares problem which minimises

$$E = \sum_{j \neq i} (d^2_{ij} - d^{*2}_{ij})^2 \tag{16}$$

where d_{ij} is the actual distance between the points. This is a similar formulation to Cox and Cox[4] and other approaches to non-Euclidean multidimensional scaling, who seek to minimise the 'stress'. Direct optimisation on the sphere is complicated by the need to restrict points to the manifold. However, as we are considering a single point at a time, we can construct a linear embedding using the log-map and optimise in the Euclidean space. This is a different approach to that of Cox and Cox[4]. If the current point-positions on the hypersphere are $\mathbf{p}_j, \forall j$, we can use the log-map to obtain point-positions for each object in the tangent space of $\mathbf{x}_j \forall j$ as follows:

$$\mathbf{x}_j = \mathrm{Log}_{\mathbf{p}_i} \mathbf{p}_j = \frac{\theta_{ij}}{\sin \theta_{ij}} (\mathbf{p}_j - \mathbf{p}_i \cos \theta_{ij}) \tag{17}$$

with $\mathbf{x}_i = 0$.

We have found standard optimisation schemes to be infeasible on larger datasets, so here we propose a gradient descent scheme with optimal step-size. In this iterative scheme, we update the position of the point $\mathbf{x}_i$ in the tangent space to obtain a better fit to the given distances. At iteration k, the point is at position $\mathbf{x}_i^{(k)}$. Initially, the point is at the origin, so $\mathbf{x}_i^{(0)} = 0$. Since the points lie in tangent space, which is Euclidean, we then have $d^2_{ij} = (\mathbf{x}_j - \mathbf{x}_i)^T (\mathbf{x}_j - \mathbf{x}_i)$ and gradient of the error is

$$\nabla E = 4 \sum_{j \neq i} (d^2_{ij} - d^{*2}_{ij})(\mathbf{x}_i - \mathbf{x}_j) \tag{18}$$

and our iterative update procedure is

$$\mathbf{x}_i^{(k+1)} = \mathbf{x}_i^{(k)} + \eta \nabla E \tag{19}$$

Finally, we can determine the optimal step size as follows: let $\Delta_j = d^2_{ij} - d^{*2}_{ij}$ and $\alpha_j = \nabla E^T (\mathbf{x}_i - \mathbf{x}_j)$, then the optimal step size is the smallest root of the cubic

$$n|\nabla E|^2 \eta^3 + 3|\nabla E|^2 (\sum_j \alpha_j) \eta^2 + (2 \sum_j \alpha_j^2 + |\nabla E|^2 \sum_j \Delta_j) \eta + \sum_j \alpha_j \Delta_j \tag{20}$$

This step-size is optimal in the sense that it minimises the error in the direction of the gradient.

After finding a new point position $\mathbf{x}_i$, we apply the exp-map to locate the new point position on the spherical manifold

$$\mathbf{p}_i' = \mathbf{p}_i \cos\theta + \frac{\sin\theta}{\theta}\mathbf{x}_i \tag{21}$$

5.1 Classifiers in the Manifold

As well as embedding distances on the spherical manifold, it is important to be able to perform operations such as classification in the manifold. Some classifiers are trivially implemented on a spherical manifold, for example the nearest-neighbors(NN). Others which utilise geometry must be modified to incorporate the non-Euclidean geometry of curved space. Here we discuss the nearest mean classifier(NMC) in a non-flat manifold.

The *intrinsic mean* of a set of points on the manifold may be computed via the generalised mean[9]

$$\bar{P} = \arg\min_P \sum_i d_g(P, P_i) \tag{22}$$

We can solve for the mean of a set of points in a manifold using the following iterative procedure involving the exponential map[9]:

$$\mathbf{m}^{(k+1)} = \mathrm{Exp}_{\mathbf{m}^{(i)}} \frac{1}{n} \sum_i \mathrm{Log}_{\mathbf{m}^{(i)}} \mathbf{p}_i \tag{23}$$

While the convergence of this process is not guaranteed in a general manifold, it is well behaved on the hypersphere[9]. As a result, we can compute the means of each class $\mathbf{m}_1, \ldots \mathbf{m}_C$ and implement the NMC:

$$c^* = \arg\min_c \left[r \cos^{-1} \frac{\langle \mathbf{x}, \mathbf{m}_c \rangle}{r^2} \right] \tag{24}$$

6 Experimental Results

We have applied our embedding method to a number of indefinite datasets. These are summarised in the table below, along with their degree of indefiniteness, as measured by the negative eigenfraction (Eqn. 4). These datasets are produced by dissimilarity measures applied to a variety of real world problems. The Coil datasets are produced by graph-matching algorithms applied to corner-graphs of some of the objects in the COIL database[10], using graduated assignment[11](CoilYork) and the JoEig approach[12](CoilDelftDiff and CoilDelftSame). The CatCortex data gives the similarity between different cortical regions in terms of connectivity[13]. The DelftGestures dataset consists of the dissimilarities computed from a set of gestures in a sign-language using a dynamic time warping procedure[14]. The FlowCyto series of datasets is based

Table 1. Properties of datasets used

Dataset	Size	NEF	Triangle violations	Asymmetry
CoilYork	288	0.258	1/23639616	0.009
DelftGestures	1500	0.308	14798/3368253000	0
FlowCyto-1	612	0.275	272052/228098520	0
FlowCyto-2	612	0.268	161517/228098520	0
FlowCyto-3	612	0.275	272879/228098520	0
FlowCyto-4	612	0.272	268991/228098520	0
Newsgroups	600	0.202	4643/214921200	0
Chickenpieces-5	446	0.216	0/88120680	0.044
Chickenpieces-10	446	0.257	1/88120680	0.046
Chickenpieces-15	446	0.286	74/88120680	0.051
Chickenpieces-20	446	0.307	695/88120680	0.057
Chickenpieces-25	446	0.320	1375/88120680	0.063
Chickenpieces-30	446	0.331	3188/88120680	0.067
Chickenpieces-35	446	0.339	4834/88120680	0.073
Chickenpieces-40	446	0.345	7549/88120680	0.076
CatCortex	65	0.272	286/262080	0
CoilDelftDiff	288	0.128	1/23639616	0
CoilDelftSame	288	0.027	0/23639616	0
WoodyPlants50	791	0.229	115253/493038210	0
ProDom	2604	0.043	136/17636907624	0
Zongker	2000	0.419	6583656/7988004000	0.051

on the L1-norm dissimilarities between flowcytometer histograms of breast cancer tissues. The data were acquired by M. Nap and N. van Rodijnen of the Atrium Medical Center in Heerlen, The Netherlands, during 2000-2004. Newsgroups is a small subset of the 20Newsgroups data of Roweis. The ProDom dataset is a set of dissimilarities derived from the structural matching of protein domain sequences[15]. WoodyPlants50 is a dataset of shape dissimilarities between plant leaves[16]. The Zongker dissimilarities are based on deformable template matching between 2000 handwritten digits in 10 classes[17]. Finally, the Chickenpiece dataset is another set of shape dissimilarities derived from string-edit distance on the contours of chicken piece silhouettes[2]. This data has a number of controllable parameters which influence the indefinite nature of the dissimilarities. Here we use and edit cost of 45 and a variety of contour lengths (5,10,15,20,25,30,35,40).

We characterise the accuracy of our embeddings in two different ways. Firstly we measure the RMS fractional error of the embedded distances:

$$\text{RMS Error} = \left(\frac{1}{n} \sum_{ij} \frac{D_{ij} - D^*_{ij}}{\bar{D}} \right) \tag{25}$$

where $\bar{D}$ is the average dissimilarity between objects in the original data. Secondly, we measure the 1NN classifier error, both before and after embedding.

Table 2. Embedding results for the datasets (in order of increasing error)

Dataset	Size	1NN (orig)	**Error**	Radius	1NN (emb)	NMC
Newsgroups	600	0.269 ± 0.015	0.022	0.6298	0.279 ± 0.012	0.208 ± 0.015
CoilDelftDiff	288	0.487 ± 0.033	0.030	0.0277	0.479 ± 0.022	0.467 ± 0.034
Chickenpieces-5	446	0.350 ± 0.022	0.030	66.9	0.417 ± 0.022	0.407 ± 0.02
WoodyPlants50	791	0.101 ± 0.008	0.034	0.4362	0.147 ± 0.015	0.197 ± 0.016
Chickenpieces-10	446	0.170 ± 0.016	0.039	33.4	0.249 ± 0.018	0.338 ± 0.022
DelftGestures	1500	0.042 ± 0.0048	0.039	3.9826	0.135 ± 0.009	0.104 ± 0.004
Chickenpieces-15	446	0.079 ± 0.011	0.049	20.73	0.116 ± 0.018	0.249 ± 0.028
Chickenpieces-20	446	0.069 ± 0.012	0.052	17	0.109 ± 0.011	0.202 ± 0.022
Chickenpieces-25	446	0.048 ± 0.01	0.057	13.1	0.086 ± 0.013	0.21 ± 0.025
FlowCyto-2	612	0.366 ± 0.019	0.059	12132	0.378 ± 0.017	0.389 ± 0.028
Chickenpieces-30	446	0.048 ± 0.009	0.062	11.01	0.091 ± 0.013	0.197 ± 0.015
CoilYork	288	0.278 ± 0.025	0.063	177.8	0.307 ± 0.024	0.471 ± 0.029
FlowCyto-3	612	0.413 ± 0.013	0.072	13078	0.421 ± 0.021	0.4 ± 0.015
Chickenpieces-35	446	0.065 ± 0.011	0.073	10.12	0.069 ± 0.007	0.178 ± 0.023
Chickenpieces-40	446	0.087 ± 0.014	0.078	8.14	0.099 ± 0.012	0.2 ± 0.015
FlowCyto-1	612	0.369 ± 0.013	0.078	12794	0.425 ± 0.008	0.385 ± 0.02
CatCortex	65	0.095 ± 0.034	0.084	2.33	0.111 ± 0.04	0.047 ± 0.025
FlowCyto-4	612	0.425 ± 0.023	0.090	11761	0.413 ± 0.018	0.436 ± 0.026
ProDom	2604	0.002 ± 0.001	0.122	471.1	0.038 ± 0.003	0.21 ± 0.011
CoilDelftSame	288	0.636 ± 0.031	0.134	0.0577	0.674 ± 0.040	0.433 ± 0.038
Zongker	2000	0.372 ± 0.016	0.233	0.2887	0.043 ± 0.005	0.109 ± 0.009

This demonstrates whether the embedding preserves the local structure of the classes adequately. In the final column we show the performance of the NMC classifier on the hypersphere.

The results show that we obtain an accuracy spherical embedding for nearly all the data. Of the 21 datasets, only three have more than 10% RMS error on the embedding. This demonstrates the effectiveness of our embedding technique at locating optimal embeddings. For ten of the datasets, we see virtually identical 1NN performance both before and after embedding, and for one a large improvement(Zongker). We do not know the cause of this unexpected behaviour, but it seems to be a feature of this particular dataset. For the other ten sets, we see deterioration in the 1NN classification, indicating that the local structure has been changed somewhat. This is particularly evident in the Chickenpieces data, for which six of the eight examples give worse 1NN scores. It seems that this data series is unsuitable for spherical embedding.

The NMC classifier shows a far wider range of permformance. The Chickenpieces data series, CoilYork, WoodyPlants50 and ProDom show a substantially worse performance with the NMC than with the original 1NN classifier, whereas Newsgroups, CatCortex, CoilDelftSame and Zongker show a substantial improvement.

7 Conclusions

In this paper we used spherical embedding as a solution to the problem of indefinite, non-Euclidean dissimilarities. This embedding preserves some of the non-Euclidean nature of the dissimilarities which may be important in other tasks such as classification. We developed an optimisation-based procedure for embedding objects on hyperspherical manifolds which uses the Lie group representation of the hypersphere and its associated Lie algebra to define the exponential map between the manifold and its local tangent space. The optimisation is then solved locally in Euclidean space. This process is efficient enough to allow us to embed datasets of several thousand objects. We also defined the nearest mean classifier on the manifold.

Experiments on a variety of non-Euclidean datasets show that we can obtain accurate embeddings representing the dissimilarities on the hypersphere. The classification results show that the embedding of some datasets is very useful (for example the Newsgroups data), and for others not effective (the Chickenpieces data).

References

1. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
2. Pekalska, E., Harol, A., Duin, R.P.W., Spillmann, B., Bunke, H.: Non-euclidean or non-metric measures can be informative. In: SSPR/SPR, pp. 871–880 (2006)
3. Lindman, H., Caelli, T.: Constant curvature riemannian scaling. Journal of Mathematical Psychology 17, 89–109 (1978)
4. Cox, T.F., Cox, M.A.A.: Multidimensional scaling on a sphere. Communications in Statistics - Theory and Methods 20(9), 2943–2953 (1991)
5. Shavitt, Y., Tankel, T.: Hyperbolic embedding of internet graph for distance estimation and overlay construction. IEEE/ACM Transactions on Networking 16, 25–36 (2008)
6. Hubert, L., Arabie, P., Meulman, J.: Linear and circular unidimensional scaling for symmetric proximity matrices. British Journal of Mathematical & Statistical Psychology 50, 253–284 (1997)
7. Robles-Kelly, A., Hancock, E.R.: A riemannian approach to graph embedding. Pattern Recognition 40, 1042–1056 (2007)
8. Pekalska, E., Duin, R.P.W.: The dissimilarity representation for pattern recognition. World Scientific, Singapore (2005)
9. Fletcher, P.T., Lu, C., Pizer, S.M., Joshi, S.: Principal geodesic analysis for the study of nonlinear statistics of shape. IEEE Transactions on Medical Imaging 23(8), 995–1005 (2004)
10. Xiao, B., Hancock, E.R.: Geometric characterisation of graphs. In: Roli, F., Vitulano, S. (eds.) ICIAP 2005. LNCS, vol. 3617, pp. 471–478. Springer, Heidelberg (2005)
11. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 18, 377–388 (1996)

12. Lee, W.J., Duin, R.P.W.: An inexact graph comparison approach in joint eigenspace. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 35–44. Springer, Heidelberg (2008)
13. Scannell, J., Blakemore, C., Young, M.: Analysis of connectivity in the cat cerebral cortex. Journal of Neuroscience 15, 1463–1483 (1995)
14. Lichtenauer, J., Hendriks, E.A., Reinders, M.J.T.: Sign language recognition by combining statistical dtw and independent classfication. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 2040–2046 (2008)
15. Roth, V., Laub, J., Buhmann, J., Mueller, K.R.: Going metric: Denoising pairwise data. Advances in Neural Information Processing Systems, 841–856 (2003)
16. Ling, H., Jacobs, D.: Shape classification using the inner-distance. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 286–299 (2007)
17. Jain, A., Zongker, D.: Representation and recognition of handwritten digits using deformable templates. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 1386–1391 (1997)

Language Detection and Tracking in Multilingual Documents Using Weak Estimators[*]

Aleksander Stensby[1,**], B. John Oommen[1,2], and Ole-Christoffer Granmo[1]

[1] Dept. of ICT, University of Agder, Grimstad, Norway
[2] School of Computer Science, Carleton University, Ottawa, Canada[***]

Abstract. This paper deals with the extremely complicated problem of language detection and tracking in real-life electronic (for example, in Word-of-Mouth (WoM)) applications, where various segments of the text are written in different languages. The difficulties in solving the problem are many-fold. First of all, the analyst has no knowledge of when one language stops and when the next starts. Further, the features which one uses for any one language (for example, the n-grams) will not be valid to recognize another. Finally, and most importantly, in most real-life applications, such as in WoM, the fragments of text available before the switching, are so *small* that it renders any meaningful classification using traditional estimation methods almost meaningless. Earlier, the authors of [10] had recommended that for a variety of problems, the use of strong estimators (i.e., estimators that converge with probability 1) is sub-optimal. In this vein, we propose to solve the current problem using novel estimators that are pertinent for non-stationary environments. The classification results which involve as many as 8 languages demonstrates that our proposed methodology is both powerful and efficient.

Keywords: Multilingual language detection, Weak estimators.

1 Introduction

In this paper, we consider the fascinating problem of language detection and tracking in real-life electronic (for example, in Word-of-Mouth (WoM)) applications. Unlike more traditional Pattern Recognition (PR) problems, in this case we encounter the scenario where the various segments of the text are written in different languages, and are both short and "chatty". We know that every PR

[*] The second author was partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada. All the authors gratefully acknowledge the support of *Integrasco*, a Word-of-Mouth company whose headquarters is in Grimstad, Norway, and in particular of its CEO, Mr. Jan Hansen.

[**] Former student. Can be contacted at: C/o Dr. Ole-Christoffer Granmo, Dept. of ICT, University of Agder, Grooseveien 36, 4876 Grimstad, Norway.

[***] *Chancellor's Professor*; *Fellow : IEEE* and *Fellow : IAPR*. The Author also holds an *Adjunct Professorship* with the Dept. of ICT, University of Agder, Norway.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 600–609, 2010.

problem essentially involves two issues, namely the training and the classification of the patterns. In the training phase, the class-conditional distribution of the features is estimated, based on the given training samples. Generally speaking, traditional PR systems assume that the class-conditional distributions are stationary, and thus that they do not change with time. However, in the case of the problem we study, as we shall see, the training data possesses non-stationary class-conditional distributions. All of these issues render the problem being studied both difficult and non-trivial.

The traditional strategy to deal with non-stationary environments has been one of using a *sliding window* [6]. The problem with this is that if the size of the window is too small, the corresponding estimates tend to be poor. If one chooses a too-large window size, the estimates prior to the change of the parameter have too much influence on the new estimates. Also, the observations during the entire window width must be maintained and updated during the process of estimation.

There are numerous problems which have been recently reported, where strong estimators pose a real-life concern. Recently Oommen and his co-authors presented a strategy by which the parameters of a binomial/multinomial distribution can be estimated when the distribution is non-stationary [10]. The method is referred to as the *Stochastic Learning Weak Estimator* (SLWE), and is a novel estimation method based on the principles of stochastic learning. We propose to use the SLWE in our particular PR problem.

1.1 Topic Detection and Tracking and Word of Mouth

The non-stationary phenomenon described above occurs in the PR problems related to Topic Detection and Tracking (TDT) in online discussions, where the content of the discussions represents the opinions of users from all over the world. This kind of information has high value for market-oriented or consumer-focused companies.

The phenomenon of consumers providing information to other consumers is often referred to as Word of Mouth (WoM). It turns out that the nature of these discussions, consisting of multiple opinions, different topics, and a variety of languages, presents us with a problem of designing training and classification strategies when the class-conditional distributions are non-stationary.

The main difference between classification of news articles or journal papers and WoM discussions, is that these discussions generally contain the opinions of *several* different authors. Considering a discussion where several authors write parts of it means that we have a document with continuous content changes.

Treating the whole discussion as one contiguous document, the task at hand is thus to segment the discussion and to classify each segment according to the pre-defined classes, whether it be topics, sentiment or language.

Another important aspect of text classification of such WoM discussions is that the postings often are composed on the fly by the different users, without any form of spell checking. Thus, when performing text classification on such data, one must tolerate the presence of different kinds of textual errors, such as spelling and grammatical errors. Abbreviations and Internet “slang” may also be present. The classification process must work reliably on all input, and must tolerate these kind of errors to some extent. *The complexity of the problem being studied should thus be obvious to the reader*!

1.2 Contributions of This Paper

The present work develops an efficient and accurate methodology for the training and testing of topic detection and tracking in multilingual online discussions. In contrast to the state-of-the-art, we introduce a novel approach to language classification in multilingual documents where the classification is done without any prior segmentation of the sample document, and where we do not require the class-conditional distributions of the “features” to be stationary. The method utilizes the principles of the SLWE proposed by Oommen *et al.* to update the probabilities of the input samples, combined with mixed-order n-grams as the discriminatory features, based on an n-gram language model [4]. In the light of the above, we believe that our work is both novel and of a pioneering sort.

2 Language Classification in Mono/Multilingual Documents

A crucial problem that has received little attention in the literature is that of classifying documents containing several languages, or so-called multilingual documents. The task of language classification has been widely studied, but most of the approaches focus on classifying documents written in a single language, often referred to as monolingual documents.

There are several different approaches to selecting features for language identification. These include, for instance, the presence of particular characters as discriminators [13] or the presence of particular character n-grams [12]. Cavnar and Trenkle approached the task of language classification in monolingual documents in [1], by using n-gram analysis.

Other frequently used approaches to language classification are the dictionary approach or use of words that commonly appear in the language of interest[5]. Such non-linguistically motivated features generally perform well for documents of moderate length, but their performance is significantly decreased when the length of the sample text gets shorter. Other approaches to language classification using linguistic factors that differ among languages are also found in the literature. One such approach is based on the use of *morphological* features presented by Creutz in [3] and [2]. The problem with these approaches is that the construction of a morphological lexicon for a given language requires a large amount of work by trained experts.

With respect to multilingual documents, Ozbek *et al.* presented an approach in [11], where they make use of the Creutz algorithm. Their approach demonstrated good results for the Turkish language, but the results were discouraging for the English language, with a worst case accuracy of 40%. Ludovik and

Zacharski proposed an algorithm for classifying multilingual documents that is based on mixed-order n-grams, Markov chains, maximum likelihood and dynamic programming in [7]. Language classification in multilingual documents using a word-window approach was studied in [8] by Mandl *et al.* Their results demonstrated a high accuracy for detecting the languages, but they pointed out that determining the location of the language shift was the hardest challenge, reporting a cumulative precision of 81% for locating the change point with at most 2 words off the real change point.

Our proposed method is distinct from all of the above. We are interested in classification tasks that involve the non-stationarity found in such multilingual documents, in which moreover, we do not require the scheme to know the boundaries of the different language segments in the document.

3 Weak Estimators: The SLWE

The fundamental estimation strategy that we advocate for the problem being studied is the SLWE alluded to earlier. We shall explain it, in some detail, here.

When dealing with an alphabet of r symbols, whose probabilities have to be estimated "on the fly", the best model is to assume that the input symbol is drawn from a multinomial random variable. The multinomial distribution is characterized by two parameters, namely, the *number* of trials, and a probability vector which determines the probability of a specific event (from a pre-specified set of events) occurring. In this regard, we assume that the number of observations is the number of trials. Therefore, the problem is to estimate the latter probability *vector* associated with the set of possible outcomes or trials.

Specifically, let X be a multinomially distributed random variable, which takes on the values from the set $\{‘1’, \ldots, ‘r’\}$. We assume that X is governed by the distribution $S = [s_1, \ldots, s_r]^T$ as $X = ‘i’$ with probability s_i, where $\sum_{i=1}^{r} s_i = 1$. Also, let $x(n)$ be a concrete realization of X at time 'n'. The intention of the exercise is to estimate S, i.e., s_i for $i = 1, \ldots, r$. We achieve this by maintaining a running estimate $P(n) = [p_1(n), \ldots, p_r(n)]^T$ of S, where $p_i(n)$ is the estimate of s_i at time 'n', for $i = 1, \ldots, r$, with $\sum_{i=1}^{r} p_i(n) = 1$. Then, the value of $p_1(n)$ is updated as per the following simple rule (the rules for other values of $p_j(n)$ are similar):

$$p_1(n+1) \leftarrow p_1 + (1-\lambda)\sum_{j\neq 1} p_j \quad \text{when } x(n) = 1 \qquad (1)$$

$$p_1(n+1) \leftarrow \lambda p_1 \quad \text{when } x(n) \neq 1 \qquad (2)$$

The vector $P(n) = [p_1(n), p_2(n), \ldots, p_r(n)]^T$ refers to the estimate of $S = [s_1, s_2, \ldots, s_r]^T$ at time 'n', and we will omit the reference to time 'n' in $P(n)$ whenever there is no confusion. The above updating rules, with $\lambda \in [0, 1]$ being the learning rate, lead to asymptotic values of P whose mean converges exactly to S. The proof of this property and the properties concerning the variance and convergence of the limiting distribution are found in [9].

4 SLWE Solution to Language Detection and Tracking

By combining the SLWE with mixed-order n-gram models, we present a novel approach to the task of language classification in multilingual WoM documents.

One important issue in all PR systems is that of selecting the feature space of the classifier. The approach we advocate is akin to the ideas of Cavnar and Trenkle, which uses mixed-order n-gram models, and builds n-gram profiles for each language that is being classified. The nature of WoM discussions were also a key motivating factor in choosing n-grams as features, due to their robustness with regard to noise in the input text and that the segments may be too short for word-based features to encapsulate sufficient information.

By utilizing n-grams, there is no need for preprocessing in the sense of spell checking or stemming since n-grams essentially gives us the information-bearing content of a word without performing such costly procedures. In addition, stemming requires sophisticated knowledge about the language, and is thus useless for our task since we do not know the language of the input text. The SLWE also possesses better scalability than, for instance, the MLE, which is used by Ludovik *et al.* [7] in their approach, with regard to a large number of features. Another important motivation for using the SLWE for this task is that there is no need for a separate segmentation process by using complex methods such as dynamic programming used by Ludovik and his co-authors [7]. Instead, the SLWE is able to adapt to changes quickly if the environment switches its probability vector, which in our case is the distribution of top n-grams for the possible languages being classified.

4.1 The Basic Algorithm

The PR system presented here for classification of language in multilingual documents, consisted of two phases. The first phase involved training mixed-order n-gram profiles for each language that the system should support. Only the most frequent n-grams of order $n = 1$ to 4 for a given language were kept in the profile. The second phase of the PR system consisted of the actual classification, or testing phase. In this phase, the estimate of the SLWE was initialized at the beginning of each document, with a feature vector consisting of all unique n-grams from each of the different language profiles. Each document in the testing corpus was processed, and for each document, each word was processed and classified according to a distance measure between the estimated probability vector and each of the language probability distributions. The running estimate of the SLWE was updated after every word was processed.

Training Language Profiles. The training set consisted of monolingual documents, pre-labeled with the language they were written in. Each document in this training set was subjected to a tokenization process. We also removed all non alphanumerical characters from the text. After the tokenization process was done, each word in the document was expanded to their mixed-order n-grams.

After all the n-grams were read, the frequencies were converted into probabilities by dividing each frequency by the total number of observed n-grams. By doing so, we were able to obtain an n-gram probability distribution for the given language.

Classification and Testing. The second phase consisted of classifying each document in the testing corpus, using the SLWE and the probability distributions for each language.

The test documents were generated by our system, by concatenating segments from monolingual documents. This approach made it possible for us to pre-label each segment of the multilingual sample document, allowing us to validate the classification results for each segment.

Each document to be classified was read into the system and was subjected to the same tokenization process as described for the training phase. The feature vector of the SLWE consisted of all the unique n-grams from all the language profiles defined for the system. The SLWE kept a running estimate of this feature vector, where each n-gram was associated with a given probability. These probabilities were initialized evenly.

After the SLWE was initialized, and the document was tokenized into a list of words, the system was ready to perform the actual classification procedure. The formal algorithm is included in the unabridged paper and omitted here due to space limitations.

For each of the words that the sample document contains, the system expanded the word into mixed-order n-grams. Then, for each of these n-grams, the probabilities of the running estimate was updated as per the multinomial updating scheme of the SLWE. If the n-gram is found in the estimate probability vector, its probability was increased according to the updating rules. The probability of all other n-grams were then accordingly reduced. If the n-gram were not in the estimate vector, it was merely ignored.

After all the n-grams for the given word were processed, the system measured a distance between the estimated probability vector and each of the language probability distributions. The word was then classified as being written in the language represented by the language profile that measured the shortest distance from the estimate (using the distance measure alluded to earlier). With the assumption that a sentence is monolingual, we counted the number of words in a sentence and classified the sentence as being written in the language that had the highest word classification count. The validation results are maintained in a so-called *confusion matrix.*

5 Experimental Results

The motivation for these experiments was to investigate how well our algorithm was suited for language classification in multilingual documents, and by testing several different languages we sought to investigate the ability to classify documents written in different languages and how well the classifier would scale with regard to the number of supported languages. We use different values for the cut-off threshold to examine how well the classifier scaled with regard to the number of features, and we experimented with different values for the learning parameter of the SLWE to evaluate the impact of slow versus fast convergence when dealing with language classification. We also measured the accuracy of our classifier operating with different sentence lengths to see how well it is able to deal with short or long sentences.

5.1 Experimental Setup

The classifier was tested on three different sets of languages, generated by concatenating sentences from monolingual documents. The languages used for our testing are English, French, and German for Experiment Set 1, and English English, French, German, Norwegian, Italian, Spanish, Dutch and Swedish for Experiment Set 3. Details of Experiment Set 2 can be found in the unabridged version of this paper. For each of these sets we generated different variants using different sentence lengths. All test sets had a corpus size of 100 documents, except for test set VI which had 200 documents. Test set I, II and III, for experiment set 1, consisted of respectively 10, 15 and 20 words per sentence. The final test set, VI, for experiment set 3, contained 20 words per sentence.

With these test sets we tested our classifier on four different test cases, using different values for the learning parameter, λ, and different cut-off thresholds. Test case A and B used a cut-off threshold of 400, whereas test case C and D

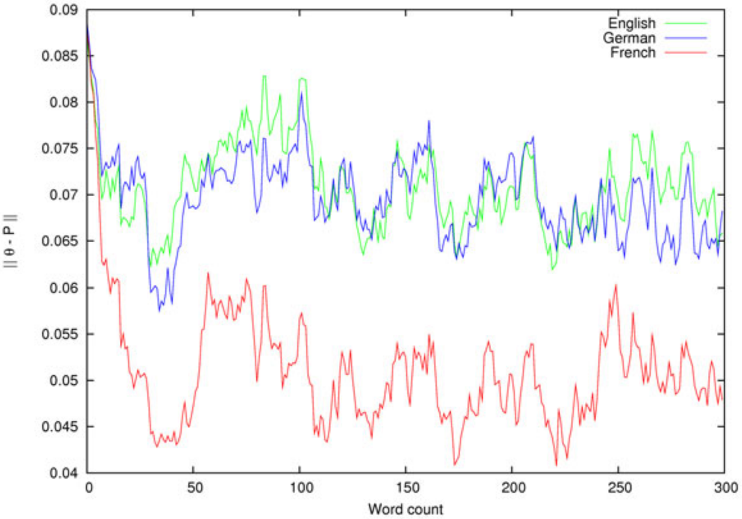


Fig. 1. Plot of the Euclidean distance from the estimated probability vector to each of the language profiles. The document being classified was monolingual, written in French, containing 300 words.

used 500 as the cut-off threshold. For the learning parameter, λ, test case A and C used a value of 0.98. Test case B and D used 0.99 as the learning parameter. Figure 1 shows a plot of the Euclidean distance between the estimate $P(n)$ and three possible language profiles for a document that is monolingual. Despite the document being monolingual, the system assumes that the document is multilingual. The sample being classified contains 300 words written in French and in this example, the classifier operates on word-level, disregarding any sentence boundaries. We observe that the SLWE converges rapidly to the true language profile, which for this sample was French. Even though the variance of the estimate is rather high, we observe that the distance to the other language profiles is far greater than the distance to the correct language profile. We used $\lambda = 0.99$ and 300 as the cut-off threshold in this experiment.

5.2 Results

Language Set 1. The classification accuracy for our first language set is reported for each of the test cases in Table 1.

Table 1. Reported classifier accuracy for each of our test cases for the first language set

Test Set	Test Case	λ	Cut-off	Accuracy (Eng)	Accuracy (Fre)	Accuracy (Ger)
I.	A	0.98	400	0.968	0.962	0.949
I.	B	0.99	400	0.941	0.891	0.920
I.	C	0.98	500	0.970	0.960	0.949
I.	D	0.99	500	0.945	0.905	0.925
II.	A	0.98	400	0.973	0.990	0.987
II.	B	0.99	400	0.951	0.963	0.966
II.	C	0.98	500	0.971	0.992	0.987
II.	D	0.99	500	0.961	0.965	0.974
III.	A	0.98	400	0.996	0.990	0.983
III.	B	0.99	400	0,987	0.986	0.974
III.	C	0.98	500	0.994	0.990	0.983
III.	D	0.99	500	0.988	0.986	0.974

We observe that best accuracy for all the test sets is achieved with the learning parameter λ set to 0.98. Higher values of λ yields slower, but more accurate convergence. When classifying short sentences, it is important that the SLWE is able to converge rather quickly so that as few words as possible in the sentences are misclassified. We also observe that the different cut-off thresholds only to a small extend affects the classifier accuracy.

Table 2 shows the confusion matrix for test case A on test set III, which demonstrated an averaged classifier accuracy of 0.9896. In this experiment, the test set consisted of 520 sentences in English, 515 sentences in French and 465 sentences in German. Each sentence consists of 20 words. By looking at the accuracies listed in Table 2, we observe that only two of the 520 sentences in English were misclassified. One of these as French and the other as German.

Table 2. Confusion matrix for test case A, using test set III

	Eng	**Fre**	**Ger**
Eng	0.996	0.002	0.002
Fre	0.010	0.990	0.000
Ger	0.013	0.004	0.983

Language Set 3. For the last language set we tested our classifier using all eight languages that we had generated language profiles for. For this case we generated the test samples using a sentence length of 20 words. This testing corpus consisted of 200 documents, and the results are listed in Table 3.

Table 3. Reported classifier accuracy for each of our test cases for the third language set with eight different languages

Test Set	**Test Case**	λ	**Cut-off**	**Averaged Acc.**	**Best Acc.**	**Worst Acc.**)
VI.	A	0.98	400	0.9695	0.988 (Fre)	0.928 (Nor)
VI.	B	0.99	400	0.9701	0.986 (Ita)	0.928 (Nor)
VI.	C	0.98	500	0.9690	0.988 (Fre)	0.916 (Nor)
VI.	D	0.99	500	0.9717	0.986 (Ita)	0.931 (Nor)

5.3 Discussion and Summary of Results

We have observed that our classifier is able to classify multilingual documents with high overall accuracy. Our experiments demonstrates that the classifier performs extremely well for moderate-sized segments, and that it performs adequately for shorter sentences with 10 words per sentence.

For the first language set, we obtained a classification accuracy for the English language as high as 0.996 using $\lambda = 0.98$ and the cut-off threshold set to 400. This accuracy was achieved with sentences consisting of 20 words. For shorter segments, with 10 words per sentence, we achieved an accuracy of 0.97. This is still a fairly good accuracy considering the length of the segments. We observe that using a cut-off threshold around 400 yields satisfying results, which is in accordance to the suggested cut-off thresholds used by Cavnar and Trenkle in their experiments. This also shows us that by reducing or increasing the feature space, the classifier scales well and is not notably handicapped by working with a limited feature set compared to a larger one.

For the last language set, using eight different languages, we observed through our experiments that our classifier is able to scale well with regard to the number of supported languages. The averaged accuracy reported for our experiments was slightly lower than for the case when dealing with only five languages, but the classifier still performs well with an error rate of only 0.0283 for eight languages, compared to an error rate of 0.0186 in the case of five languages.

6 Conclusion and Future Work

In this paper we have studied the problems of topic detection and tracking in multilingual online discussions, which is particularly difficult because the content involve the brief and “chatty” opinions of users in multiple languages. Unlike the traditional PR problem, in this scenario, the class-conditional distributions are non-stationary. By using the estimation philosophy recommended in [10], we have proposed a solution to the current problem using novel estimators that are pertinent for non-stationary environments. The classification results obtained for various data sets which involve as many as 8 languages demonstrates that our proposed methodology is both powerful and efficient.

References

1. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: Proceedings of SDAIR 1994, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US, pp. 161–175 (1994)
2. Creutz, M.: Unsupervised segmentation of words using prior distributions of morph length and frequency. In: ACL 2003: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, pp. 280–287. Association for Computational Linguistics (2003)
3. Creutz, M., Lagus, K.: Unsupervised discovery of morphemes (2002)
4. Dunning, T.: Statistical Identification of Language. Technical report MCCS 94-273. New Mexico State University (1994)
5. Ingle, N.C.: A language identification table. The Incorporated Linguist. 15(4), 98–101 (1976)
6. Jang, Y.M.: Estimation and Prediction-Based Connection Admission Control in Broadband Satellite Systems. ETRI Journal 22(4), 40–50 (2000)
7. Ludovik, Y., Zacharski, R.: Multilingual document language recognition for creating corpora. Technical report, New Mexico State University (1999)
8. Mandl, T., Shramko, M., Tartakovski, O., Womser-Hacker, C.: Language identification in multi-lingual web-documents. In: Kop, C., Fliedl, G., Mayr, H.C., Métais, E. (eds.) NLDB 2006. LNCS, vol. 3999, pp. 153–163. Springer, Heidelberg (2006)
9. Oommen, B.J., Rueda, L.: Stochastic Learning-based Weak Estimation of Multinomial Random Variables and Its Applications to Non-stationary Environments. Pattern Recognition (2006) (in Press)
10. Oommen, B.J., Rueda, L.: Stochastic Learning-based Weak Estimation of Multinomial Random Variables and Its Applications to Non-stationary Environments. Pattern Recognition 39(1), 328–341 (2006)
11. Ozbek, G., Rosenn, I., Yeh, E.: Language classification in multilingual documents. Technical report, Stanford University (2006)
12. Souter, C., Churcher, G., Hayes, J., Hughes, J., Johnson, S.: Natural language identification using corpus-based models. Hermes Journal of Linguistics 13, 183–203 (1994)
13. Ziegler, D.: The automatic identification of languages using linguistic recognition signals. PhD thesis, Buffalo, NY, USA (1991)

Similarity Word-Sequence Kernels for Sentence Clustering

Jesús Andrés-Ferrer, Germán Sanchis-Trilles, and Francisco Casacuberta

Instituto Tecnológico de Informática
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
{jandres,gsanchis,fcn}@dsic.upv.es

Abstract. In this paper, we present a novel clustering approach based on the use of kernels as similarity functions and the C-means algorithm. Several word-sequence kernels are defined and extended to verify the properties of similarity functions. Afterwards, these monolingual word-sequence kernels are extended to bilingual word-sequence kernels, and applied to the task of monolingual and bilingual sentence clustering. The motivation of this proposal is to group similar sentences into clusters so that specialised models can be trained for each cluster, with the purpose of reducing in this way both the size and complexity of the initial task. We provide empirical evidence for proving that the use of bilingual kernels can lead to better clusters, in terms of intra-cluster perplexities.

1 Introduction

Text categorisation [1] is the task of finding the class to which a given document belongs to. The categories or classes in which a document can be classified are known beforehand, and, usually, a database of documents with their corresponding category is enough for training an automatic categorisation system. Several approaches have been applied to text categorisation, ranging from naive Bayes classifiers [1] to support vector machines (SVM) [2,3].

A variant of text categorisation is the text clustering task. Unlike text categorisation, in text clustering we do not know the classes into which the documents should be classified, which means that the only data available is a database of documents without class information. Therefore, text clustering is entailed as a more difficult task than text categorisation. Several attempts have been made in text clustering. For instance, in [4] several kernel-based text categorisation techniques are adapted to text clustering by using the C-means algorithm.

An especially appealing problem in document clustering is sentence clustering, in which each document is made up of only one single sentence. This problem has been receiving special attention in the natural language processing (NLP) community since it allows for training specific models for each of the obtained clusters, leading to more task-focused models [5,6]. Moreover, sentence clustering can also be of interest for kernel-based methods when applied to NLP tasks, such as done for text recognition [7] or statistical machine translation [8,9]. In

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 610–619, 2010.

these scenarios, kernel methods often suffer scalability problems, and sentence clustering is a natural way in which the training data can be divided so as to obtain smaller (but more specific) models.

Throughout literature, the "sum-of-squares" cost for a given data set, forms the basis for a number of clustering methods [10]. The aim of such clustering algorithms is to partition a data set of N samples, $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$, into C clusters, so as to minimise the intra-cluster mean squared error. The standard algorithm, also known as Lloyd's algorithm or C-means [11], relies on assigning each data point $\boldsymbol{x}_n$ to the cluster with the closest mean. Once all the data points have been assigned, the means of each cluster are updated according to the samples contained within it. Then, the data points are reassigned to the cluster with the closest mean, and this procedure is repeated iteratively until no data points are changed. The C-means algorithm is considered a fast clustering method because it is a sub-optimal algorithm that does not require the computation of the distance matrix between all samples. One of the disadvantages of C-means is that it is unable to find suitable clusters whenever the given data are not linearly separable, leading to degenerated solutions in which the number of clusters computed exceeds the desirable amount. In order to circumvent this problem, M. Girolami proposed in [12] an extension of C-means that relies on a transformation of the original sample $\boldsymbol{x}$ into a higher-dimensionality feature space $\phi(\boldsymbol{x})$. Although such proposal is based on the computation of Mercer kernels [13], it still relies on the distance metric of the original C-means algorithm.

Kernel methods have attracted much interest since they were introduced by V. Vapnik [14]. Traditionally being applied to classification problems in the form of Support Vector Machines (SVM) [15,16], kernel methods rely on the idea of establishing a mapping from the current feature space to a higher-dimensionality feature space, with the purpose of achieving linear separability among classes which are non-separable in the current feature space. Under this perspective, a kernel function between two data points is defined as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^T \phi(\boldsymbol{x}'), \tag{1}$$

where $\boldsymbol{x}$ and $\boldsymbol{x}'$ are the data points considered and $\phi(\boldsymbol{x})$ is the mapping function to a higher-dimensionality feature space. From the definition above, it is clear that a kernel is a symmetric function, i.e., $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}', \boldsymbol{x})$.

Given that kernels are implemented as a dot product in a metric space, some kernels can also be used to measure the *similarity* (or distance) between the data points considered and are appropriate for *direct* application within distance-based clustering algorithms.

In this paper, we present a new approach for using similarity Mercer kernels for clustering based on the C-means algorithm. This new approach is evaluated in practise under the scope of sentence clustering and bilingual sentence clustering.

This paper is structured as follows: in the next Section, we extend the C-means algorithm for using kernel methods. In Section 3 word-sequence kernels are introduced, and they are extended to *bilingual* word-sequence kernels in the following Section. The empirical results are gathered in Section 5, and concluding remarks are discussed in Section 6.

2 Kernel-Based C-Means Clustering

The C-means algorithm [11] seeks to minimise the sum-of-squares distance from each sample to the centre of the cluster it belongs to. Given a number of categories C, the algorithm finds a local optimum for for the following minimisation

$$\hat{\boldsymbol{z}} = \arg\min_{\boldsymbol{z}} \left\{ \frac{1}{N} \sum_{c=1}^{C} \sum_{n=1}^{N} z_{nc}\, \mathrm{d}(\boldsymbol{x}_n, \boldsymbol{m}_c) \right\}, \tag{2}$$

where $z_{nc} = 1$ if $\boldsymbol{x}_n$ belongs to the c-th cluster and 0 otherwise, and with $\boldsymbol{m}_c$ being the centre of the c-th cluster, $\boldsymbol{m}_c = N_c^{-1} \sum_{n=1}^{N} z_{nc}\boldsymbol{x}_{nc}$, where N_c stands for the number of samples in the c-th cluster, i.e., $N_c = \sum_{n=1}^{N} z_{nc}$. The function $\mathrm{d}(\boldsymbol{x}_n, \boldsymbol{m}_c)$ is a distance function between the sample $\boldsymbol{x}_n$ and the centre $\boldsymbol{m}_c$, usually the euclidean distance

$$\mathrm{d}(\boldsymbol{x}_n, \boldsymbol{m}_c) = (\boldsymbol{x}_n - \boldsymbol{m}_c)^T (\boldsymbol{x}_n - \boldsymbol{m}_c). \tag{3}$$

The distance used by the C-means algorithm can either be a semi-metric or a metric, depending on whether the triangle inequality is verified or not.

In [12], C-means was extended with the help of Mercer kernels by changing the distance and the centres of the standard algorithm, so that it can better handle non linearly-separable data. The distance proposed in [12] is given by

$$\mathrm{d}(\boldsymbol{x}_n, \boldsymbol{m}_c) = (\phi(\boldsymbol{x}_n) - \boldsymbol{m}_c)^T (\phi(\boldsymbol{x}_n) - \boldsymbol{m}_c), \tag{4}$$

with $\boldsymbol{m}_c = N_c^{-1} \sum_{n=1}^{N} \boldsymbol{z}_{nc}\phi(\boldsymbol{x}_{nc})$.

Since kernel functions are symmetric, they only need to verify two more conditions to be a semi-metric distance. Therefore, the kernel itself can be used as the distance inside the C-means algorithm. Moreover, if a given kernel also verifies the triangle inequality, then the kernel itself can be used as a metric distance. However, many kernels are more naturally redefined as similarity functions instead of distances. Given a distance, a similarity can be defined and vice-versa. In such case, C-means can be re-defined in terms of similarities as follows

$$\hat{\boldsymbol{z}} = \arg\max_{\boldsymbol{z}} \left\{ \frac{1}{N} \sum_{c=1}^{C} \sum_{n=1}^{N} z_{nc}\, \mathrm{s}(\boldsymbol{x}_n, \boldsymbol{m}_c) \right\}, \tag{5}$$

with $\boldsymbol{m}_c = N_c^{-1} \sum_{n=1}^{N} z_{nc}\phi(\boldsymbol{x}_{nc})$, and where $\mathrm{s}(\boldsymbol{x}_n, \boldsymbol{m}_c) = \phi(\boldsymbol{x}_{nc})^T \boldsymbol{m}_c$ is assumed to be a (semi-)metric kernel, i.e., a kernel that measures the similarity generated by a (semi-)metric distance. In this work, several (semi-)metric kernels are proposed so that they can be used within the similarity version of C-means. However, the approaches in Eqs. (2) and (5) could be related in some way.

3 Word-Sequence Kernels

Recently, Word-sequence Kernels (WSK) were introduced in [17]. The main purpose of WSK is to compute document similarity based on matching non-consecutive sequences of words. WSK are defined as a mapping $\Sigma^n \rightarrow \mathbb{R}^{|\Sigma|^n}$, where n stands for the maximum length of the segment to be considered. For a given order n and document pair $(\boldsymbol{x}, \boldsymbol{x}')$, we define the following kernel

$$K_n(\boldsymbol{x}, \boldsymbol{x}') = \sum_{u \in \Sigma^n} |\boldsymbol{x}|_u |\boldsymbol{x}'|_u, \tag{6}$$

where $|\boldsymbol{x}|_u$ stands for the number of occurrences of u in document $\boldsymbol{x}$. In their work, [17] reported interesting improvements by using WSK when applied to text categorisation tasks. However, the best results were achieved using a small order, $n = 2$.

Although the kernel defined in Eq. (6), is intuitively correct, it does not verify some of the requirements to be a semi-metric similarity. Hence, we also define the kernel K_n^1 as follows:

$$K_n^1(\boldsymbol{x}, \boldsymbol{x}') = \sum_{u \in \Sigma^n} 1_u(\boldsymbol{x}) 1_u(\boldsymbol{x}'), \tag{7}$$

where $1_u(\boldsymbol{x}) = 1$ if u appears in $\boldsymbol{x}$, and 0 otherwise.

The intuitive justification for defining this last kernel can be explained with a small example. We start by defining the following strings:

$$\boldsymbol{s}_1 = \{abcb\} \quad \boldsymbol{s}_2 = \{abab\}$$
$$\boldsymbol{s}_3 = \{abeb\} \quad \boldsymbol{s}_4 = \{abcbab\}$$

One would state that $\boldsymbol{s}_1$ is as similar to $\boldsymbol{s}_2$ as to $\boldsymbol{s}_3$, under the prior assumption of a Levenshtein distance. However, $K_2(\boldsymbol{s}_1, \boldsymbol{s}_2) = 2$ and $K_2(\boldsymbol{s}_1, \boldsymbol{s}_3) = 1$. This is exactly the reason why we introduce kernel K_2^1, since $K_2^1(\boldsymbol{s}_1, \boldsymbol{s}_2) = K_2^1(\boldsymbol{s}_1, \boldsymbol{s}_3) = 1$. On the other hand, the similarity of $\boldsymbol{s}_1$ with itself is $K_2^1(\boldsymbol{s}_1, \boldsymbol{s}_1) = 3$, which is the same than that of $\boldsymbol{s}_1$ with $\boldsymbol{s}_4$, $K_2^1(\boldsymbol{s}_1, \boldsymbol{s}_4) = 3$. This is because the kernel K_2^1 is a pseudo-metric similarity. It is worth noting that K_n is not a pseudo-metric, which implies that a given element, as in the example $\boldsymbol{s}_4$, may be more similar to a given element, such as $\boldsymbol{s}_1$, than the element itself, i.e. $K_2(\boldsymbol{s}_1, \boldsymbol{s}_4) = 4 > K_2(\boldsymbol{s}_1, \boldsymbol{s}_1) = 3$. Such problem cannot be underestimated, since it can imply that C-means will fail to converge.

To solve this undesirable property, K_n^1 is redefined using a normalisation score depending on the different number of n-grams of the sample, i.e.,

$$\hat{K}_n^1 = \sum_{u \in \Sigma^n} \frac{1_u(\boldsymbol{x})}{\sqrt{\sum_{v \in \Sigma^n} 1_v(\boldsymbol{x})}} \frac{1_u(\boldsymbol{x}')}{\sqrt{\sum_{v \in \Sigma^n} 1_v(\boldsymbol{x}')}} \tag{8}$$

The kernel defined in Eq. (8) solves the previously outlined problem, i.e. the similarity of $\boldsymbol{s}_1$ with itself is $\hat{K}_2^1(\boldsymbol{s}_1, \boldsymbol{s}_1) = 1$, which is larger than the similarity of $\boldsymbol{s}_1$ with $\boldsymbol{s}_4$, $\hat{K}_2^1(\boldsymbol{s}_1, \boldsymbol{s}_4) = 0.866$. With this last kernel, we achieve a very desirable property for its use within C-means, i.e., a given element achieves maximum similarity only when it is compared with itself.

Similarly, we also redefine the kernel K_n,

$$\hat{K}_n(\boldsymbol{x}, \boldsymbol{x}') = \sum_{u \in \Sigma^n} \frac{|\boldsymbol{x}|_u}{\sqrt{\sum_{v \in \Sigma^n} |\boldsymbol{x}|_v}} \frac{|\boldsymbol{x}'|_u}{\sqrt{\sum_{v \in \Sigma^n} |\boldsymbol{x}'|_v}} \tag{9}$$

However, the re-normalised version of K_n, $\hat{K}_n$, only reduces the cases in which the problem of not being a semi-metric can appear, but it does not solve it.

Given the definition in Eq. (9), a WSK $\bar{K}_n$ is defined as

$$\bar{K}_n(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{n} \hat{K}_i(\boldsymbol{x}, \boldsymbol{x}'). \tag{10}$$

Analogously as done above, the kernel defined in Eq. (8) is extended to $\bar{K}_n^1$.

4 Bilingual Word-Sequence Kernels

In [17], cross-lingual WSK were also defined, by first defining a *soft* matching WSK and assuming that the samples being considered (i.e. $\boldsymbol{x}$ and $\boldsymbol{x}'$) belonged to different languages. In this context, soft matching refers to a probabilistic matching, i.e. a matching that does not require both samples to have exactly identical parts. By doing so, they were able to find similarities between documents written in different languages.

Our purpose, however, is not to perform cross-lingual classification (or clustering). Our case is different, since we assume that we have a sentence-aligned bilingual corpus and we intend to cluster the data by taking into account such bilingual information. Hence, we need to define a *bilingual* WSK (BWSK), which can be easily extended from the one defined in Eq. (10) by taking into account two different vocabularies, namely Σ for the source language and Δ for the target language. Let be $\boldsymbol{w} = \{\boldsymbol{x}, \boldsymbol{y}\}$ a bilingual sentence pair, where $\boldsymbol{x}$ is the sentence belonging to the source language and $\boldsymbol{y}$ is the sentence belonging to the target language. Then, a BWSK can be defined as

$$B_n(\boldsymbol{w}, \boldsymbol{w}') = K_n(\boldsymbol{x}, \boldsymbol{x}') + K_n(\boldsymbol{y}, \boldsymbol{y}') = \sum_{u \in \Sigma^n} |\boldsymbol{x}|_u |\boldsymbol{x}'|_u + \sum_{v \in \Delta^n} |\boldsymbol{y}|_v |\boldsymbol{y}'|_v \tag{11}$$

Note that $B_n(\boldsymbol{w}, \boldsymbol{w}')$ is a kernel because it can be expressed as the sum of two kernels, which is a valid kernel composition rule.

As done for the monolingual case, we can also define $B_n^1(\boldsymbol{w}, \boldsymbol{w}')$; and all its extensions: $\hat{B}_n^1(\boldsymbol{w}, \boldsymbol{w}')$,$\bar{B}_n^1(\boldsymbol{w}, \boldsymbol{w}')$; and $\hat{B}_n(\boldsymbol{w}, \boldsymbol{w}')$,$\bar{B}_n(\boldsymbol{w}, \boldsymbol{w}')$.

5 Experiments

We ran most of our clustering experiments on the BTEC (Basic Travel Expression Corpus), which is the corpus provided for the IWSLT[1] statistical machine translation campaign. The BTEC corpus includes several bilingual, sentence-aligned sub-corpora, among which we selected the Chinese-English one. The figures of this corpus are summarised in Table 1. Prior to performing clustering on the data, all English words were lowercased. This was not necessary on the Chinese side since Chinese has no case information.

[1] http://mastarpj.nict.go.jp/IWSLT2009/

The problem of automatically measuring the quality of the produced clusters was addressed by means of the perplexity concept. In natural language procesing (NLP) the perplexity of a test set ($\boldsymbol{w} = w_1, \ldots, w_L$) is defined as follows:

$$ppl(\boldsymbol{w}) = 2^{\frac{1}{L} \log_2 p(\boldsymbol{w})}, \tag{12}$$

where $p(\boldsymbol{w})$ is the probability of the test set accordingly to a language model. The intuitive meaning of perplexity is the average number of words that can follow a given word, according to a given language model. For instance, if the perplexity for a given data set is 28, it means that in order to predict the word which follows a given prefix, a total average of 28 different words should be taken into account. Hence, the perplexity for a given data set according to the language model trained on that same data is a measure of how compact (i.e. not sparse) the data is. We will be assessing the quality of the clusters using the intra-cluster perplexity (IC-PPL) average, measured on the English data, given by

$$ppl_{avg} = 2^{\sum_{c=1}^{C} \frac{1}{C} \frac{1}{W_c} \log_2 p(c)}, \tag{13}$$

where $p(c)$ is the probability of the samples of cluster c according to the language model estimated on that same cluster; W_c is the total number of words in the sentences belonging to the cluster c; and C is the total number of clusters. Since we will be computing clusters using the kernels proposed in Secs. 3 and 4 with the order n ranging from 1 up to 4; we decided to compute IC-PPL based on a 5-gram language model computed using SRILM toolkit [18]. Furthermore, since cluster sizes dropped to less than 1 000 sentences in some cases, we decided to smooth these models with the interpolated version of Knesser-Ney smoothing [19]. For consistency purposes, the perplexities listed in Table 1 are also smoothed with the same smoothing.

Table 1. Statistics of the BTEC corpus. K stands for thousands of elements

Language	N. Sentences	Running words	Vocabulary	Perplexity
Chinese	20K	172K	8428	24.3
English	20K	183K	7298	20.8

In preliminary investigation, we also researched the use of average edit distance from each sentence of a cluster to all other sentences in the same cluster as quality metric, but the differences reported were similar to those reported by IC-PPL, which is much faster to compute.

One way to reduce the computational requirements of our clustering algorithm without any loss of information was to remove all singletons, since their effect on the calculation of the kernels is minimum, if any. For a similar reasons, we also decided to remove stop words, since if a word appears in almost every sentence, then its discriminative capacity should not be very high either.

We computed 2 to 20 clusters of the training data, with steps of two, for all kernels described in Sections 3 and 4. Since the C-means algorithm needs a

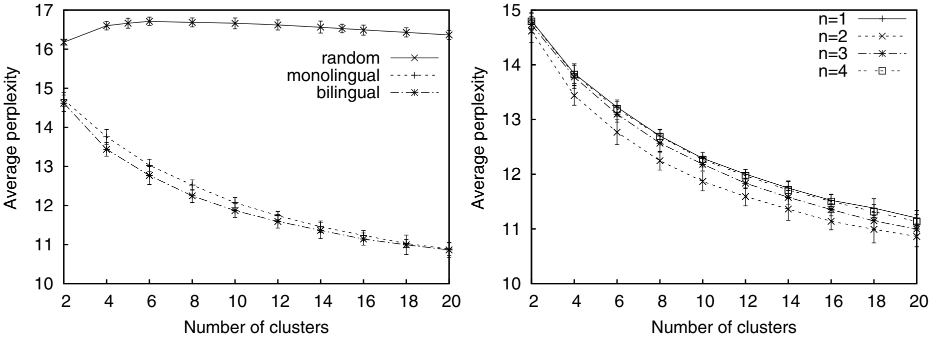


Fig. 1. Clusterings for the BTEC corpus. Left: IC-PPL for random, $\hat{K}_2^1$ (monolingual) and $\hat{B}_2^1$ (bilingual). Right: IC-PPL for $\bar{B}_1^1$,$\bar{B}_2^1$,$\bar{B}_3^1$ and $\bar{B}_4^1$ (right).

random initialisation, we performed 20 repetitions of each experiment and report the average and the confidence interval at 95%.

Let us now analyse the results in detail. First, we show in Figure 1 the performance of the $\bar{K}_2^1$ and $\bar{B}_2^1$ kernels when used in C-means, as compared to a random clustering. Other proposed kernels, such as e.g. $\bar{K}_1^1$ or $\bar{B}_3$, present a similar behaviour, and are not shown here for the sake of simplicity. The first thing to be noted is that IC-PPL stays almost unchanged for every number of clusters considered in the case of random clustering, whereas for the kernel-clusterings IC-PPL drops logarithmically when increasing the number of clusters. This fact was actually expected: if we consider 20K clusters (as many as sentences in the corpus), IC-PPL will eventually drop to 1. However, since we are only considering up to 20 clusters and an average of 1000 sentences are included into each cluster, perplexity will only drop when such grouping is done in an informed way. It seems that considering bilingual information has beneficial effects since BWSK lead to smaller IC-PPL than regular WSK.

As for the effect of considering different n orders, in Figure 1 we show the result of comparing $\bar{B}_1^1$,$\bar{B}_2^1$,$\bar{B}_3^1$ and $\bar{B}_4^1$. Again, other kernels such as the monolingual ones, perform similarly and are omitted for clarity. We can see that the best performance is given by $\bar{B}_2^1$, and that increasing the order of n above 2 does not provide further improvements, but rather has a degrading effect on IC-PPL. This fact is consistent with what [17] reported for document classification tasks.

In order to check the scalability of the results reported on the BTEC corpus to other larger corpora, we also performed some experiments on a reduced version of the Spanish–English Europarl corpus [20]. Such version was restricted to maximum sentence length of 20 (Euro<20), for both English and Spanish. The statistics of this corpus are summarised in Table 2. As for the BTEC corpus, we will measure cluster quality with IC-PPL measured on the English side.

The first thing that we notice when observing Figure 2 is that the monolingual and the bilingual kernel clusterings behave similarly, as with the BTEC corpus. This is probably due to the fact that, once the monolingual information

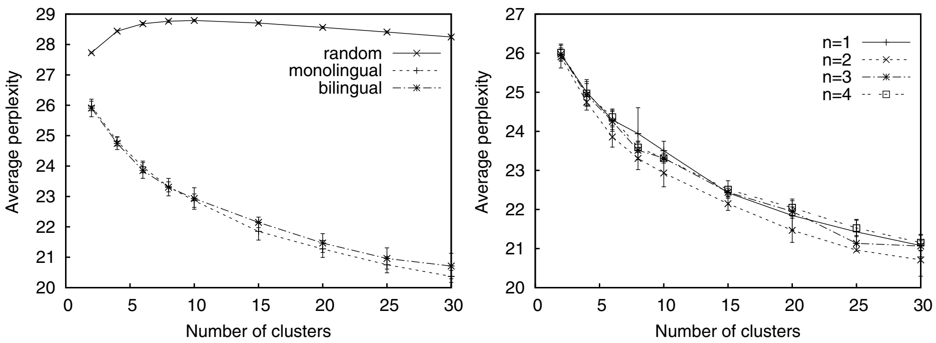


Fig. 2. Clusterings for Europarl with maximum sentence length of 20 (Euro<20). Left: IC-PPL for random, $\bar{K}_2^1$ and $\bar{B}_2^1$. Right: IC-PPL for $\bar{B}_1^1$,$\bar{B}_2^1$,$\bar{B}_3^1$ and $\bar{B}_4^1$.

Table 2. Statistics of the Spanish–English Europarl corpus, when restricted to maximum sentence length of 20. K stands for thousands of elements.

Language	N. Sentences	Running words	Vocabulary	perplexity
Spanish	312K	4.0M	58K	28.2
English	312K	3.9M	37K	26.7

is added to the cluster, the bilingual information is not able to produce a further refinement over the initial clusters. Nevertheless, thought not statistically significant, it can be observed that for small number of clusters, the bilingual information does seem to help, but as the number of clusters increase the bilingual information tends to confuse the clustering algorithm. These results, which are similar to the results observed with BTEC, suggest that bilingual information only helps when there is a large amount of samples within a cluster. This can be due to the fact that the smaller the clusters, the more focused they become in specific word-sequences, and the more easily extra-cluster information can distort them. When dealing with large cluster sizes, however, introducing bilingual information may help to group word-sequences which are not so similar in the English side, but appear more similar in their bilingual counterpart.

As for increasing the order of n in the Euro<20 corpus, a similar behaviour as for BTEC is be observed in Figure 2. Similarly to the results shown above, $\bar{B}_2^1$ seems to be the best performing kernel in terms of IC-PPL. In order to understand the reason why $n = 2$ is the best performing kernel-family, let us have a closer look at some statistics of the corpora considered. Looking at Table 3, it is quite obvious why increasing the order of n above 2 does not provide any improvements: only 10% of the trigrams and 5% of the 4-grams appear more than twice. This means that such features, when introduced into the clustering algorithm via WSK or BWSK will most likely only introduce noise.

As for the difference between the families of kernels defined by $|\boldsymbol{x}|_u$ and $1_u(\boldsymbol{x})$, our experiments show that they are indistinguishable according to IC-PPL. This

Table 3. Statistics of the IWSLT and Euro<20 in terms of singletons and doubletons. Single stands for singletons and double for doubletons. All data are in %.

	1-grams		2-grams		3-grams		4-grams	
Corpus	single	double	single	double	single	double	single	double
BTEC	43.8	14.0	65.3	13.6	79.0	10.5	87.5	7.5
Euro<20	36.7	13.3	62.7	13.3	78.9	9.8	88.4	6.2

is due to the fact that, although the theoretical motivation is clear as seen in Section 3, in practise it is not very often the case that a given n-gram occurs more than once within a single sentence – not for unigrams and even less for bigrams. In fact, nearly no bigram happens twice in a single sentence once stop-words have been removed. This implies that K_n is practically equivalent to K_n^1 (and all the variations thereof).

6 Conclusions and Future Work

In this work, we have proposed the direct use of kernels as similarity measure, and applied it to the specific case of sentence clustering via C-means. Specifically, we have described several families of kernels suitable for this task, and shown that the $\bar{B}_2$ and $\bar{B}_2^1$ kernels are the ones which perform the best. Although for other corpora it might be beneficial to increase the order of n, such corpora should be less sparse if improvements are to be expected. It is also observed that, in order to take full advantage of bilingual information, cluster sizes need to be larger.

As most of the cluster quality measures, such as cluster sparseness, IC-PPL does not provide any insight towards deciding the optimal number of clusters, C. For finding the optimal number of cluster, a possiblity is to use the bayesian scheme proposed in [12].

Given the generality provided by using kernels as similarity measure, the C-means algorithm used in this paper can be easily extended by just adding more components while sticking to the kernel composing rules. In this way, we plan to introduce other features inherent to NLP tasks, such as part-of-speech tags, automatic word classes, n-gram probability, or even bilingual lexicon probability for the case of bilingual kernels. We plan to address these issues in future works.

Acknowledgements

This paper is based upon work supported by the EC (FEDER, FSE), the Spanish Government (MICINN, MITyC, ”Plan E” under grants MIPRCV “Consolider Ingenio 2010”, iTrans2 TIN2009-14511 and erudito.com TSI-020110-2009-439) and the Generalitat Valenciana (grant Prometeo/2009/014).

The authors would also like to thank the anonimous reviewers for their constructive and detailed comments.

References

1. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: Proc. of AAAI/ICML 1998 Workshop on Learning for Text Categorization, pp. 41–48. AAAI Press, Menlo Park (1998)
2. Joachims, T.: Text categorisation with support vector machines: learning with many relevant features. In: Proceedings of 10th ECML, pp. 137–142 (1998)
3. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.J.C.H.: Text classification using string kernels. JMLR 2, 419–444 (2002)
4. Karatzoglou, A., Feinerer, I.: Text clustering with string kernels in r. In: Proc. of the 30th Annual Conference of the Gesellschaft für Klassifikation e.V., Freie Universität Berlin (March 2006)
5. Sanchis-Trilles, G., Cettolo, M.: Online language model adaptation via n-gram mixtures in statistical machine translation. In: Proc. of 14th Annual Conference of the European Association for Machine Translation, Saint-Raphaël, France, (May 27-28, 2010)
6. Lagarda, A., Juan, A.: Topic detection and classification techniques. In: WP4 deliverable, TransType2 (2003)
7. Cortes, C., Mohri, M., Weston, J.: A general regression technique for learning transductions. In: Proc. of 22nd. ICML, pp. 153–160. ACM, NY, USA (2005)
8. Serrano, N., Andrés-Ferrer, J., Casacuberta, F.: On a kernel regression approach to machine translation. In: IbPRIA 2009. LNCS, vol. 5524, pp. 394–401. Springer, Heidelberg (2009)
9. Szedmak, Z.W.S.T.: Kernel regression based machine translation, pp. 185–188. Association for Computational Linguistics (2007)
10. Jain, A.K., Dubes, R.C.: Algorithms for clustering data. Prentice-Hall, Inc., Upper Saddle River (1988)
11. Lloyd, S.P.: Least squares quantization in pcm. IEEE Transactions on Information Theory 28(2), 129–137 (1982)
12. Girolami, M.: Mercer kernel based clustering in feature space. IEEE Transactions on Neural Networks (2001)
13. Mercer, J.: Functions of positive and negative type and their connection with the theory of integral equations. Philosophical Transactions of the Royal Society London (A) 209, 415–446 (1909)
14. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
15. Boser, B., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Annual Workshop on Computational Learning Theory, pp. 144–152 (1992)
16. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
17. Cancedda, N., Gaussier, E., Goutteand, C., Renders, J.: Word-sequence kernels. Journal of Machine Learning Research 3, 1059–1082 (2003)
18. Stolcke, A.: SRILM – an extensible language modeling toolkit. In: Proc. of ICSLP 2002, September 2002, pp. 901–904 (2002)
19. Kneser, R., Ney, H.: Improved backing-off for m-gram language modeling. In: IEEE Int. Conf. on Acoustics, Speech and Signal Processing, vol. II, pp. 181–184 (May 1995)
20. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: Proc. of the MT Summit X, pp. 79–86 (2005)

Bayesian Adaptation for Statistical Machine Translation

Germán Sanchis-Trilles and Francisco Casacuberta

Instituto Tecnológico de Informática
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
{gsanchis,fcn}@dsic.upv.es

Abstract. In many pattern recognition problems, learning from training samples is a process that requires important amounts of training data and a high computational effort. Sometimes, only limited training data and/or limited computational resources are available, but there is also available a previous system trained for a closely related task and with enough training material. This scenario is very frequent in statistical machine translation and adaptation can be a solution to deal with this problem. In this paper, we present an adaptation technique for (state-of-the-art) log-linear modelling based on the well-known Bayesian learning paradigm. This technique has been applied to statistical machine translation and can be easily extended to other pattern recognition areas in which log-linear models are used. We show empirical results in which a small amount of adaptation data is able to improve both the non-adapted system and a system that optimises the above-mentioned weights only on the adaptation set.

1 Introduction

Adaptation in pattern recognition is the task of porting a system trained on a specific task or domain so that it can be used in a different environment. This problem is particularly challenging in natural language processing and other fields where the process of acquiring labelled training samples from a specific domain or task is very costly, but a large collection of labelled data from a similar task is already available. Hence, the challenge consists in being able to modify the original models in such a way, that we are able to take advantage of such large amounts of data available while having at our disposal only very limited amounts of adaptation data.

The adaptation problem is a very common problem in statistical machine translation (SMT), where it is very common to have very large collections of bilingual data belonging to e.g. proceeedings from international entities such as the European Parliament, the Canadian Parliament or the United Nations. However, if we are currently interested in translating e.g. printer manuals, we will need to find a way in which we can take advantage of such data.

The grounds of modern SMT, a pattern recognition approach to machine translation, were established in [1], where the problem of machine translation

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 620–629, 2010.

was defined as follows: given a sentence $\boldsymbol{x}$ from a certain source language, an equivalent sentence $\hat{\boldsymbol{y}}$ in a given target language that maximises the posterior probability is to be found. Such a statement can be specified, according to the Bayes decision rule, as follows:

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y}} \Pr(\boldsymbol{y}|\boldsymbol{x}) \tag{1}$$

Recently, a direct modelling of the posterior probability $Pr(\boldsymbol{y}|\boldsymbol{x})$ has been widely adopted, and, to this purpose, different authors [2,3] proposed the use of the so-called log-linear models, where

$$\Pr(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp \sum_{k=1}^{K} \lambda_k h_k(\boldsymbol{x}, \boldsymbol{y})}{\sum_{\boldsymbol{y}'} \exp \sum_{k=1}^{K} \lambda_k h_k(\boldsymbol{x}, \boldsymbol{y}')} \tag{2}$$

and the decision rule is given by the expression

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y}} \sum_{k=1}^{K} \lambda_k h_k(\boldsymbol{x}, \boldsymbol{y}) \tag{3}$$

where $h_k(\boldsymbol{x}, \boldsymbol{y})$ is a score function representing an important feature for the translation of $\boldsymbol{x}$ into $\boldsymbol{y}$, as for example the language model of the target language, a reordering model or several translation models. K is the number of models (or features) and λ_k are the weights of the log-linear combination. Typically, the weights $\boldsymbol{\Lambda} = \lambda_1 \ldots \lambda_K$ are optimised with the use of a development set.

Log-linear models implied an important break-through in SMT, allowing for a significant increase in translation quality. In addition, log-linear models have also been applied successfully in other pattern recognition tasks, such as text recognition [4] and speech recognition [5]. In this work, we present a Bayesian technique for adapting the weights of such log-linear models according to a small set of adaptation data. Such technique, although applied to SMT in the current paper, is easily extensible to other fields were log-linear models are used.

The rest of this paper is structured as follows. In the next Section, we perform a brief review of current approaches to adaptation and Bayesian learning in SMT. Section 3 describes the typical procedure for weight optimisation in SMT. In Section 4, we present the way in which we apply Bayesian adaptation (BA) to log-linear models in SMT. In Section 5, experimental design and experimental results are detailed. Finally, conclusions and future work are explained in Section 6.

2 Related Work

Adaptation in SMT is a research field that is receiving an increasing amount of attention. One of the first approaches to this task was performed by [6], in which the translation model (TM) is implemented as an unsupervised multinomial mixture of TMs, where each one was supposed to concentrate most of its probability mass in a certain topic. Later, [7] applied other adaptation techniques to interactive machine translation, following the ideas by [8] and adding cache language models (LM) and TMs to their system. In [9], different ways to combine available data belonging to two different sources was explored; in [10] similar experiments were performed, but considering only additional source data. In [11], alignment model mixtures were explored as a way of performing topic-specific adaptation, the alignments being used only to extract phrases. Finally, other authors [12,13], have proposed the use of clustering in order to extract the sub-domains of a large parallel corpus and build more specific LMs and TMs, which are re-combined in test time.

With respect to BA in SMT, the authors are not aware of any work up to the date that follows such paradigm. Nevertheless, there have been some recent approaches towards dealing with SMT from the Bayesian learning point of view, such as [14], in which Bayesian learning is applied in order to estimate appropriate word-alignments within a synchronous grammar.

3 Weight Optimisation in SMT

One of the most popular instantiations of log-linear models in SMT are phrase-based models [15,16]. Phrase-based models allow to capture contextual information to learn translations for whole phrases instead of single words. The basic idea of phrase-based translation is to segment the source sentence into phrases, then to translate each source phrase into a target phrase, and finally to reorder the translated target phrases in order to compose the target sentence. For this purpose, phrase-tables are produced, in which a source phrase is listed together with several target phrases and the probability of translating the former into the latter. Phrase-based models were employed throughout this work.

Typically, the weights $\boldsymbol{\Lambda}$ of the log-linear combination in Equation 3 $\boldsymbol{\Lambda}$ are optimised by means of Minimum Error Rate Training (MERT) [17]: first, n-best hypotheses are extracted for each one of the sentences of a given development set. Next, the optimum $\boldsymbol{\Lambda}$ is computed so that the best hypotheses in the n-best list, according to a reference translation and a given metric, are the ones that the search algorithm would produce. These two steps are repeated until convergence, where the weight vector $\boldsymbol{\Lambda}$ remains unchanged.

This approach has two main problems. On the one hand, it heavily relies on having a fair amount of data available as development set. On the other hand, it *only* relies on the data in the development set. These two problems have as consequence that, if the development set made available to the system is not big enough, MERT will most likely become unstable and fail in obtaining an appropriate weight vector $\boldsymbol{\Lambda}$. In addition, running MERT in systems where the user is waiting actively for the translation to be produced may not be acceptable.

However, it is quite common to have a great amount of data available in a given domain, but only a small amount of data available from the domain we are interested in translating. Precisely this scenario is appropriate for BA: under this paradigm, the weight vector $\boldsymbol{\Lambda}$ is *biased* towards the optimal one according to the adaptation set. However, over-training towards such set is avoided by not forgetting the generality provided by the training set.

4 Bayesian Adaptation for SMT

The main idea behind Bayesian learning is that parameters are viewed as random variables which have some kind of a priori distribution. In such case, observing these random variables leads to a posterior density, which typically peaks at the optimal values of these parameters. Following the notation in [1], the previous statement can be specified as

$$p(\boldsymbol{y}|\boldsymbol{x};T) = \int p(\boldsymbol{y},\theta|\boldsymbol{x};T)d\theta \tag{4}$$

where T represents the complete training set and θ are the model parameters.

Since in this case we are interested in Bayesian *adaptation*, we need to consider one training set T and one adaptation set A, leading to

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{x};T,A) &= \int p(\boldsymbol{y},\theta|\boldsymbol{x};T,A)d\theta \\ &= \int p(\theta|T,A)p(\boldsymbol{y}|\boldsymbol{x},\theta)d\theta \end{aligned} \tag{5}$$

In Equation 5, the integral over the complete parametric space forces the model to take into account all possible values of the model parameters, although the prior over the parameters implies that our model will prefer parameter values which are closer to our prior knowledge. Two assumptions have been made: first, that the output sentence $\boldsymbol{y}$ only depends on the model parameters (not on the complete training and adaptation data), and second, that model parameters do not depend on the actual input sentence $\boldsymbol{x}$. Such simplifications lead to a decomposition of the integral into two parts: the first one, $p(\theta|T,A)$ will assess how good the current model parameters are, and the second one, $p(\boldsymbol{y}|\boldsymbol{x},\theta)$, will account for the quality of the translation $\boldsymbol{y}$ given the current model parameters.

Operating with the probability of the model parameters, we obtain:

$$p(\theta|T,A) = \frac{p(A|\theta;T)\ p(\theta|T)}{\int p(A|\theta)\ p(\theta|T)\ d\theta} \tag{6}$$

$$p(A|\theta;T) = p(A|\theta) = \prod_{\forall a \in A} p(\boldsymbol{x}_a|\theta)\ p(\boldsymbol{y}_a|\boldsymbol{x}_a,\theta) \tag{7}$$

where the probability of the adaptation data has been assumed to be independent of the training data and has been modelled as the probability of each bilingual sample $(\boldsymbol{x}_a, \boldsymbol{y}_a) \in A$ being generated by our translation model.

Assuming that the model parameters follow a normal distribution, we obtain

$$p(\theta|T) = \frac{1}{(2\pi)^{-\sigma_T/2}|\sigma_T|^{-1/2}}\ \exp\left\{-\frac{1}{2}(\theta-\theta_T)^T\sigma_T^{-1}(\theta-\theta_T)\right\} \tag{8}$$

where θ_T is the set of parameters estimated on the training set and σ_T is the variance, which has been assumed to be bounded for all parameters.

Lastly, assuming that our translation model is a log-linear model (Equation 3) and that the only parameters we want to adapt are the log-linear weights:

$$p(\boldsymbol{y}|\boldsymbol{x},\theta) = \frac{\exp \sum_k \theta_k \; f_k(\boldsymbol{x},\boldsymbol{y})}{\sum_{\boldsymbol{y}'} \exp \sum_k \theta_k \; f_k(\boldsymbol{x},\boldsymbol{y}')} \tag{9}$$

Finally, combining Equations 7, 8 and 9, yields:

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{x};T,A) &= \int \frac{p(A|\theta;T)\; p(\theta|T)}{\int p(A|\theta)\; p(\theta|T)\; d\theta}\; p(\boldsymbol{y}|\boldsymbol{x},\theta)\; d\theta \\ &= \mathcal{Z} \int \prod_{\forall a \in A} p(\boldsymbol{x}_a|\theta)\; p(\boldsymbol{y}_a|\boldsymbol{x}_a,\theta)\; \mathcal{N}(\theta;\theta_T,\sigma_T) p(\boldsymbol{y}|\boldsymbol{x},\theta)\; d\theta \qquad (10) \\ &= \mathcal{Z}' \int \prod_{\forall a \in A} \frac{\exp \sum_k \theta_k \; f_k(\boldsymbol{x}_a,\boldsymbol{y}_a)}{\sum_{\boldsymbol{y}'} \exp \sum_k \theta_k \; f_k(\boldsymbol{x}_a,\boldsymbol{y}')} \qquad (11) \\ &\qquad \exp\left\{-\frac{1}{2}(\theta-\theta_T)^T \sigma_T^{-1} (\theta-\theta_T)\right\} \frac{\exp \sum_k \theta_k \; f_k(\boldsymbol{x},\boldsymbol{y})}{\sum_{\boldsymbol{y}'} \exp \sum_k \theta_k \; f_k(\boldsymbol{x},\boldsymbol{y}')}\; d\theta \end{aligned}$$

where, in Equation 10, $\mathcal{Z}$ is the denominator present in the previous equation and may be out-factored because it does not depend on the integration variable. In Equation 11, it has been assumed that the probability of the input sentence does not depend on the model parameters, and hence it can also be out-factored.

5 Experiments

In this section we will detail the experiments carried out. We will first train a SMT system on training data, and then we will analyse the performance of such system when used for translating data which does not belong to the same domain as the training data. We will follow two adaptation procedures. On the one hand, log-linear model weights are estimated on the adaptation data, forgetting about the estimates obtained in training time. On the other hand, we will perform experiments with our BA technique, and finally compare both approaches.

5.1 Experimental Setup

In this work, we will be assessing translation quality by means of two standard scoring metrics in SMT, namely BLEU and TER scores. BLEU measures the precision of n-grams [18] with a penalty for too short sentences, whereas TER [19] is an error metric that computes the minimum number of edits required to modify the system hypotheses so that they match the references. Possible edits include insertion, deletion, substitution of single words and shifts of word sequences.

To train the baseline system, we used the Europarl corpus [20], with the partition established for the Workshop of SMT of NAACL 2006 [21]. Specifically, we performed experiments on Spanish–English translation. The corpus Europarl corpus is divided into three separate sets: one for training, one for development and one for test. The figures of the Europarl corpus are shown in Table 1.

Table 1. Main figures of the Europarl corpus. *OoV* stands for Out of Vocabulary.

		Spanish	English
Training	Sentences	731K	
	Run. words	15.7M	15.2M
	Vocabulary	103K	64K
Development	Sentences	2000	
	Run. words	61K	59K
	OoV words	208	127
Test	Sentences	2000	
	Run. words	60K	58K
	OOV words	207	125

Table 2. Main figures of the Xerox and EU corpora. *OoV* stands for Out of Vocabulary.

		Xerox		EU	
		Spanish	English	Spanish	English
Training	Sentences	55K		164K	
	Run. words	712K	631K	3.4M	3.1M
	Vocabulary	11K	8K	45M	34M
Test	Sentences	1120		800	
	Run. words	10K	8K	23K	20K
	OoV words	42	27	97	81
	OoV w.r.t. Europarl	131	139	156	178
	ppl w.r.t. Europarl	2555	9595	130	194

Since we will be performing adaptation, we also used two other corpora, namely the Xerox corpus [22] and the *EU* corpus [23]. The Xerox corpus is a compendium of user manuals for Xerox printers and photocopiers and was translated from English into other languages by Xerox's language services. The EU corpus was built from the Bulletin of the European Union and is publicly available on the Internet. In this paper, we will focus on the Spanish–English sub-corpora. These two corpora are divided into two separate subsets, one for training and one for test. Their characteristics can be seen in Table 2. It must be noted that EU and Europarl corpora belong to very similar domains, whereas Xerox belongs to a very different domain. This fact is the reason why the Xerox corpus reports such high perplexity (ppl) rates with respect to a language model estimated on the Europarl corpus. Intuitively, perplexity measures how "surprised" the language model is when provided a given test set, i.e. how different such set is with respect to the data it was trained on.

We conducted our experiments by means of the Moses toolkit [24], which implements a statistical log-linear model including five translation scores, a language model, a distortion model, and word and phrase penalties. The five translation scores included provide standard direct and inverted frequency-based and lexical-based probabilities for each phrase pair in the phrase-table.

The initial weights for the log-linear model were estimated by means of MERT on the Europarl development set, as is typically done in SMT. The score to be optimised in this case was BLEU.

5.2 Practical Approximations

In order to find the best scoring sentence according to Equation 11, we asked the decoder to output a list of 500-best for each one of the translated sentences. Such n-best list was then re-ranked according to the score provided by Equation 11 after dropping the normalisation factor Z', since such factor is constant when choosing the maximum scoring output sentence.

Since a true integral over all possible weights is not feasible for computational reasons, we discretised the integral to consider only a set of sample weight vectors. Here, such sampling was performed by taking into account the weights considered by MERT for the in-domain development set. The idea behind such sampling is to perform a Monte Carlo-like sampling of the model parameters.

A last consideration when attempting to implement the Equation 11 is that the first part of the integral, the product over all samples in the adaptation set, cannot be computed with typical state-of-the-art phrase-based SMT systems, since e.g. out-of-vocabulary words may imply that the SMT model is unable to explain a certain bilingual sentence completely. Hence, instead of computing

$$\prod_{\forall a \in A} \frac{\exp \sum_k \theta_k \; f_k(\boldsymbol{x}_a, \boldsymbol{y}_a)}{\sum_{\boldsymbol{y}'} \exp \sum_k \theta_k \; f_k(\boldsymbol{x}_a, \boldsymbol{y}')} \tag{12}$$

we will need to compute

$$\prod_{\forall a \in A} \frac{\exp \sum_k \theta_k \; f_k(\boldsymbol{x}_a, \boldsymbol{y}^*_a)}{\sum_{\boldsymbol{y}'} \exp \sum_k \theta_k \; f_k(\boldsymbol{x}_a, \boldsymbol{y}')} \tag{13}$$

where $\boldsymbol{y}^*$ represents the best hypothesis the search algorithm is able to produce, according to a given translation quality measure. Since BLEU is not well defined at the level of sentence because it implements a geometrical average which can be zero, we will be using TER for this purpose.

5.3 Experimental Results

We conducted adaptation experiments by using the SMT system trained on Europarl as a baseline system and translated the Xerox and EU test sets. Then, increasing the number of adaptation samples made available to the system was considered, starting from 10 up to 140. These adaptation samples were drawn from the respective training corpus, i.e. when translating the Xerox test set, the adaptation samples were drawn from the Xerox training corpus. In order to provide robustness to the results presented here, 15 random samplings for each size of the adaptation subset were drawn. These adaptation data were used either for weight estimation via MERT, or as adaptation set for our BA technique. Results can be seen in Figures 1 and 2. It is important to remember that the higher the BLEU score the better, as opposed to TER, where lower scores imply better translation quality.

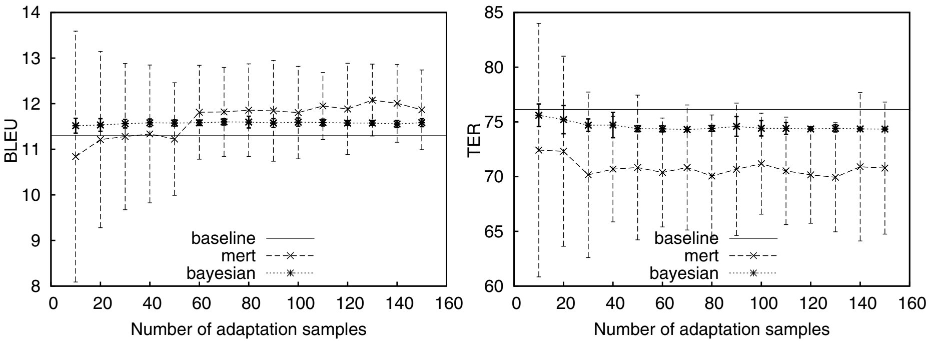


Fig. 1. Performance of baseline and both adaptation techniques when increasing the number of adaptation samples. Translation quality is measured with BLEU and TER for the Xerox test data. 95% confidence intervals are shown.

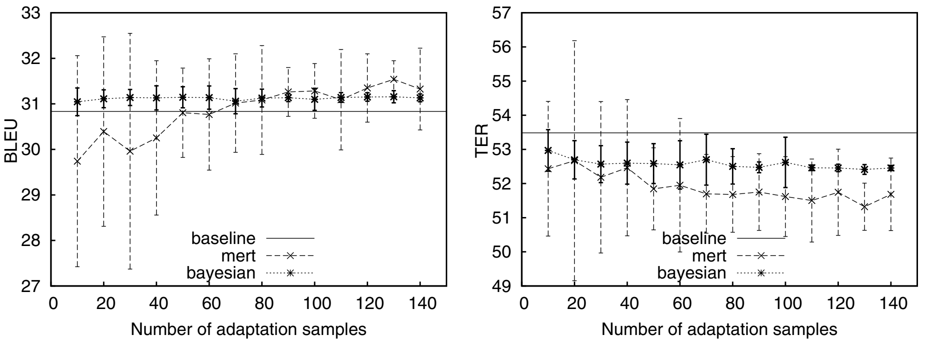


Fig. 2. Performance of baseline and both adaptation techniques when increasing the number of adaptation samples. Translation quality is measured with BLEU and TER for the EU test data. 95% confidence intervals are shown.

As the figures show, the translation quality produced by the system with $\boldsymbol{\Lambda}$ adjusted by means of MERT turns very unstable, and the confidence intervals get very big. In average, such system is able to improve the baseline, but at the risk of producing very bad quality translations. This is not an acceptable behaviour for a system that is set on-line for translating. Furthermore, the computational cost of running the MERT algorithm, even for small amounts of adaptation data, is prohibitive whenever the system is required to produce translations in real-time environments, in which the user awaits for a translation to be produced almost immediately. In contrast, the BA technique is able to yield improvements over the baseline translation quality even when very small amounts of adaptation data are available, with a much more predictable behaviour: while the confidence intervals have a range of about 7 points for BLEU and even 23 points for TER, BA is able to reduce the intervals to less than a single point in almost every case. Although estimating $\boldsymbol{\Lambda}$ only on the adaptation set seems to perform *on average* better than

BA, this comes at the risk of producing much worse translations. Moreover, the formula described in 11 can easily be incorporated into the decoder, without any significant increase in computational complexity.

6 Conclusions and Future Work

The results presented in the previous section show that the BA technique implemented is able to provide consistent improvements over the baseline, although these are not very big, even when a very small amount of adaptation data is available. Precisely in this scenario is in which a true adaptation technique is to be applied: if enough adaptation data is available, then the best "adapted" system is the system trained only on the adaptation data. Hence, when the amount of adaptation data available increases, MERT is able to yield better results. However, it must also be noted that MERT heavily depends on the data provided, as the confidence intervals show, and this can lead to unexpectedly high or low translation quality without being able to know the behaviour in advance.

Nevertheless, there are several details that must still be taken care of, and that we plan to address in future work. First, if we look at Equation 11, it seems very obvious that the first and the second component of the integral, i.e. the probability of the adaptation data and the prior over the model parameters, are clearly in very different numeric ranges. This has as effect that the probability of the adaptation sample may have less discriminative power than the prior, and this, in turn, may be the reason why the results presented are so stable, but do not yield very big improvements. We plan to address this in future work by introducing weighting coefficients to compensate for this. Such coefficients might need to be trained, but most likely only once, independently of the corpus used.

The way in which the weight sampling is done is also bound to have an important impact on the final results. We also plan to address it in future work.

The derivation presented here can be quite easily extended in order to adapt the feature functions of the log-linear model (i.e. not the weights). This is bound to have a more important impact on the quality of the translations produced, since the amount of parameters to be adapted is much higher.

Acknowledgements

Work partially supported by the EC (FEDER/FSE), the Spanish Government (MICINN, MITyC, "Plan E") under MIPRCV "Consolider Ingenio 2010" (CSD2007-00018), erudito.com (TSI-020110-2009-439) and iTrans2 (TIN2009-14511), and by the Generalitat Valenciana (Prometeo/2009/014). The authors would like to thank the anonimous reviewers for their constructive comments.

References

1. Brown, P., Pietra, S.D., Pietra, V.D., Mercer, R.: The mathematics of machine translation. In: Computational Linguistics, vol. 19, pp. 263–311 (June 1993)
2. Papineni, K., Roukos, S., Ward, T.: Maximum likelihood and discriminative training of direct translation models. In: Proc. of ICASSP, pp. 189–192 (1998)

3. Och, F., Ney, H.: Discriminative training and maximum entropy models for statistical machine translation. In: Proc. of the ACL 2002, pp. 295–302 (2002)
4. Heigold, G., Rybach, D., Schlüter, R., Ney, H.: Investigations on convex optimization using log-linear hmms for digit string recognition. In: Interspeech, Brighton, U.K., September 2009, pp. 216–219 (2009)
5. Tahir, M.A., Heigold, G., Plahl, C., Schlueter, R., Ney, H.: Log-linear framework for linear feature transformations in speech recognition. In: IEEE Automatic Speech Recognition and Understanding Workshop, Merano, Italy (December 2009)
6. Lagarda, A., Juan, A.: Topic detection and classification techniques. In: WP4 deliverable, TransType2 (2003)
7. Nepveu, L., Lapalme, G., Langlais, P., Foster, G.: Adaptive language and translation models for interactive machine translation. In: Proc. of EMNLP (2004)
8. Kuhn, R., Mori, R.D.: A cache-based natural language model for speech recognition. IEEE Transactions on PAMI 12(6), 570–583 (1990)
9. Koehn, P., Schroeder, J.: Experiments in domain adaptation for statistical machine translation. In: Proc. of ACL WMT (2007)
10. Bertoldi, N., Federico, M.: Domain adaptation in statistical machine translation with monolingual resources. In: Proc. of EACL WMT (2009)
11. Civera, J., Juan, A.: Domain adaptation in statistical machine translation with mixture modelling. In: Proc. of ACL WMT (2007)
12. Zhao, B., Eck, M., Vogel, S.: Language model adaptation for statistical machine translation with structured query models. In: Proc. of CoLing (2004)
13. Sanchis-Trilles, G., Cettolo, M., Bertoldi, N., Federico, M.: Online Language Model Adaptation for Spoken Dialog Translation. In: Proc. of IWSLT, Tokyo (2009)
14. Zhang, H., Quirk, C., Moore, R.C., Gildea, D.: Bayesian learning of non-compositional phrases with synchronous parsing. In: Proceedings of ACL 2008: HLT. Association for Computational Linguistics, June 2008, pp. 97–105 (2008)
15. Zens, R., Och, F., Ney, H.: Phrase-based statistical machine translation. In: Jarke, M., Koehler, J., Lakemeyer, G. (eds.) KI 2002. LNCS (LNAI), vol. 2479, pp. 18–32. Springer, Heidelberg (2002)
16. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Proc. HLT/NAACL 2003, pp. 48–54 (2003)
17. Och, F.: Minimum error rate training for statistical machine translation. In: Proc. of Annual Meeting of the ACL (July 2003)
18. Papineni, K., Kishore, A., Roukos, S., Ward, T., Zhu, W.J.: Bleu: A method for automatic evaluation of machine translation. In: Technical Report RC22176, W0109-022 (2001)
19. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proc. of AMTA 2006 (2006)
20. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: MT Summit (2005)
21. Koehn, P., Monz, C. (eds.): Proc. on the Workshop on SMT. Association for Computational Linguistics (June 2006)
22. Esteban, J., Lorenzo, J., Valderrábanos, A., Lapalme, G.: Transtype2 - an innovative computer-assisted translation system. In: Proc. of 42nd ACL, Barcelona, Spain, July 2004, pp. 94–97 (2004)
23. Khadivi, S., Goute, C.: Tools for corpus alignment and evaluation of the alignments (deliverable d4.9). In: Technical Report, TransType2, IST-2001-32091 (2003)
24. Koehn, P., et al.: Moses: Open source toolkit for statistical machine translation. In: Proc. of ACL Demo and Poster Sessions, Czech Republic, Prague, pp. 177–180 (2007)

A Generative Score Space for Statistical Dialog Characterization in Social Signalling

Anna Pesarin[1], Marco Cristani[1,2], Paolo Calanca[1], and Vittorio Murino[1,2]

[1] Dipartimento di Informatica, University of Verona, Italy
[2] Istituto Italiano di Tecnologia (IIT), Genova, Italy

Abstract. The analysis of human conversations under a social signalling perspective recently raised the joint attention of pattern recognition and psychology researchers. In particular, the dialog classification represents an appealing recent application whose aim is to go beyond the meaning of the spoken words, focusing instead on the way the sentences are pronounced by capturing natural (or hidden) characteristics, such the mood of the conversation. An effective strategy to face this issue is to encode the turn-taking dynamics in a generative model, whose structure is composed by conditional dependencies among first-order Markov processes. In this paper, we follow this strategy, investigating how to boost the classification performances of this model and of the related higher-order Markov extensions, through the definition of a novel generative score space. Generative score spaces are employed to increase generative classification in a discriminative way, also allowing a deep understanding of the processed data through the use of standard pattern recognition strategies. Experiments on real data certify the goodness of our intuition.

Keywords: social signalling, dialogue analysis, observed influence model.

1 Introduction

Social signal processing (SSP) aims at developing theories and algorithms that codify how human beings behave while involved in social interactions, putting together perspectives from sociology, psychology and computer science [1,2]. Here, the main entities to analyse are the social signals [2], i.e., temporal co-occurences of social cues [3], that can be basically defined as a set of temporally sequenced changes in neuromuscular, neurocognitive and neurophysiological activity. Social cues are organized into five categories that are heterogeneous, multimodal aspects of a social interplay [2]: 1) physical appearance, 2) gesture and posture, 3) face and eyes behavior, 4) vocal behavior, and 5) space and environment.

The analysis of the social cues in the vocal behavior category is one of the issues most related to pattern recognition and machine learning themes. In general, this analysis consists in evaluating all the spoken cues that surround the verbal message and influence its actual meaning, characterizing, for example, particular social roles (*e.g.*, *dominance*, [4,5,6], *mirroring*, [7] and others [8]). A more recent challenge is to consider a conversation in its entirety, as a sample in a multidimensional space, in order to perform un/supervised clustering, indexing, retrieval and

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 630–639, 2010.

other novel applications. For example, in [9], the goal was to predict the outcome of a specific conversational exchange finding and exploiting short meaningful portions of interaction. Another issue is that of the dialog classification (or chracterization), aimed at capturing general aspects or characteristics of a dialog. For example, in [10], the main "atmosphere"(e.g., flat, aggressive, etc.) for a indefinitely long piece of dialogic exchange was classified, together with the capability of recognizing the presence in the dialog of a particular class of speakers, such as adults or children. It is worth noting that all these tasks usually do not involve speech recognition: actually, it happens very often that the meaning of the spoken sentences and the vocal behavior of a subject are completely in discordance.

In all the approaches above, generative models seem to be the main technique for exploiting vocal behavior cues for social signalling. Usually, they are employed for the analysis of the turn taking, *i.e.*, the sequence of turns in which a dialog participant can be in one of two states: silent or talking. Turn-taking dynamics may be effectively modelled as conditional dependencies among stochastic processes, where each process models the behavior of a single speaker. In particular, dynamic Bayesian networks were employed as efficient and expressive tools, especially, hidden Markov models (HMMs) and extensions [11], and influence model and extensions [12,7] (a.k.a. mixed memory Markov processes [13]). The common idea is to sample a dialog at fixed time intervals, to learn a representative model, and infer over the model parameters for detecting social aspects of that dialog. In [11], a two-layer HMM was employed to model individual and group action. In [12,7], the purpose was to detect the dominant interlocutor through social cues of mimicking. The authors employ an Observed Influence Model (OIM), *i.e.*, an aggregate of first-order Markov processes, each one addressing an interlocutor. OIM's main feature is the capability of translating complex conditional dependencies among random variables with pairwise dependencies by means of weights called influence factors. Recently, in [10] a generative framework has been proposed, aimed at classifying a piece of conversation of variable length (from few minutes to hour), considering the nature of the people involved within (children, adults) and the main mood (flat, arguing). The framework is basically an OIM, fed by low-level auditory social signals, dubbed steady conversational periods (SCPs). They are built on duration of continuous slots of silence or speech, and, in addition, they take into account conversational turn-taking. In practice, SCPs allow to capture the attitude of self selecting for turn-taking even though the interlocutor has not yet completed his own turn. Further, they also indirectly model speech planning by characterizing the tendency to utter short sentences instead of longer propositions. We name here this generative framework as SCP model for brevity.

Employing generative machines for modelling dialog data is advantageous: the parameters of the model are intelligible, permitting inferences that highlight intuitively the nature of the data. For example, in [10], the coefficients of the transition matrices of the Markov model utilised suggest the dynamics of a process in a straightforward way, by identifying, for instance, highly probable or rare transitions.

In this paper, we focus on the dialog classification, considering the SCP model and proposing a principled way to boost its classification performances, also permitting, at the same time, a more informative analysis of the model.

To this end, we exploit a set of SCP models in an unconventional way, i.e., not as classifiers, but as feature ensembles: the idea is to build a novel generative score space [14], where the parameters of the SCP models are treated as features. More in detail, each sample (a dialog) of a given dialog class is used to learn an SCP model, and the model parameters can be considered as features in a joint multidimensional space. This embedding is repeated for all the classes of dialogs considered. Then, classical feature selection and ranking strategies are carried out, individuating the more discriminant features (i.e., parameters) for distinguishing the different dialog classes. Finally, discriminative classifiers are employed to perform the classification.

The concept of generative score space has recently raised the attention of the researchers, being a principled way to boost the classification performances of generative classifiers. In the majority of the approaches, discriminative techniques, like Support Vector Machines (SVMs), are fed with the (generative) features derived from the learned generative models, providing state-of-the-art performances [14,15,16,17]. The problem of such a formulation is that the discriminative part of the system hides one of the advantages brought from the generative modeling, i.e., we lose the intelligibility of the extracted features and, then, of the entire process.

In our framework, pre-processing feature selection and ranking strategies are carried out on the generative score space, allowing a full control and understanding of the collected features. In particular, we can observe what are the most useful features for classification, i.e., the most important parameters of the model that, in turn, means to highlight which transitions are more characteristic for a certain dialog class. In addition, we augment the complexity of the original generative framework, by embedding higher-order OIMs in the SCP model (i.e., considering Markov processes of higher order). This because another nice feature of the generative score space based approaches for classification is their ability to deal with overfitted models or with a small training sample size.

Summarizing, in this paper we reach three goals for dialog characterization: first, we perform classification in a very effective way; second, we employ more structured models for the dialog analysis, investigating whether higher order can encode a finer characterization of the turn taking; third, we understand the most important differences among different SCP models due to the embedding in the generative score space, realizing in a very intuitive way what are the behavioral patterns that characterize the different classes of dialogs.

The rest of the paper is organized as follows: in Sec. 2, mathematical recaps are given together with a brief description of the SCP model. Sec. 3 details our generative score space, and Sec. 4 reports the experimental results on a public dataset. Finally, Sec. 5 concludes the paper, summarising final observations on the turn-taking dynamics modelling and future perspectives of the work.

2 Mathematical Background

2.1 The Observed Influence Model

The observed influence model (OIM) is a simplified version of the influence model [12], that in practice operates on Markov processes instead of hidden Markov processes. Inheriting the notation of [10], the state variable of a Markov process is $S_t \in \{1, \ldots, N\}$, and $P(S_t|S_{t-1},..., S_{t-k})$ is the transition probability for a Markov model of order k. OIM factorizes the multi-process conditional relations among C Markov chains by means of a weighted linear combination of pairwise *inter-chain* and *intra-chain* transition probabilities. Considering first-order Markov chains with N states, the (full) factorization of the multi-process transition probability is

$$P({}^cS_t|{}^1S_{t-1}, \ldots, {}^CS_{t-1}) = \sum_{d=1}^{C} {}^{(c,d)}\theta P({}^cS_t|{}^dS_{t-1}) \tag{1}$$

with $1 \leq c, d \leq C$, ${}^{(c,d)}\theta \geq 0$, $\sum_{d=1}^{C} {}^{(c,d)}\theta = 1$. The value $P({}^cS_t|{}^dS_{t-1})$ represents the probability of going from state S_{t-1} of the chain d to state S_t of the chain c. The weight ${}^{(c,d)}\theta$ represents the influence that chain d exerts on chain c. A sketch of the model is depicted in Fig.1 a. A first-order influence model is thus defined as $\lambda = \{\{A^{(c,d)}\}, \Theta, \pi\}$, where $A^{(c,d)}$ is the *intra*-chain transition matrix when $c = d$, and represents the dynamics of a single process *per se*. When $c \neq d$, we consider the *inter*-chain matrices, modeling how much a state of a chain conditions the next state of the other chain. The $C \times C$ matrix Θ contains the influence weights, and π contains the (independent) initial probability distributions for all processes, *i.e.*, $\pi^{(c)} = \{\pi_i^{(c)}\}$ where $\pi_i^{(c)} = P(S_1^{(c)} = i)$.

The OIM transition factorization has space complexity $O(C^2N^2 + C^2)$, where C^2N^2 is due to the transition tables parameters, and C^2 to the influence coefficients.OIM learning of the $\{\theta\}$ coefficients is performed by standard constrained gradient descent [7,18], while the $\{\{A^{(c,d)}\}, \pi\}$ parameters are estimated by simple state counting.

2.2 The SCP Model

In their original framework [10], the authors focused on two-person conversations, where subjects 1 and 2 were captured in an appropriate environment, obtaining two synchronized separated audio sources. From the raw signals, a speech/silence thresholding was performed, obtaining a signal D, formed by two binary arrays $D^{(1)}$ and $D^{(2)}$, of length T.

Under this setting, a dialogue can be represented by an OIM, but the lack of synchronization between the start/end instants of the periods leads to problems in evaluating inter-chain conditional dependencies (see Fig.1 b).

Therefore, the authors propose the use of a turn taking-based feature, called *steady conversational period* (SCP), that is built on duration of continuous slots

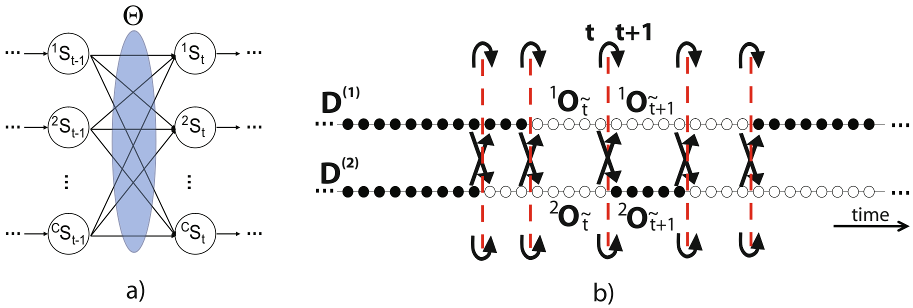


Fig. 1. The generative framework: a) State factorization exploited in an Observed Influence Model. The area named Θ indicates the influence factors that apply to the state transitions, depicted as directed arrows. b) Synchronization through Steady Conversational Periods. We have two audio processes, $D^{(1)}$ and $D^{(2)}$, sampled at a given frequency, where audio samples are shown as *speech* (black dots) and *silence* (white dots) values. Continuous periods of speech or silence are not synchronized, so it is not possible to evaluate a first-order statistical transition probability among the periods. The global transitions (dashed red lines) define the SCPs so permitting to calculate first-order transition probabilities (black arrows).

of silence or speech. The extraction process of the SCPs assumes that whenever a process changes its state, it causes a *global* transition that affects also the opposite process, inserting a novel auto-transition state (see Fig. 1 b). The fragmentation caused by global transitions synchronizes the processes, creating $\tilde{T} < T$ different SCPs $^cO_{\tilde{t}}$, where the apex $c \in 1, 2$ indexes the speaker and $\tilde{t} = 1, \ldots, \tilde{T}$ enumerates the different SCPs.

The introduction of SCPs in the model makes it feasible to evaluate first-order intra- and inter-chain conditional probabilities (red dashed line in Fig. 1 b). In order to take into account the different durations of each silence and speech segment, all SCPs related to the speech and to the silence are labelled into $\langle short, long \rangle$, after a Gaussian clustering over a training dataset.

More formally, given the clustering, each SCP $^cO_{\tilde{t}}$ takes one label among $1, 2, 3, 4$, where 1,2 address short and long continuous periods of speech, respectively, and the same applies with 3,4 for the silence periods.

After that, an observed influence model $\lambda = \{\{A^{(c,d)}\}, \Theta, \pi\}$ is fitted to the SCP labels.

The parameters of the model intuitively indicate the conversational trend of each subject considered separately. The inter-chain transition parameters encode first-order state dependencies among processes, and influence factors mirror the influence that a process exerts on the other.

3 The Generative Score Space

In order to increase the classification accuracy of the generative framework, and, at the same time, to allow a discriminative analysis of the model parameters, we build a generative score space $\mathcal{I}$. Following [19], such spaces can be built from the data available by considering each observed SCP sequence composed by the two synchronized SCP streams coming from the process 1 and 2 $O = (O_1, \ldots, O_{\tilde{t}}, \ldots, O_{\tilde{T}})$, and a family of generative models $\mathcal{P} = \{P\left(O|\Psi_i\right)\}$ parameterized by Ψ_i.

The observed dialog O is mapped into a fixed-length score vector $\varphi^f_{\hat{F}}(O)$,

$$\varphi^f_{\hat{F}}(O) = \varphi_{\hat{F}} f(\{P_i\left(O|\Psi_i\right))\}), \tag{2}$$

where f is a function of the set of probability densities under the different models, and $\hat{F}$ is some operator applied to it. For instance, in case of the Fisher score [14], f is the log likelihood, and the operator $\hat{F}$ produces the first-order derivatives with respect to the parameters. Another example is the TOP kernel [15] for which the function f is the posterior log-odds and $\hat{F}$ is still the gradient operator.

In these cases, the generative score-space approaches help to distill the relationship between a model parameter θ_i and the particular data sample. After the mapping, a score-space metric must be defined in order to employ discriminative approaches.

In our case, f is the parameter extractor function (i.e., the function that estimates the parameters of a statistical distribution), and F is the identity operator. In practice, f extracts the transition parameters (by simple counting) and the influence coefficients (by gradient descent).

Given a set of M classes of dialogs, each formed by W sequences, the space $\mathcal{I}$, equipped with the traditional norm and Euclidean metric, could be seen thus formed by a set of multidimensional class-labeled samples; actually, on each sequence, a model is trained, that furnishes a set of features/parameters. Therefore, standard tools of data analysis can be applied. In our case, we want to highlight the discriminative power of the features in a classification context, and therefore we apply a feature selection (or ranking) strategy, and, subsequently, we apply different discriminative classifiers on the features subset. The feature selection/ranking strategies together with the discriminative classifiers employed will be detailed in the next section. Discriminative classifiers are preferred here, because they directly focus on estimating class posterior probabilities instead of modeling class distributions. Such classifiers should also be less affected by the curse of dimensionality problem.

In order to stress this aspect, and to assess how strong the improvement in the classification can be, when dealing with more structured models, we augment the order of the OIM embedded in the SCP-based generative framework. This is based on the following factorization:

$$P({}^cS_t|{}^1S_{t-1},\ldots,{}^CS_{t-1},\ldots,{}^1S_{t-k},\ldots,{}^CS_{t-k}) = \sum_{d=1}^{C}{}^{(c,d)}\theta P({}^cS_t|{}^dS_{t-1},\ldots,{}^dS_{t-k}) \tag{3}$$

Encapsulating higher-order OIMs in the SCP-based generative framework is straightforward. The embedding in $\mathcal{I}$ leads to having the ensemble of features $\varphi^f_{\hat{F}}(x) = \{\{A^{(c,d)}\}, \Theta, \pi\}$, for each model (note that in this case, $A^{(c,d)}$ contains

N^{k+1} values). Considering in particular the number of parameters, we have $C^2N^{k+1} + C^2$. For example, fixing $N = 4$ and $C = 2$ brings to 132 elements in the case of second order OIMs.

The rationale under the choice of this score space is that, employing parameters as discriminative features, we can understand what portions of a model differs from the other models at hand. For example, capturing the fact that a particular state transition is strongly discriminant for a certain class, means that such transition is peculiar for that model. This property cannot be mimicked by Fisher score based approaches, where the basic tool is the differentiation with respect to particular quantities (i.e., the log-likelihood in the Fisher score), that can suffer of the so-called "wrap-around" problem, where very different data points may map to the same derivative (see [17] for an example).

4 Experiments

In the experimental section, we employ the same database used for [10][1], adopting the results reported in their paper as comparison. The code was written in MATLAB, and the classifier adopted, together with the feature selection strategies considered, were instantiated employing the PRTOOLS [20].

The corpus contains 41 dialogic conversations played by 30 subjects that can be grouped by age and mood in order to recognize three dialogue classes:

C1: 13 flat semi-structured plus 5 flat unstructured dialogues between two adults ranging from 22 to 40 years.

C2: 14 flat semi-structured dialogues between a child, ranging from 4 to 6 years, and an adult.

C3: 9 arguing unstructured dialogues between two adults, ranging from 22 to 40 years

Each sample is approximately 10 min. long. In semi-structured conversation the moderator, a research-trained female psychologist who did not know the aim of the experiment, introduced in sequence 5 predetermined topics with fixed questions in a given order (school time, hobbies, friends, food, family). The class C3 was extracted by a corpus of phone office conversations driven by an operator who was aware of the experimental goal, and other subjects (Computer Science department employees) which were only warned about the possibility that an arguing issue might arise.

The classification task is performed into four different scenarios:

(A) flat *vs* dispute - ($cat.1$ *vs* $cat.3$);
(B) flat *vs* dispute, *general* - (($cat.1 \cup cat.2$) *vs* $cat.3$);
(C) with *vs* without child - ($cat.2$ *vs* $cat.1$);
(D) all *vs* all;

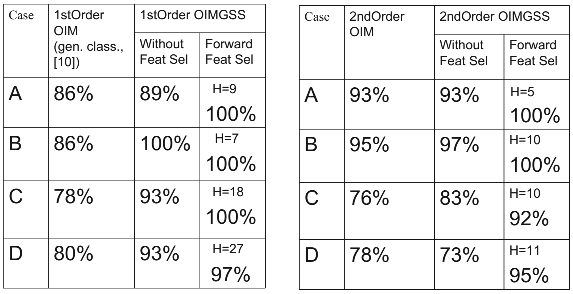
For the sake of clarity, let us suppose of having for each class L samples. In [10], the classification was performed in a generative way, using a Maximum Likelihood scenario, and cross-validating via leave-one-out (LOO). In other words, a

[1] The database is downloadable by contacting the authors.

class model was learned with L-1 samples, and testing was performed considering the last sample. In our case, we also employ LOO cross-validation, but we learn L-1 models for each class, one for each sample, projecting their scores into the generative score space, using the last sample as test. After that, we follow two different directions. First, we perform classification adopting all the features for each model, selecting different classifiers:

- *kernelc* [18]: a classifier based on a kernel or dissimilarity representation defined by Fisher approach;
- *knnc* [21]: a classifier based on k-nearest neighbor rule;
- *parzenc* [22]: a parzen classifier, using the best smoothing parameter of the kernel;

The best performances were reached by the *kernelc* classifier, and we report only these results in Fig.2 for brevity.



Case	1stOrder OIM (gen. class., [10])	1stOrder OIMGSS	
		Without Feat Sel	Forward Feat Sel
A	86%	89%	H=9 100%
B	86%	100%	H=7 100%
C	78%	93%	H=18 100%
D	80%	93%	H=27 97%

Case	2ndOrder OIM	2ndOrder OIMGSS	
		Without Feat Sel	Forward Feat Sel
A	93%	93%	H=5 100%
B	95%	97%	H=10 100%
C	76%	83%	H=10 92%
D	78%	73%	H=11 95%

Fig. 2. Classification results. In each table, the second column reports the results obtained by the generative approach, the third column shows the use of the *kernelc* classifier on all the features in the score space, the fourth column reports the results obtained after feature selection (H is the number of features considered).

Through the generative score space embedding, the classification performances augmented, except for the scenario D (all vs all) in the second order case. Investigating the feature space, we found that several parameters where shared among classes. Therefore, we employ forward feature selection (*ffs*) based on the 1-nearest-neighbor classification criterion. In this case, the capability of our score space to explain the data is evident. In all the cases, the generative performances were outperformed. Please note that each scenario required a different number of features for reaching the best performance. In particular, the scenario A and B were the simplest, and required a small set of features. Scenario C and D were more difficult, and a bigger number of feature were evaluated. It is worth noting how the second order model produced both a pure generative modeling and a score space that are less informative than those of order 1 (this considering the scenario C and D, which are more challenging wrt the first two). In this case, we can assume that the first order reasoning works better in this kind of scenario.

In order to understand the importance of all the selected features, we rank them, employing the *featrank* 1-nearest-neighbor feature ranking strategy, which evaluates the performances of each single feature taken separately. In this way, we highlight the parameters more discriminative for classification. For example, in the scenario A, in the first-order case, the transition probability between long speech state (SCP value = 4) of speaker 1 and long silence state of speaker 2 (SCP value = 2) was present as important feature (rank 1). This information serves to address a quantitative analysis about the nature of the models learned. In specific, the value of the above transition probability was high for the flat conversations (0.8), very low (0.1) for the arguing discussion. This mirrors the fact that a turn taking dynamics of a calm dialog implies that, especially after a long period of speech of a speaker, the other takes a while for thinking and elaborating its turn. In the arguing conversation, this dynamics is not present. Another important feature/parameter we found (rank 2) is that of a short speech of a subject (SCP value = 3) after a short speech of the other person. This transition has high probability for the arguing conversation, low for the flat, and witnesses the fact that in a fight, periods are usually shorter, and the speakers talk on each other.

5 Conclusions

In this paper, we propose a novel generative score space that operates directly on the parameters of a generative model, for increasing the classification performance on a social signal application. The peculiarity of our approach is to extract directly the parameters of the model, instead of relying on differentiating over the log-likelihood. This allows to highlight better the importance of the parameters, by means of feature selection/ranking strategies. This leads to higher classification performance thanks to discriminative reasoning on the selected features, and to understand better the data modeled. The future perspective is to better characterize theoretically such space, reasoning on expected classification bounds that can be achieved with it.

References

1. Pantic, M., Pentland, A., Nijholt, A.: Special issue on human computing. IEEE Trans. on Systems, Man, and Cybernetics, Part B 39(1) (2009)
2. Vinciarelli, A., Pantic, M., Bourlard, H.: Social signal processing: Survey of an emerging domain. Image and Vision Computing 27(12), 1743–1759 (2009)
3. Ambady, N., Rosenthal, R.: Thin slices of expressive behavior as predictors of interpersonal consequences: A meta-analysis. Psychological Bulletin 11(2), 256–274 (1992)
4. Jayagopi, D., Hung, H., Yeo, C., Gatica-Perez, D.: Modeling dominance in group conversations using nonverbal activity cues. Trans. Audio, Speech and Lang. Proc. 17(3), 501–513 (2009)
5. Choudhury, T., Basu, S.: Modeling conversational dynamics as a mixed-memory markov process. In: Saul, L.K., Weiss, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems 17, pp. 281–288. MIT Press, Cambridge (2005)

6. Vinciarelli, A.: Speakers role recognition in multiparty audio recordings using social network analysis and duration distribution modeling. IEEE Transactions on Multimedia 9(6), 1215–1226 (2007)
7. Basu, S., Choudhury, T., Clarkson, B., Pentland, A.: Learning human interaction with the influence model. MIT MediaLab, Tech. Rep. 539 (2001)
8. Pentland, A.: Social signal processing. IEEE Signal Processing Magazine 24(4), 108–111 (2007)
9. Curhan, J., Pentland, A.: Thin slices of negotiation: Predicting outcomes from conversational dynamics within the first five minutes. Journal of Applied Psychology 92, 802–811 (2007)
10. Cristani, M., Pesarin, A., Drioli, C., Perina, A., Tavano, A., Murino, V.: Auditory dialog analysis and understanding by generative modelling of interactional dynamics. In: Second IEEE Workshop on CVPR4HB, Miami, Florida (2009)
11. Zhang, D., Gatica-Perez, D., Bengio, S., McCowan, I., Lathoud, G.: Modeling individual and group actions in meetings with layered hmms. IEEE Transactions on Multimedia (May 2005)
12. Asavathiratham, C.: A tractable representation for the dynamics of networked markov chain. Ph.D. dissertation, Dept. of ECS, MIT (2000)
13. Saul, L., Jordan, M.: Mixed memory markov models: Decomposing complex stochastic processes as mixtures of simpler ones. Machine Learning 37(1), 75–87 (1999)
14. Jaakkola, T., Haussler, D.: Exploiting generative models in discriminative classifiers. In: Proceedings of the 1998 conference on Advances in neural information processing systems II, pp. 487–493. MIT Press, Cambridge (1999)
15. Tsuda, K., Kawanabe, M., Rätsch, G., Sonnenburg, S., Müller, K.: A new discriminative kernel from probabilistic models. Neural Comput. 14(10), 2397–2414 (2002)
16. Bicego, M., Pekalska, E., Tax, D., Duin, R.: Component-based discriminative classification for hidden markov models. Pattern Recogn. 42(11), 2637–2648 (2009)
17. Perina, A., Cristani, M., Castellani, U., Murino, V., Jojic, N.: Free energy score space. In: Advances in Neural Information Processing Systems, vol. 22, pp. 1428–1436 (2009)
18. Duda, R., Hart, P., Stork, D.: Pattern Classification. John Wiley and Sons, Chichester (2001)
19. Smith, N., Gales, M.: Speech recognition using svms. In: NIPS, pp. 1197–1204 (2001)
20. Duin, R., Juszczak, P., Paclík, P., Pekalska, E., DeRidder, D., Tax, D.: Prtools version 4.1: A matlab toolbox for pattern recognition. Internet (2004), http://www.prtools.org
21. Fukunaga, K.: Statistical Pattern Recognition, 2nd edn. Academic Press, London (1990)
22. Lissack, T., Fu, K.: Error estimation in pattern recognition via l-distance between posterior density functions. IEEE Trans. Inform. Theory 22, 34–35 (1976)

A Modular Approach to Training Cascades of Boosted Ensembles

Teo Susnjak, Andre L. Barczak, and Ken A. Hawick

Institute of Information and Mathematical Sciences,
Massey University, Albany, New Zealand
teo.susnjak.1@uni.massey.ac.nz
http://iims.massey.ac.nz

Abstract. Building on the ideas of Viola-Jones [1] we present a framework for training cascades of boosted ensembles (CoBE) which introduces further modularity and tractability to the training process. It addresses the challenges faced by CoBE frameworks such as protracted runtimes, slow layer convergences and classifier optimization. The framework possesses the ability to bootstrap positive samples and may in turn be extended into the domain of incremental learning. This paper aims to address our framework's susceptibility to overfitting with possible solutions. Experiments are conducted on face detectors using the bootstrapping of large positive datasets and their accuracy, with respect to overfitting, is examined.

Keywords: cascades of boosted ensembles, AdaBoost, classification, classifier training, face detection.

1 Introduction

Face detection has received much attention in recent years in the field of computer vision. Though a number of notable face detectors with accurate and fast execution runtimes in controlled environments have been developed, the problem of developing robust face detectors that operate in variable environments is still an open problem.

The most successful methods so far have been extensions of the seminal work by Viola-Jones [1], which combined AdaBoost as the learning algorithm together with Haar-like features that can be computed rapidly through integral images. The key feature of this detector was the decomposition of a monolithic ensemble of boosted weak classifiers into cascades.

Despite the successes achieved using cascades of boosted ensembles in both accuracy and real-time performance, one of the greatest obstacles to their wider proliferation when deployed in face detection or similarly computationally intensive domains, lies in their protracted training runtimes [2]. Though massive feature spaces are an obvious contributing factor, particularly as dataset sizes increase [3], other factors are slow training convergences [1] and limited classifier optimization capabilities [4]. Additionally, the lack of positive sample boostrapping capabilities of CoBEs has meant that all positive samples needed to be

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 640–649, 2010.

learned simultaneously, thus prohibiting the usage of massive positive datasets. Lastly, the limited abilities of the CoBE frameworks to learn incrementally also leads to significant total training runtime overheads in instances where it is not feasible, requiring the re-training of entire classifiers each time new and *relevant* datasets become available.

[3] minimize the problem of massive feature spaces by applying statistical methods and assumptions to it regarding its distribution and achieve a dramatic reduction in the amount of time required to train each weak classifier while [5] employs feature filtering. [6] attempted to accelerate the cascade layer convergence speed by strengthening the discriminatory ability of the feature types. Alternatively, [7] and others have modified the AdaBoost learning algorithm to produce variants with same intentions, however none have significantly contributed to a training runtime reduction in respect to faster layer convergences. Automating the optimization of cascade parameters remains an unsolved problem though [4] provided significant contributions.

Only recently has research [8,9] surfaced with methods to enable positive sample bootstrapping. While, [10] introduces on-line incremental learning using AdaBoost implemented using neural networks rather than CoBEs.

The PSL (*Parallel Strong* classifier within the same *Layer*) training framework introduced by Barczak et al [11] originally sought to address the convergence bottleneck during the training of cascade layers. However, the modularity of the approach also simplified cascade optimization. Moreover, it provided the basis for addressing the issue of bootstrapping positive samples, seen in initial experiments on the Bootstrapped Dual-Cascaded framework (BDC) [12], as well as for further extensions that enable incremental learning.

The shortcomings of the PSL-based frameworks, have been an elevation in false detection rates due to a tendency to overfit. This characteristic has been more evident in rare-event domains like face detection where exceptionally low false positive rates are needed in order to produce practical detectors.

The purpose of this paper is to explore the causes of overfitting in PSL-based frameworks and to present modifications to them which preserve their ability to rapidly train real-time execution-capable classifiers. In order to provide a thorough analysis of the overfitting issue, this paper will make use of the face detection classifiers from [12], which were created using the positive sample bootstrapping method (BDC) and will compare them with classifiers trained using the modified BDC framework designed to address the overfitting.

The structure for this paper is as follows: Section 2 sets forth the fundamental ideas of modularizing CoBE training using the PSL-based method. Section 3 discusses extensions to PSL which led to the development of the BDC framework that enabled positive sample bootstrapping. The same section explores the framework's ability to implement incremental learning. Following sections present the analysis of the occurrence of overfitting in these frameworks and propose a solution to it. Subsequent sections explain the implementation of the experiments followed by their analysis and a conclusion.

2 PSL Training Framework

The architecture of the PSL framework can be seen in Figure 1b and is contrasted with the standard cascading approach of Viola-Jones in Figure 1a. The PSL architecture extends the standard cascading structure by introducing an additional nested cascade within each layer of a strong classifier, thus creating a quasi two dimensional cascade structure. While the Viola-Jones approach executes an independent round of AdaBoost training for each layer, the PSL framework executes multiple independent rounds of AdaBoost within each layer and in the process constructs a complementing cascade with an alternate goal. We refer to each layer of an internal cascade as an *intra-layer stage*.

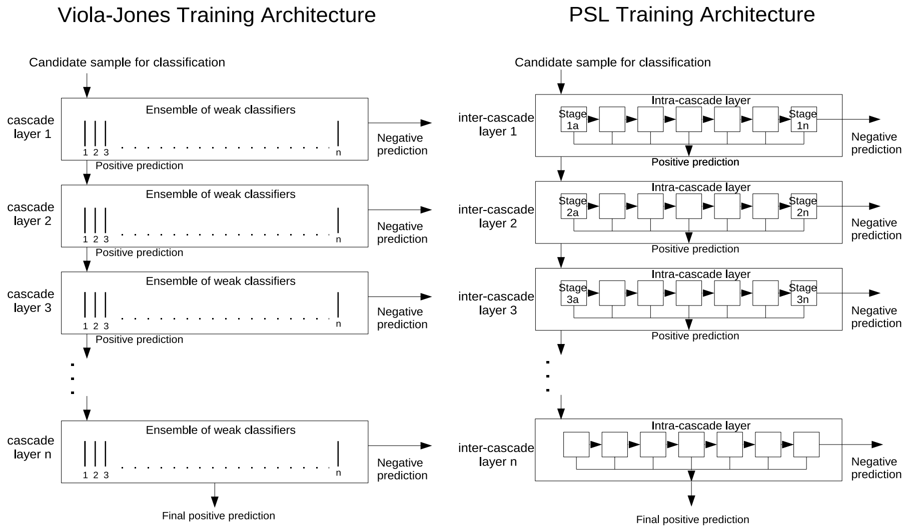


Fig. 1. a) The standard cascade structure of Viola-Jones. b) the PSL structure [11].

Whereas the cascading of the Viola-Jones method focuses on rejecting negative training samples, the intra-layer cascading of the PSL framework focuses on correctly predicting positive samples. Thus, the underlying principle found in the Viola-Jones method with respect to its approach to handling more difficult negative samples with each succeeding layer, is replicated to the positive samples in the internal stage-to-stage propagation. The propagation of the positive training samples of the PSL framework is seen in Figure 2a. As the intra-layer cascade of stages is constructed, correctly predicted positive samples are removed from succeeding stages while the misclassified positives are retained until all the positive samples have been correctly predicted. By removing correctly predicted positives, faster layer convergences are realized, while 100% hit rates are attained without artificial threshold adjustments, thus ultimately resulting in accelerated overall training runtimes.

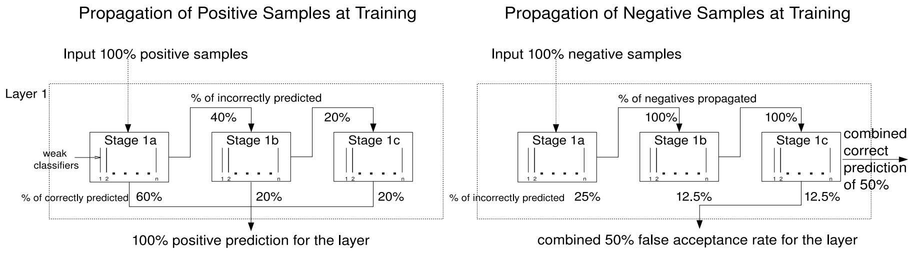

Propagation of Positive Samples at Training
Input 100% positive samples
Layer 1
% of incorrectly predicted
40%
20%
Stage 1a
Stage 1b
Stage 1c
weak classifiers
1 2 n
% of correctly predicted
60%
20%
20%
100% positive prediction for the layer
Propagation of Negative Samples at Training
Input 100% negative samples
% of negatives propagated
100%
100%
Stage 1a
Stage 1b
Stage 1c
combined correct prediction of 50%
% of incorrectly predicted
25%
12.5%
12.5%
combined 50% false acceptance rate for the layer

Fig. 2. a) The propagation of positive training samples within the cascade of PSL stages inside a layer. 1 b) The usage of negative training samples within the cascade of PSL stages inside a layer.

During training, all negative training samples propagate to each stage irrespective of how successfully previous stages have learned to predict them as seen in Figure 2b. Each stage is assigned a target to learn to reject 50% of the negative samples and to achieve a 100% hit rate. However, a key constraint in the form of a maximum number of weak classifiers is added to each stage which not only accelerates layer convergences but also simplifies classifier optimization through the variation in size of this constraint at different layers.

At detection time, the classification process also becomes modularized and more efficient. A candidate sample is predicted as a negative by a layer only if *all* nested stages within it classify it as a negative. A sample is predicted as a positive once *any* nested stage predicts it as a positive thereby not requiring the computation of the remaining internal stages.

3 Positive Sample Bootstrapping

The BDC framework builds upon the concepts of the PSL structure and extends it in order to implement a positive sample bootstrap capability. Unlike the positive sample bootstrapping approaches of [8,9], the BDC training framework utilizes the modularity offered by the PSL's nested cascade-of-stages to achieve further malleability. Through a strategy of divide and conquer, massive positive datasets can be employed while only a fraction of its samples undergo training at each stage.

The whole negative dataset and a subset of the entire positive dataset constitute the training sets used for each stage of a BDC nested cascade. The positive sample subset which the learning algorithm *sees* and trains on explicitly we call the *base set*. The entire positive dataset from which new positive samples are bootstrapped is referred to as the *reserve set*.

The procedure for intra-layer cascade training can be seen in Algorithm 1. The training of an intra-layer cascade initiates with randomly selecting a comparably small subset of positive samples from the reserve set in order to construct the base set. The base set is then trained against the negative dataset to produce

```
Given:
$C_n$ = $n_{th}$ inter-layer layer sub-classifier
$S_i$ = $i_{th}$ intra-layer stage sub-classifier
$PB_i$ = positive base set used on $S_i$
$PR$ = positive reserve set
$f_{min}$ = minimum false acceptance rate
$d_{min}$ = minimum required hit rate set at 100%
$WK_{max}$ = max number of weak classifiers
1 randomly select positive samples from $PR$ to create $PB_i$
2 train $C_nS_i$ against $PB_i$ until $f_{min}$ and $d_{min}$ or $WK_{max}$
3 validate $PR$ using $C_nS_i$ and remove from it correctly classified samples
4 if all samples in $PR$ have been correctly predicted then start a new layer
  $C_{n+1}$ otherwise start new stage $S_{i+1}$ repeat step 1
```

Algorithm 1. BDC bootstrapping method for each cascade layer

individual stages. Each stage of this nested cascade is trained with a target hit rate of 100% and a high rejection rate. As in PSL, the size of each nested stage is restricted by the maximum number of weak classifiers that can comprise it. Once this maximum number has been reached, the training for that stage ceases and a new intra-layer stage begins. The positive bootstrapping procedure is then initiated. The positive samples in the reserve set are validated against the resulting stage classifier and all correctly predicted samples are removed from training subsequent nested stages. The remaining positive samples are randomly selected to comprise the new *base set* for the next intra-layer stage together with all the incorrectly predicted positive samples from the previous stage's base set.

3.1 Incremental Learning with PSL

The modular nature of the PSL framework, combined with the ideas from BDC leads to the possibility of implementing effective incremental learning in a novel approach. Incremental learning can be achieved in this scenario by constructing additional intra-layer stages trained on new positive samples which are incorrectly predicted at each layer. The new stages can then either be appended to the existing cascade-of-stages or a strategy can be devised to replace less accurate existing stages with new ones. The incremental training would be initiated on batches of incorrectly predicted positive samples once they reach the minimum required number for each *base set.* The composition of the negative set is less trivial and has to consist of similar patterns which previous stages in a layer have learned to predict otherwise false detection rates for a layer would increase. It is proposed that the negative set comprises of those images which have up until that cascade layer been misclassified and that a substantially larger negative dataset be used for incremental learning than that of the initial off-line training phase.

3.2 PSL Framework and Overfitting

Experiments in [12] have demonstrated the capability of the BDC framework to potentially train classifiers on massive positive datasets with relatively small increases in training runtimes whilst maintaining 100% layer hit rates on the training data. The face detectors trained in those experiments showed that the training runtimes using the BDC bootstrapping method result in a fraction of the runtimes required by standard training structures without bootstrapping. However, the framework also exhibited a susceptibility to elevate false acceptance rates which makes it less suitable for rare-event operating domains like face detection.

Further analysis of the classifiers obtained in [12] has identified varying degrees of overfitting occurring in final intra-layer stages. The nature of the BDC training approach delays training most difficult positive samples until the last stages. These stages often tend to be trained on positive datasets that comprise of a small number of samples which mostly contain highly unrepresentative patterns in respect to the overall positive dataset. Figure 3 shows examples of images trained by first intra-layer stages and contrasted with those learned in latter stages. The figures point to large concentrations of positive images in final stages which exhibit extensive variations in illumination and also occlusions of vital facial features.

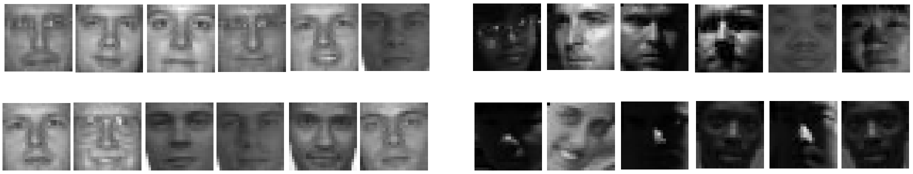


Fig. 3. Examples of positive images learned at different points within the cascade-of-stages on a 15000 sample BDC classifier. Cluster a) stage 1 layer 30 b) last stage layer 30 c) stage 1 layer 42 d) last stage layer 42.

The accuracy of training that takes place in the trailing stages is further compromised by high weights assigned to the positive samples initially before each round of boosting. This occurs since an even 50%-50% distribution of total weights is shared between the positive and negative training sets irrespective of their sizes. Consequently, as the number of stages in each layer grows, fewer positive samples remain. This leads to a proportional increase in their weights, while at the same time, their patterns also become less representative of the whole dataset.

In order to demonstrate the effects of the final stages on classifiers' accuracy, we compared the generalization patterns of classifiers, with and without their last stages, using receiver operating curves (ROC) in Figures 4a-c. The data shows improved generalization of truncated classifiers particularly for segments of the ROC graphs which portray the lower end of false acceptance rates.

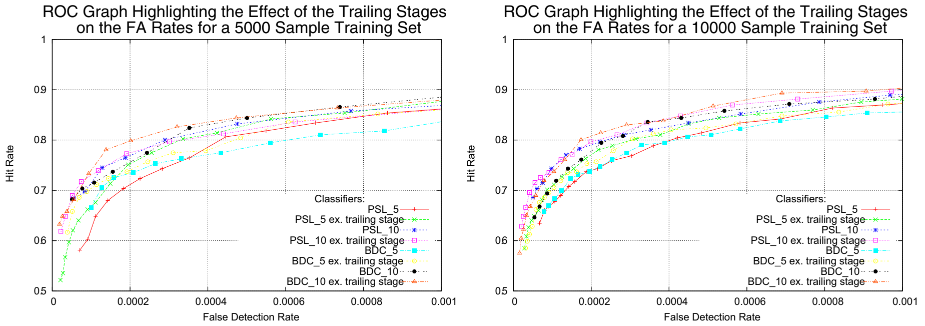


Fig. 4. ROC graphs displaying the generalization patterns of the BDC classifiers with their final intra-layer stages excluded

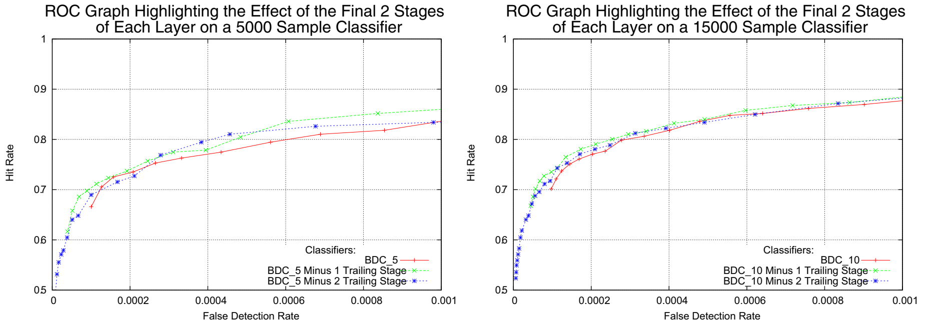


Fig. 5. ROC graphs displaying the generalization patterns of a 5000 and 15000 BDC classifiers with their final two intra-layer stages excluded

The ROC graphs in Figures 5a-b go a step further and demonstrate the effects of excluding the last two stages of each cascade. In both instances, an improvement in the generalization of the truncated classifiers is observed indicating that a degree of overfitting is occurring.

It can be concluded that the effectiveness of each cascade layer is only as strong as the accuracy of its weakest stage. Overall, the generalization ability of a BDC classifier can be summarized as being only as strong as the combined accuracy of all its weakest stages from each layer.

3.3 Anti-overfitting Modifications

Our proposed solution to the problem of overfitting found in the underlying foundation of the BDC structure, is based on incorporating additional positive samples into the datasets of the trailing stages of each layer. With this strategy, our intent is to offset the overfitting brought on by a high concentration of less representative positives. We propose augmenting the trailing stages of each layer with positive samples which have already been correctly predicted by previous stages. By including these samples into the dataset a degree of protection against overfitting is expected to be achieved and thus the likelihood of producing more generalizable intra-layer stages.

The inclusion of redundant positive samples is also expected to have negative effects. The learning process will become more complicated since the convergence speed of layer targets towards required 100% hit rates will decrease and a degree of weak classifier redundancy is likely to be introduced. In order to assist rapid layer convergences, greater weights are initially assigned to *relevant* positive samples at the beginning of each boosting round. Additionally, to counterbalance the generation of an exceeding number of intra-layer stages, the requirement to maintain fixed stage sizes is removed. Instead, the maximum number of weak classifiers is increased as the number of the misclassified positive samples, in respect to the size of the base set, decreases. Since generating a greater number of weak classifiers on a small base set can itself result in overfitting, we also increased the base sets from 500 in prior experiments [12] to 2000 positive samples.

4 Method

The experiments consisted of training face detection classifiers using the modified BDC structure and comparing it to the classifiers trained by the original *naive* BDC structure in [12]. The datasets used on all training were identical as were the parameters. The total of 15000 facial images were collected from various publicly available datasets; FERET, Yale *Face Database B* [13] and the face database from the Vision Group of Essex University. Three main groups of classifiers were trained which were divided into 5000, 10000 and 15000 sample datasets. For each dataset, a classifier was trained with a flexible stage size of 10 weak classifiers. All classifiers were trained with a base set size of 2000 positives against 2000 negatives extracted from a total of 2500 images which generated millions of negative sub-windows. An additional set of classifiers using the naive BDC were trained on base sets of 2000 positive samples in order to isolate the proposed increase in base sets as the determining factor in addressing overfitting. Finally, classifiers were trained to attain a 0% training error in a maximum of 100 layers, using no more than 50 stages per layer.

Testing was performed on the CMU MIT image dataset containing 130 images which contain 506 positive face images.

5 Results

The BDC classifiers with overfitting adjustments generated training runtimes that were 15%-20% longer than those of the naive BDC, however they were still significantly lower than those of the PSL and Viola-Jones. Classifiers trained on naive BDC with base sets of 2000 produced shortest training runtimes, thus highlighting a modest additional cost involved in our approach.

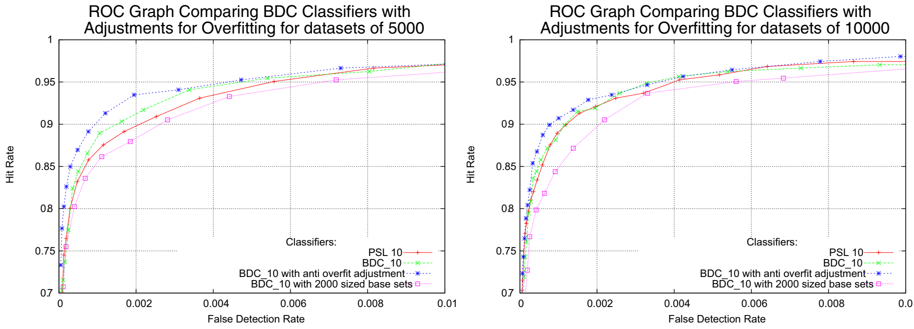


Fig. 6. ROC graphs displaying the generalization patterns of the modified BDC structure

Additionally, the size of the modified BDC classifiers increased over the naive implementation by 15%-20% extra weak classifiers which is likely to incur a larger detection runtime cost too. Both structures generated similar numbers of stages per layer, which ranged from three in earlier layers, through to six as training became more difficult.

Figures 6a-b show the generalization patterns of the classifiers. In both figures, it is evident that the modified BDC classifiers have achieved a superior generalization over all other classifiers on the CMU MIT test dataset. It is worth noting that the weakest accuracy was exhibited by the naive BDC classifiers trained on base sets of 2000 positive samples. This eliminates the possibility of attributing improvements in accuracy of the modified BDC to solely its increase in base set sizes, but instead demonstrates that the solution to overfitting was the result of a combined new strategy.

6 Conclusion

In this paper we demonstrated how classifier training using CoBEs can be modularized using the PSL framework, thereby addressing issues of slow convergence rates and protracted training runtimes, while eliminating many of the post-training classifier optimization overheads. The framework's ability to implement positive sample bootstrapping on large datasets was put forward and its potential to enable incremental learning was also introduced. A thorough analysis of the framework's susceptibility to overfit data was presented, to which an effective solution was proposed.

Future research will focus on extending PSL to enable incremental learning.

References

1. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR 2001, Kauai, HI, December 2001, vol. I, pp. 511–518. IEEE, Los Alamitos (2001)

2. Brubaker, S.C., Mullin, M.D., Rehg, J.M.: Towards optimal training of cascaded detectors. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 325–337. Springer, Heidelberg (2006)
3. Pham, M.T., Cham, T.J.: Fast training and selection of haar features using statistics in boosting-based face detection. In: IEEE 11th International Conference on Computer Vision, 2007, October 2007, pp. 1–7 (2007)
4. Brubaker, S.C., Wu, J., Sun, J., Mullin, M.D., Rehg, J.M.: On the design of cascades of boosted ensembles for face detection. Int. J. Comput. Vision 77(1-3), 65–86 (2008)
5. Wu, J., Rehg, J.M., Mullin, M.D.: Learning a rare event detection cascade by direct feature selection. In: NIPS Advances in Neural Information Processing Systems 2003, Vancouver, Canada (2003)
6. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: ICIP 2002, Rochester, NY, September 2002, vol. I, pp. 900–903 (2002)
7. Viola, P.A., Jones, M.J.: Fast and robust classification using asymmetric adaboost and a detector cascade. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) NIPS, pp. 1311–1318 (2001)
8. Xiao, R., Zhu, H., Sun, H., Tang, X.: Dynamic cascades for face detection. In: IEEE 11th International Conference on Computer Vision 2007, October 2007, pp. 1–8 (2007)
9. Yan, S., Shan, S., Chen, X., Gao, W., Chen, J.: Matrix-Structural Learning (MSL) of cascaded classifier from enormous training set (2007)
10. Huang, C., Ai, H., Yamashita, T., Lao, S., Kawade, M.: Incremental learning of boosted face detector. In: ICCV, pp. 1–8. IEEE, Los Alamitos (2007)
11. Barczak, A.L.C., Johnson, M.J., Messom, C.H.: Empirical evaluation of a new structure for adaboost. In: SAC 2008, pp. 1764–1765. ACM, New York (2008)
12. Susnjak, T., Barczak, A.L.C., Hawick, K.A.: A novel bootstrapping method for positive datasets in cascades of boosted ensembles. Research Letters in the Information and Mathematical Sciences Vol. 14, pp.17-24, Institute of Information and Mathematical Sciences, Massey University Albany (2010) ISSN 1175-2777
13. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intelligence 23(6), 643–660 (2001)

A Linear Combination of Classifiers via Rank Margin Maximization

Claudio Marrocco, Paolo Simeone, and Francesco Tortorella

DAEIMI - Università degli Studi di Cassino
Via G. Di Biasio 43, 03043 Cassino (FR), Italia
{c.marrocco,paolo.simeone,tortorella}@unicas.it

Abstract. The method we present aims at building a weighted linear combination of already trained dichotomizers, where the weights are determined to maximize the minimum rank margin of the resulting ranking system. This is particularly suited for real applications where it is difficult to exactly determine key parameters such as costs and priors. In such cases ranking is needed rather than classification. A ranker can be seen as a more basic system than a classifier since it ranks the samples according to the value assigned by the classifier to each of them. Experiments on popular benchmarks along with a comparison with other typical rankers are proposed to show how effective can be the approach.

Keywords: Margin, Ranking, Combination of Classifiers.

1 Introduction

Many effective classification systems adopted in a variety of real applications make a proficient use of combining techniques to solve two class problems. As a matter of fact the combination of classifiers is a reliable technique to improve the overall performance, since it exploits the strength of the classifiers to be combined while reduces the effects of their weaknesses. Moreover the fusion of already available classifiers gives the user the opportunity to obtain simply and quickly an optimized system using them as building blocks, thus avoiding to restart from the beginning the design of a new classification system.

Several methods have been proposed to combine classifiers [11] and, among them, one of the most common technique is certainly the linear combination of the outputs of the classifiers. Extended studies have been conducted on this issue [8], and in particular have considered the weighted averaging strategies which are the basis of some popular algorithms like Bagging [2] or Boosting [7]. Boosting techniques build a classifier as a convex combination of several *weak* classifiers; each of them is in turn generated by dynamically reweighing training samples on the basis of previous classification results provided by the weak classifiers already constructed.

Such approach revealed to be really effective in obtaining classifiers with good generalization characteristics. To this regard, the work of Schapire et al. [13] has analyzed the boosting approach in terms of *margin maximization*, where the *margin* is a measure for the accuracy confidence of a classifier which can be considered as an important indicator of its generalization capacity. They calculated an upper bound on the generalization

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 650–659, 2010.

error of resulting classifier and showed how the increase of the margin corresponded to an improvement of such bound. However, it is worth noting that this framework is applicable only in the cases where the accuracy is the most suitable index to evaluate the performance of the classification system, i.e. when the values of the classification costs and of the priors are known and fixed. For applications for which these parameters are not precisely known or are changing over time (*imprecise environments*), the accuracy becomes useless and other indices should be preferred such as the *Area under the ROC curve* (*AUC*). To understand the reason for this preference, we have to recall that, when the accuracy is used, we assume that a threshold is fixed on the classifier output on the basis of given costs and priors; accordingly, the accuracy measures the probability that the samples to be classified are correctly ordered with respect to the threshold. On the other side, the AUC measures the probability that a classifier correctly ranks two samples belonging to opposite classes and does not take into account any threshold; in other words, AUC provides an evaluation of the classifier quality independent of a particular setting of costs/priors.

In this framework, the concept of margin cannot be used and the *rank margin* should be employed instead, which gives a measure of the ranking confidence of the classifier. On this basis, Rudin et al. [12] have studied the generalization capability of RankBoost [6], a learning algorithm expressly designed to build systems for ranking preferences, and defined some bounds related to the rank margin value reached during the training phase. However these papers focus exclusively on how to build a new classifier from the scratch.

The aim of this paper is different from [12] and [6] since it presents a method to build a linear combination of already trained dichotomizers. The weights are determined in such a way to maximize the rank margin of the resulting system and thus to optimize its performance in terms of AUC. Several experiments performed on publicly available data sets have shown that this method is particularly effective.

The paper has been organized as follows: in section 2 the concepts of margin and rank margin are briefly explained together with their characteristics, while section 3 presents the method for calculating the weights of the linear combination based on the rank margin maximization. In section 4 experiments on some popular benchmark data are illustrated. Finally, in section 5 we draw some conclusions and propose some future developments.

2 Margins and Ranking

Let us consider a two class problem defined on a training set $S = (X, Y)$ containing N samples $X = \{\mathbf{x}_i\}$ associated to N labels $Y = \{\mathbf{y}_i\}$ with $y_i \in \{-1, +1\}$ where $i = 1, \cdots, N$. A classifier f can be described as a mapping from X to the interval $[-1, +1]$ such that a sample $\mathbf{x} \in X$ is assigned to one of the classes according to $\operatorname{sgn}(f(\mathbf{x}))$. If we assume that y_i is the correct label of $\mathbf{x}_i$, the *sample margin* (or *hard margin*) associated to $\mathbf{x}_i$ is given by $y_i f(\mathbf{x}_i)$. As a consequence, f provides a wrong prediction for $\mathbf{x}_i$ if the sample margin is negative.

Generally *the margin of a classifier* (or *minimum margin*) f can be defined as the minimum margin value over the training set: $\mu(f) = \min_i(y_i f(\mathbf{x}_i))$. The classifier margin has a straightforward interpretation [4]: it is the distance that the classifier can travel in the feature space without changing the way it labels any of the sample points and thus, it represents one of the most relevant factor for improving generalization.

However, the concept of margin can not be used when we are in an imprecise environment where priors and costs are not known. In such a case a ranker becomes more useful than a classifier. The notion of ranking is germane to that of classification. In particular, ranking can be seen as an action on data more basic than classification: if no threshold is imposed on the output of the classifier (i.e. we are evaluating its performance independently of class priors and costs), the only possible operation is to rank the samples according to the value assigned by the classifier to each of them. Thus, the margin of a classifier should be replaced by the margin of the ranking function. To illustrate this point, let us define *crucial pair* and indicate with the concise notation (i, k) a pair of samples $\mathbf{x}_i \in X$ and $\mathbf{x}_k \in X$ associated respectively to a positive and a negative label $y_i = +1$ and $y_k = -1$. The term *crucial* is due to the fact that, for this kind of pairs, the classifier should guarantee that $f(\mathbf{x}_i) > f(\mathbf{x}_k)$, while this is not required for two samples belonging to the same class. On this basis, the *crucial pair margin* can be defined as the difference $f(\mathbf{x}_i) - f(\mathbf{x}_k)$; it is evident that a negative value for the margin indicates that the corresponding pair is erroneously ranked. Analogously to the sample margin, it is possible to define the *margin of the ranking function* or *rank margin* as the minimum value of the margin over all the existing crucial pairs:

$$\rho(f) = \min_{(i,k):\, \substack{i\ =\ 1,\ldots,N^+ \\ k\ =\ 1,\ldots,N^-}} \Big(f(\mathbf{x}_i) - f(\mathbf{x}_k) \Big). \tag{1}$$

As for classification, the rank margin theory has been used as a tool to analyze the generalization ability of learning algorithm for rankers based on boosting techniques. An algorithm belonging to this category is RankBoost [6] where the redistribution of the weights on the crucial pairs is done after the weak learners have been employed for ranking the pairs. As for AdaBoost [13], it has been proved that there is a strict relation between the generalization capability of RankBoost and its rank margin maximization. It is worth noting, however, that this method does not rely on a global optimization of the rank margin, but works locally. In fact, at each iteration of Rankboost, the crucial pairs with the minimum rank margin receive the highest weights and thus affect the construction of the whole ranker. Notwithstanding, this process converges towards the maximization of the rank margin [12].

Another issue to be pointed out is that this algorithm only constructs from the scratch an ensemble of classifiers as different instances of a same base learning algorithm. Instead, as far as we know, the potential effectiveness of such a combination has not yet been examined when the classifiers of the ensemble are built independently and not according to a boosting approach.

3 Rank Margin Maximization via Linear Programming

In this section we extend the concept of rank margin to the combination of K already trained classifiers $f_j(\mathbf{x}) \rightarrow [-1, +1]$ with $j = 1, \ldots, K$. Let us consider the N^+ and

N^- samples of the training set *X*. The rank margin provided by the *j*-th classifier over the crucial pair (i, k) is defined as:

$$\rho_{(i,k)}(f_j) = f_j\left(\mathbf{x}_i\right) - f_j\left(\mathbf{x}_k\right), \quad i = 1, 2, \ldots, N^+, k = 1, 2, \ldots, N^- \tag{2}$$

i.e., f_j correctly ranks $\mathbf{x}_i$ iff $\rho_{(i,k)}(f_j) > 0$. Let us now consider the linear combination of the K classifiers:

$$f_c\left(\mathbf{x}\right) = \sum_{j=1}^{K} w_j f_j(\mathbf{x}) \tag{3}$$

with $w_j \geq 0$ and $\sum\limits_{j=1}^{K} w_j = 1$. The rank margin provided by f_c over the crucial pair (i, k) is thus

$$\rho_{(i,k)}(f_c) = \sum_{j=1}^{K} w_j f_j(\mathbf{x}_i) - \sum_{j=1}^{K} w_j f_j(\mathbf{x}_k) = \sum_{j=1}^{K} w_j \rho_{(i,k)}(f_j) \tag{4}$$

while the margin of f_c is $\rho = \min\limits_{(i,k)} \rho_{(i,k)}(f_c)$. Actually the margin ρ depends on the weights $\mathbf{w} = \{w_1, w_2, \cdots, w_K\}$ and thus such weights can be chosen to make the margin as large as possible. In this way we have a max-min problem which can be written as:

$$\begin{array}{ll}
\text{maximize} & \left(\min\limits_{i} \sum\limits_{j=1}^{K} w_j \rho_{(i,k)}(f_j)\right) \\
\text{subject to} & \sum\limits_{j=1}^{K} w_j = 1 \\
& \quad w_j \geq 0 \qquad\qquad j = 1, 2, \ldots, K
\end{array}$$

The problem can be recast as a linear problem [15] if we introduce the margin ρ as a new variable:

$$\begin{array}{ll}
\text{maximize} & \rho \\
\text{subject to} & \\
& \sum\limits_{j=1}^{K} w_j \rho_{(i,k)}(f_j) \geq \rho \quad i = 1, 2, \ldots, N^+, k = 1, 2, \ldots, N^- \\
& \quad \sum\limits_{j=1}^{K} w_j = 1 \\
& \quad\quad w_j \geq 0 \qquad j = 1, 2, \ldots, K
\end{array}$$

If we collect the margins in a $N^+N^- \times K$ matrix $\mathbf{R} = \{\rho_{(i,k)}(f_j)\}$, the weights in a vector $\mathbf{w}$ and define $\mathbf{e}_t$ the column vector consisting of t ones and $\mathbf{z}_t$ the column vector consisting of t zeros, the problem can be written in block-matrix form:

$$
\begin{aligned}
&\text{maximize} \qquad \begin{bmatrix} \mathbf{z}_K^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mu \end{bmatrix} \\
&\text{subject to} \\
&\qquad \begin{bmatrix} -\mathbf{R} & \mathbf{e}_N \\ \mathbf{e}_N^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \rho \end{bmatrix} \begin{matrix} \leq \\ = \end{matrix} \begin{bmatrix} \mathbf{z}_N \\ 1 \end{bmatrix} \\
&\qquad \mathbf{w} \geq \mathbf{z}_K
\end{aligned}
$$

As a final remark, it is worth noting that to solve this problem we could use any one of the numerous linear programming methods available. However, it should be taken into account that the number of constraints could be very large since it equals the number of crucial pairs in the training set.

4 Experiments

Ten publicly available two class data sets were chosen from the UCI machine learning repository [1] to evaluate the performance of our approach. A summary of the employed data sets is reported in table 1. The features were previously scaled in order to have zero mean and unitary standard deviation. To avoid any bias in the comparison, 10 runs of a multiple hold out procedure have been performed on all the data sets. Each data set has been divided in three parts: a training set for the dicothomizers, a tuning set to train the combiner in order to have the optimal weights and a test set to evaluate the performance.

Modest AdaBoost [16] has been chosen as base classifier. Its algorithm adopts a CART decision tree with a maximum depth equal to 3 and decision stumps as nodes functions and a number of boosting steps equal to 10. To have a lower correlation between the built classifiers a random, but uniformly distributed, weight initialization has been done.

In order to compare the combining rules we considered the AUC as a performance measure. AUC values are unitary when all the instances are correctly interpreted by the learner, i.e. what is called a separable case. In terms of ranking it means that the algorithm is consistent with all the crucial pairs: all the positive instances are ranked

Table 1. Summary of the used data sets

Name	Samples	Features	% N^+	% N^-
Australian	690	14	44.49	55.51
Balance	625	4	54.01	45.99
Breast	699	16	65.01	34.99
Cleveland	303	13	54.13	45.87
Contraceptive	1473	9	42.70	57.30
Hayes	132	4	50.39	49.61
Housing	506	12	49.21	50.79
Ionosphere	351	34	64.10	35.90
Liver	345	6	57.97	42.03
Sonar	260	60	53.37	46.63

Table 2. AUCs obtained using 5 classifiers

Data Sets	**RankMargin**	**RankBoost**	**SVM**
Australian	**0.935**(**0.008**)	0.920(0.008)	0.929(0.009)
Balance	0.984(0.001)	0.959(0.016)	**0.986**(**0.004**)
Breast	**0.991**(**0.001**)	0.979(0.003)	0.979(0.010)
Cleveland	**0.885**(**0.010**)	0.840(0.026)	0.858(0.025)
Contraceptive	0.751(0.024)	0.752(0.013)	**0.762**(**0.012**)
Hayes	0.885(0.014)	0.865(0.030)	**0.893**(**0.039**)
Housing	**0.942**(**0.007**)	0.924(0.012)	**0.940**(**0.012**)
Ionosphere	**0.962**(**0.003**)	0.927(0.011)	0.944(0.019)
Liver	**0.737**(**0.033**)	0.707(0.035)	0.721(0.032)
Sonar	**0.892**(**0.016**)	0.837(0.033)	0.875(0.036)

Table 3. AUCs obtained using 7 classifiers

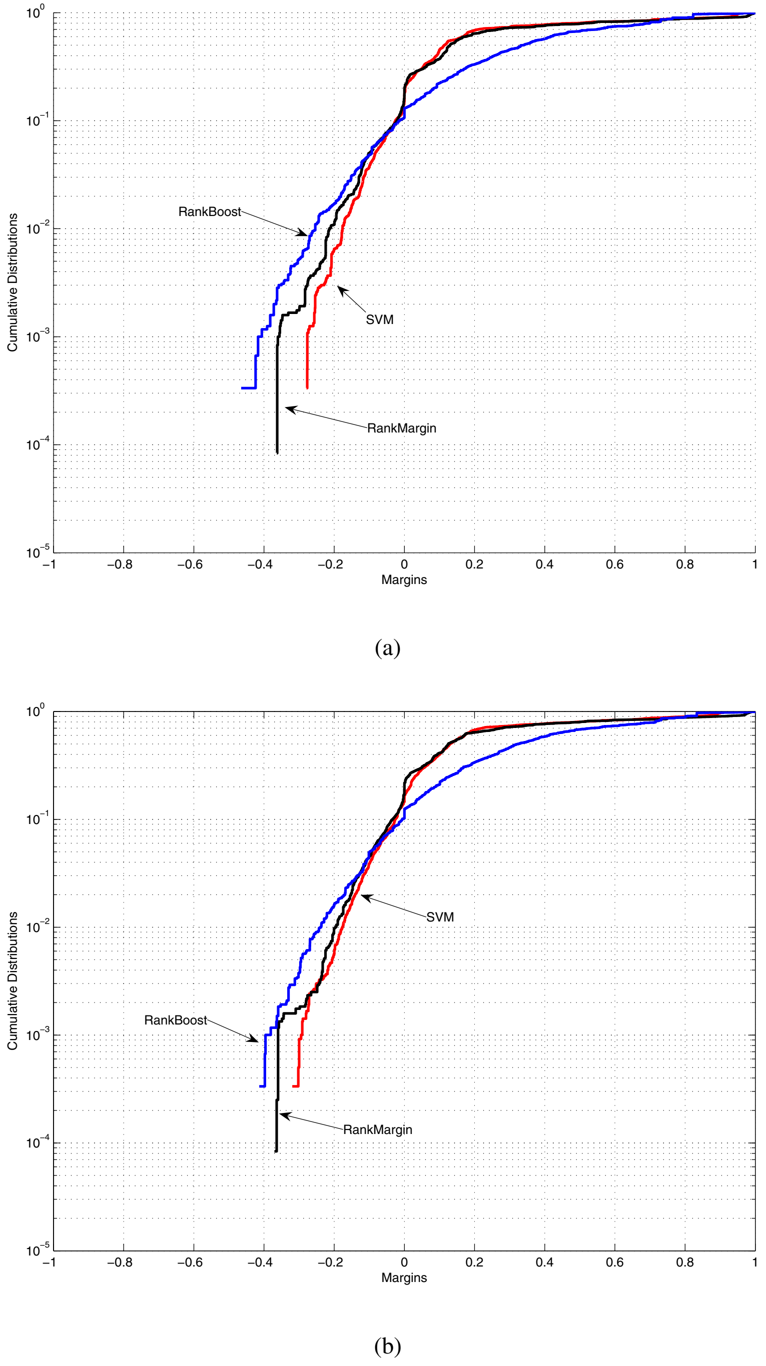
Data Sets	**RankMargin**	**RankBoost**	**SVM**
Australian	**0.932**(**0.007**)	0.920(0.008)	0.921(0.010)
Balance	0.984(0.001)	0.959(0.016)	**0.985**(**0.004**)
Breast	**0.991**(**0.001**)	0.979(0.003)	0.972(0.010)
Cleveland	**0.884**(**0.007**)	0.840(0.026)	0.847(0.022)
Contraceptive	0.753(0.015)	0.751(0.012)	**0.758**(**0.011**)
Hayes	**0.888**(**0.010**)	0.864(0.030)	0.878(0.025)
Housing	**0.942**(**0.005**)	0.924(0.012)	0.932(0.014)
Ionosphere	**0.962**(**0.002**)	0.927(0.011)	0.931(0.020)
Liver	**0.737**(**0.021**)	0.707(0.035)	0.702(0.034)
Sonar	**0.891**(**0.012**)	0.837(0.033)	0.863(0.037)

above the negative. Indeed an higher measure of the AUC is a quality factor for our combining rules.

Two classifiers notable for their ranking capacity have been adopted for a comparison with our RankMargin technique: RankBoost and Support Vector Machines (SVMs). The first one has been implemented by setting $T = 100$ iterations using a Matlab toolbox publicly available [3], the other one has been implemented by using SVM^{light} [10] with a linear kernel and default parameters.

To assess the performance of our method in comparison with the other considered combination rules, we have employed the *Friedman Two-Way Analysis of Variance by Ranks* test [14,5], a statistical non-parametric test which evaluates if in a set of L samples, at least two of the samples represent populations with different median value[1]. In this case, the null hypothesis corresponds to a not statistically significant difference in performance among the combination rules. When the null hypothesis is rejected, the *Holm's step-down procedure* [9,5] is applied as a $post - hoc$ test to identify which rule

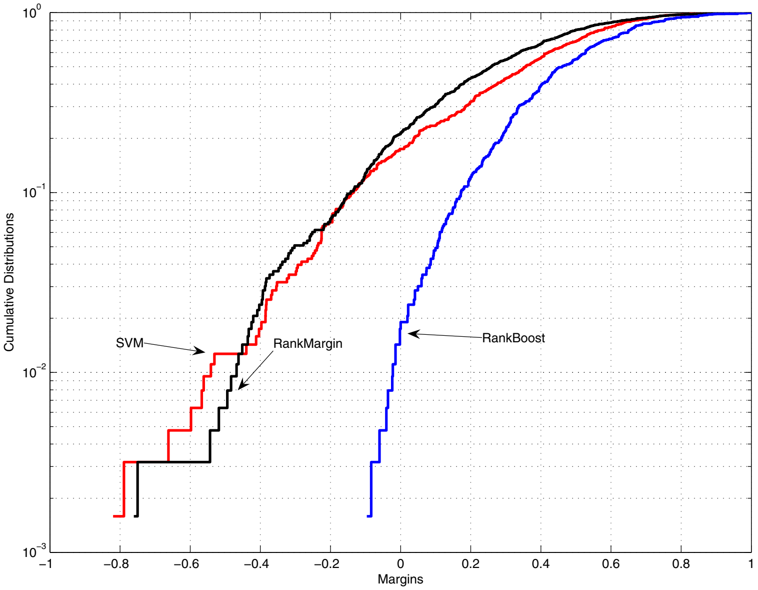
[1] We chose this test since its parametric counterpart, i.e. ANOVA, requires that the samples are drawn from normal distributions and the distributions have equal variance [14] and this is not assured in our test bed.



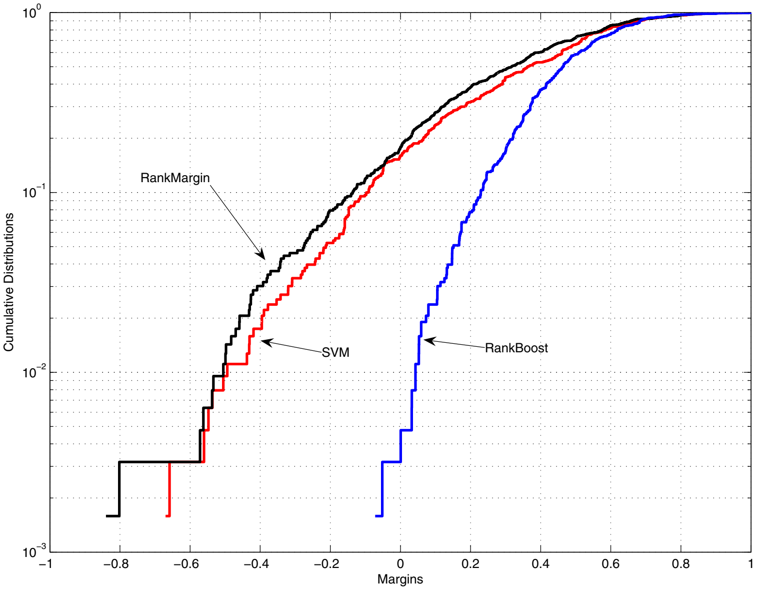
(a)

(b)

Fig. 1. Rank margin distributions graphs for the employed combination rules on the Contraceptive data set when combining 5 (a) and 7 (b) classifiers. The scale on y-axis is logarithmic.



(a)



(b)

Fig. 2. Rank margin distributions graphs for the employed combination rules on the Liver data set when combining 5 (a) and 7 (b) classifiers. The scale on y-axis is logarithmic.

performs significantly better or worse than the proposed method. Both the tests have been performed with $\alpha = 0.01$.

Results in terms of mean AUC (and standard deviation) are shown in tables 2 and 3 which differs for the number of combined classifiers (respectively 5 and 7). A bolded value means that the corresponding ranker has a statistically better performance on such data set.

Performance of our algorithm proved to be better for the majority of examined data sets. In particular in only 3 cases SVMs gave better performance when combining 5 classifiers, while there was a tie for the Housing data set. When combining 7 classifiers the results are even better: 8 out of 10 data sets. It is worth noting that RankBoost never outperforms our method. Some final considerations could be made about the comparison with RankBoost that never outperforms our method. Since RankBoost algorithm is not conceived to maximize the margin of the rank function at each iteration, such result is an empirical proof of how RankMargin gives an improvement of the overall performance of a ranker.

A second experiment has been done to show the behavior of the rank margin based combination rule on the training set. Accordingly we plotted the cumulative distributions of rank margins on the training set provided by RankMargin and the other employed fusion rules. In fig. 1 and 2 we report the margin cdfs for the proposed approach in comparison with the other rules respectively for the Contraceptive and Liver data sets when using 5 and 7 Modest AdaBoost as base classifiers.

The first graphs, both (a) and (b), show that the SVM gives better results on Contraceptive data set. This is perfectly coherent with the test results shown in tables 2 and 3. It can be observed how SVM maintains the same trend observed for training set when predicting test results, thus SVM keeps performing better of RankMargin in this case. RankBoost instead performs worse of both approaches even if the minimum rank margin on the training set is comparable with the other two techniques.

On the other hand in the second graphs it is possible to note that RankBoost exhibits clearly higher performance than the other approaches in terms of minimum rank margin. This is probably due to the fact that the boosting approach focuses on the most difficult samples of the training set to be classified giving almost perfect results on them. Another possible reason is given by the non linear nature of the combination built by RankBoost which could increase the minimum rank margin much more than SVM and RankMargin. Nevertheless, the higher complexity of the RankBoost combination reveals on the test set a worse generalization capability with respect to both SVM and RankMargin. These latter methods construct both a linear combination and thus the distribution of the margins are quite similar. However, SVM provide an optimal separating hyperplane with equal margins from the two classes, while RankMargin has not such a constraint of symmetrical margins and this reflects in a better generalization capability.

5 Conclusions and Future Works

In this paper we have studied a new algorithm to combine scores of base classifiers. Such algorithm aims at the maximization of the margin for the ranking function in order to accomplish a better performance in terms of AUC for the linear combination of already trained dichotomizers. Results on the UCI data sets proved that our approach is reasonable and could be extended to plenty of applications.

Future developments will focus on the application of such technique to highly unbalanced data sets where AUC, which is independent from prior probabilities and costs, is a good performance measure, e.g. biometrics data. Another development can be in the relaxation of the constraint in the rank margin maximization by introducing slack variables that could be useful to face with noisy data.

References

1. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007)
2. Breiman, L.: Bagging predictors. Machine Learning 26(2), 123–140 (1996)
3. Canu, S., Grandvalet, Y., Guigue, V., Rakotomamonjy, A.: Svm and kernel methods matlab toolbox. Perception Systèmes et Information, INSA de Rouen, Rouen, France (2005)
4. Crammer, K., Gilad-Bachrach, R., Navot, A., Tishby, N.: Margin analysis of the lvq algorithm. In: NIPS, pp. 462–469 (2002)
5. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research (7), 1–30 (2006)
6. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research 4, 933 (2003)
7. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of online learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119 (1997)
8. Fumera, G., Roli, F.: A theoretical and experimental analysis of linear combiners for multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(6), 942 (2005)
9. Holm, S.: A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics 6, 65–70 (1979)
10. Joachims, T.: SVM light (2002), http://svmlight.joachims.org
11. Kuncheva, L.I.: Combining Pattern Classifiers. Methods and Algorithms. John Wiley & Sons, Chichester (2004)
12. Rudin, C., Cortes, C., Mohri, M., Schapire, R.: Margin-based ranking meets boosting in the middle. In: Proceedings of 18th Annual Conference on Computational Learning Theory (2005)
13. Schapire, R.E., Freund, Y., Barlett, P., Lee, W.S.: Boosting the margin: A new explanation for the effectiveness of voting methods. In: ICML, pp. 322–330 (1997)
14. Sheskin, D.J.: Handbook of parametric and nonparametric statistical procedures. Chapman & Hall, CRC (2000)
15. Vanderbei, R.J.: Linear Programming: Foundations and Extensions, 2nd edn. Springer, Heidelberg (2001)
16. Vezhnevets, A., Vezhnevets, V.: Modest adaboost - teaching adaboost to generalize better. In: Graphicon 2005 (2005)

Combination of Dichotomizers for Maximizing the Partial Area under the ROC Curve

Maria Teresa Ricamato and Francesco Tortorella

DAEIMI - Università degli Studi di Cassino
via G. Di Biasio 43, 03043 Cassino, Italy
{mt.ricamato,tortorella}@unicas.it

Abstract. In recent years, classifier combination has been of great interest for the pattern recognition community as a method to improve classification performance. The most part of combination rules are based on maximizing the accuracy and, only recently, the Area under the ROC curve (AUC) has been proposed as an alternative measure. However, there are several applications which focus only on particular regions of the ROC curve, i.e. the most relevant for the problem. In these cases, looking on a partial section of the AUC is the most suitable approach to adopt. In this paper we propose a new algorithm able to maximize only a part of the AUC by means of a linear combination of dichotomizers. Moreover, we empirically show that algorithms that maximize the AUC do not maximize the partial AUC, i.e., the two kinds of maximization are independent.

Keywords: Classifiers combination, ROC curve, partial AUC.

1 Introduction

Classifier combination has received considerable attention in the last years becoming an established technique for improving classification performance. In a classifier combination system, the output information of all the individual classifiers are combined in order to improve their performance. It has been proved that a successful combination rule exploits variations between individual classifiers, using their strengths to take advantages and to decrease their weaknesses. Among the various classifier combination methods previously proposed, linear classifier combination has been used mainly for its simplicity and good comprehensibility. In particular, there are some methods designed to increase the Area under the ROC curve (AUC), a more suitable performance measure than the classification accuracy [1], specially for those applications characterized by imprecise environment or imbalanced class priors [2]. AUC reduces an entire ROC curve to a single quantitative index showing classifier performance over all the false positive rate (FPR) values. However, there are many applications that are interested only to a particular range of FPRs. For example, in a biometric authentication system used to identify people, or to verify the claimed identity of registered users when entering in a protected area, a false positive is considered

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 660–669, 2010.

the most serious error, since it gives unauthorized users access to the systems that expressly are trying to keep them out. Another example is given by medical screening tests, where a false positive involves more expensive and sophisticated exams in order to be sorted out. In both cases, the FPR values considered are the ones that correspond to lower values, and the partial AUC [3] is the most indicate index to use, since it allows us to focus on particular regions of the ROC space. The partial AUC (pAUC) has already been used as a performance measure in applications for screening research [4] [5], but it has been given little attention to it as a performance measure in machine learning in order to build classification systems and to evaluate learning algorithms.

Our main purpose is to introduce the partial AUC measure and use it in the particular context of classifiers combination. Specifically, we propose a new algorithm able to find the weight vector in a linear combination of $K \geq 2$ dichotomizers, such that the pAUC is maximized.

The paper is organized as follow. The next section presents the pAUC index and its main properties. The proposed algorithm is analyzed in section 3 for a combination of two dichotomizers, and in section 4 for a combination of more than two dichotomizers. The performed experiments and obtained results are shown in section 5, while section 6 concludes the paper.

2 ROC Analysis and Partial Area under the ROC Curve

Receiver Operating Characteristics (ROC) graphs are useful for visualizing, organizing and selecting classifiers based on their performance. Given a two-class classification model, the ROC curve describes the trade-off between the fraction of correctly classified actually-positive cases (True Positive Rate, TPR) and the fraction of wrongly classified actually-negative cases (False Positive Rate, FPR), giving a description of the performance of the decision rule at different operating points.

In some cases, it is preferable to use the Area under the ROC Curve (AUC) [6] [7], a single metric able to summarize the performance of the classifiers system:

$$AUC = \int_0^1 ROC(t)dt \tag{1}$$

Remembering that some applications do not use all the range of false positive rates, it is worth to introduce another summary index that considers only those FPRs between t_0 and t_1, called partial AUC (pAUC), and defined as:

$$pAUC = \int_{t_0}^{t_1} ROC(t)dt \tag{2}$$

where the interval (t_0, t_1) denotes the false positive rates of interest. Its choice depends on the particular application, and it is related to the involved cost of a false positive diagnosis.

Moreover, the pAUC can be also defined as the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one, such that this latter belongs to the $1-t_k$ quantiles[1] range $\left(q_y^{t_1}, q_y^{t_0}\right)$:

$$pAUC = P\left\{x_i > y_j, y_j \in \left(q_y^{t_1}, q_y^{t_0}\right)\right\} \tag{3}$$

where $x_i = f(\mathbf{p}_i)$ and $y_j = f(\mathbf{n}_j)$ are the outcomes of the dichotomizer f on a positive sample $\mathbf{p}_i$ and a negative sample $\mathbf{n}_j$.

In order to evaluate the pAUC of a dichotomizer avoiding to perform a numerical integration on the ROC curve, we use the non-parametric estimator [3], which is defined as:

$$pAUC = \frac{1}{m_P m_N} \sum_i^{m_P} \sum_j^{m_N} V_{ij}^{q_y^{t_0}, q_y^{t_1}} \tag{4}$$

where m_P and m_N are the cardinalities of the positive and negative subsets, respectively, and

$$V_{ij}^{q_y^{t_0}, q_y^{t_1}} = I\{x_i > y_j, y_j \in \left(q_y^{t_1}, q_y^{t_0}\right)\} = \begin{cases} 1 & \text{if } x_i > y_j \bigwedge y_j \in \left(q_y^{t_1}, q_y^{t_0}\right); \\ 0.5 & \text{if } x_i = y_j \bigwedge y_j \in \left(q_y^{t_1}, q_y^{t_0}\right); \\ 0 & \text{if } x_i < y_j \bigwedge y_j \in \left(q_y^{t_1}, q_y^{t_0}\right). \end{cases} \tag{5}$$

Since the most part of biometric and medical applications [8] work on false positive rate close to the zero value, for the following analysis we consider $t_0 = 0$. In this case, equation 5 can be rewritten as $V_{ij}^{q_y^{t_1}} = I\{x_i > y_j, y_j > q_y^{t_1}\}$[2].

3 Linear Combination of Two Dichotomizers

Let us consider a set T of samples, and define the outputs of two generic dichotomizers f_h and f_k on two positive and negative samples $\mathbf{p}_i$ and $\mathbf{n}_j$:

$$x_i^h = f_h(\mathbf{p}_i), \qquad x_i^k = f_k(\mathbf{p}_i), \qquad y_j^h = f_h(\mathbf{n}_j), \qquad y_j^k = f_k(\mathbf{n}_j).$$

The pAUCs for the two dichotomizers, considering the FPR interval $(0, t_1)$, are:

$$pAUC_h = \frac{\sum_{i=1}^{m_P} \sum_{j=1}^{m_N} I\left(x_i^h > y_j^h, y_j^h > q_{y^h}^{t_1}\right)}{m_P m_N}, \quad pAUC_k = \frac{\sum_{i=1}^{m_P} \sum_{j=1}^{m_N} I\left(x_i^k > y_j^k, y_j^k > q_{y^k}^{t_1}\right)}{m_P m_N} \tag{6}$$

It is worth to note that finding the linear combination of two generic dichotomizers $f_{lc} = \alpha_h f_h + \alpha_k f_k$ such that maximizes the pAUC, is equivalent to find the weight $\alpha = \frac{\alpha_k}{\alpha_h} \in (-\infty, +\infty)$ which maximizes the pAUC for $f_{lc} = f_h + \alpha f_k$. Therefore, considering the linear combination, the outcomes on $\mathbf{p}_i$ and $\mathbf{n}_j$ are:

$$\xi_i = f_{\text{lc}}(\mathbf{p}_i) = x_i^h + \alpha x_i^k, \qquad \eta_j = f_{\text{lc}}(\mathbf{n}_j) = y_j^h + \alpha y_j^k. \tag{7}$$

[1] The quantile function returns the value below which random draws from the negative population would fall, $(1 - t_k) \times 100$ percent of the time.

[2] If $t_0 = 0$, then the $1 - t_0$ quantile $q_y^{t_0}$ is equal to $+\infty$.

and the pAUC is:

$$pAUC_{\mathrm{lc}} = \frac{1}{m_P m_N}\left(\sum_{i=1}^{m_P}\sum_{j=1}^{m_N} I\left(\xi_i > \eta_j, \left(\eta_j > q_\eta^{t_1}\left(\alpha\right)\right)\right)\right) \tag{8}$$

In order to find the value α_{opt} which maximizes $pAUC_{\mathrm{lc}}$, let us analyze the term $I(\xi_i > \eta_j)$ without considering the constraint on the quantile. In particular, let us remind from [9] how it depends on the values of $I(x_i^h, y_j^h)$ and $I(x_i^k, y_j^k)$:

- $I(x_i^h, y_j^h) = 1$ and $I(x_i^k, y_j^k) = 1$. In this case the two samples are correctly ranked by the two dichotomizers, and $I(\xi_i > \eta_j) = 1$.
- $I(x_i^h, y_j^h) = 0$ and $I(x_i^k, y_j^k) = 0$. In this case neither dichotomizer ranks correctly the samples and thus the contribution for the $pAUC$ is 0.
- $I(x_i^h, y_j^h) \,\mathrm{xor}\, I(x_i^k, y_j^k) = 1$. Only one dichotomizer ranks correctly the samples while the other one is wrong. In this case the value of $I(\xi_i > \eta_j)$ depends on the weight α.

The subset T can be divided into four subsets: T_{hk}, $T_{h\bar{k}}$, $T_{\bar{h}k}$ and $T_{\bar{h}\bar{k}}$ defined as:

$$\begin{aligned}
T_{hk} &= \{(\mathbf{p}_i, \mathbf{n}_j) | I(x_i^h, y_j^h) = 1 \text{ and } I(x_i^k, y_j^k) = 1\},\\
T_{\bar{h}k} &= \{(\mathbf{p}_i, \mathbf{n}_j) | I(x_i^h, y_j^h) = 0 \text{ and } I(x_i^k, y_j^k) = 1\},\\
T_{h\bar{k}} &= \{(\mathbf{p}_i, \mathbf{n}_j) | I(x_i^h, y_j^h) = 1 \text{ and } I(x_i^k, y_j^k) = 0\},\\
T_{\bar{h}\bar{k}} &= \{(\mathbf{p}_i, \mathbf{n}_j) | I(x_i^h, y_j^h) = 0 \text{ and } I(x_i^k, y_j^k) = 0\}
\end{aligned}$$

Now, let us consider the constraint on the negative samples related to the quantile, and define the following set:

$$\Gamma_\alpha = \{(\mathbf{p}_i, \mathbf{n}_j) \in P \times N | y_j^h + \alpha y_j^k > q_\eta^{t_1}\} \tag{9}$$

where $q_\eta^{t_1}$ is the $1 - t_1$ of η, which depends on the weight α. If we define the sets $T'_{hk}, T'_{\bar{h}k}, T'_{h\bar{k}}, T'_{\bar{h}\bar{k}}$ as:

$$\begin{aligned}
T'_{hk} &= T_{hk} \cap \Gamma_\alpha, \quad T'_{\bar{h}k} = T_{\bar{h}k} \cap \Gamma_\alpha,\\
T'_{h\bar{k}} &= T_{h\bar{k}} \cap \Gamma_\alpha, \quad T'_{\bar{h}\bar{k}} = T_{\bar{h}\bar{k}} \cap \Gamma_\alpha,
\end{aligned}$$

the expression for $pAUC_{\mathrm{lc}}$ in equation 8 can be written as:

$$pAUC_{\mathrm{lc}} = \frac{1}{m_P m_N}\left(\sum_{(\mathbf{p}_i,\mathbf{n}_j)\in T'_{\bar{h}\bar{k}}} I(\xi_i > \eta_j) + \sum_{(\mathbf{p}_i,\mathbf{n}_j)\in T'_{hk}} I(\xi_i > \eta_j)\right.$$
$$\left. + \sum_{(\mathbf{p}_i,\mathbf{n}_j)\in T'_{h\bar{k}} \cup T'_{\bar{h}k}} I(\xi_i > \eta_j)\right) = \frac{0 + \gamma(\alpha) + \nu(\alpha)}{m_P m_N}.$$

and the optimal weight is given by:

$$\alpha_{\mathrm{opt}} = \arg\max_\alpha \left(\gamma(\alpha) + \nu(\alpha)\right). \tag{10}$$

In order to find α_{opt}, let us recall that $I(\xi_i > \eta_j) = 1$ only if:

$$\left(x_i^h - y_j^h\right) + \alpha \left(x_i^k - y_j^k\right) > 0 \tag{11}$$

such that: $y_j^h + \alpha y_j^k > q_\eta^{t_1}(\alpha)$. If we define $\Delta_{ij}^h = x_i^h - y_j^h$ and $\Delta_{ij}^k = x_i^k - y_j^k$ and considering the three subsets $T'_{hk}, T'_{h\bar{k}}, T'_{\bar{h}k}$, we obtain three different constraints:

$$\alpha < -\frac{\Delta_{ij}^h}{\Delta_{ij}^k} \text{ if } (\mathbf{p}_i, \mathbf{n}_j) \in T'_{h\bar{k}}, \quad \alpha > -\frac{\Delta_{ij}^h}{\Delta_{ij}^k} \text{ if } (\mathbf{p}_i, \mathbf{n}_j) \in T'_{\bar{h}k}, \quad \alpha > -\frac{\Delta_{ij}^h}{\Delta_{ij}^k} \text{ if } (\mathbf{p}_i, \mathbf{n}_j) \in T'_{hk}.$$

The pAUC is maximized when the number of pairs satisfying the previous constraints is maximized. Introducing the cumulative functions as follow:

$$F'_{h\bar{k}}(\alpha) = \text{card}\left((\mathbf{p}_i, \mathbf{n}_j) \in T'_{h\bar{k}} \middle| -\frac{\Delta_{ij}^h}{\Delta_{ij}^k} > \alpha\right)$$

$$F'_{\bar{h}k}(\alpha) = \text{card}\left((\mathbf{p}_i, \mathbf{n}_j) \in T'_{\bar{h}k} \middle| -\frac{\Delta_{ij}^h}{\Delta_{ij}^k} < \alpha\right)$$

$$F'_{hk}(\alpha) = \text{card}\left((\mathbf{p}_i, \mathbf{n}_j) \in T'_{hk} \middle| -\frac{\Delta_{ij}^h}{\Delta_{ij}^k} < \alpha\right)$$

then, the function to be maximized is defined as:

$$\gamma(\alpha) + \nu(\alpha) = F'_{h\bar{k}}(\alpha) + F'_{\bar{h}k}(\alpha) + F'_{hk}(\alpha) \tag{12}$$

It is worth to note, from the previous analysis, that $F'_{h\bar{k}}$ and $F'_{\bar{h}k}$ depend on the interaction between the pair values and α, while F'_{hk} only depends on α, due to the quantile values.

The optimal values can be easily found by means of linear search:

$$\alpha_{\text{opt}} = \arg\max_{\alpha} \left(F'_{h\bar{k}}(\alpha) + F'_{\bar{h}k}(\alpha) + F'_{hk}(\alpha)\right) \tag{13}$$

4 Linear Combination of $K > 2$ Dichotomizers

The linear combination of $K > 2$ dichotomizers is defined as:

$$f_{lc}(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \ldots + \alpha_K f_K(x) = \sum_{i=1}^{K} \alpha_i f_i(x) \tag{14}$$

In order to find the optimal weight vector $\alpha_{opt} = (\alpha_1, \ldots, \alpha_K)$ that maximizes the pAUC associated to $f_{lc}(x)$, the method described in the previous section cannot be generalized in a computational feasible algorithm. Therefore, the proposed algorithm is based on the approximation of the solution by dividing the whole K-combination problem into a series of feasible pairwise combination problems using the greedy approach.

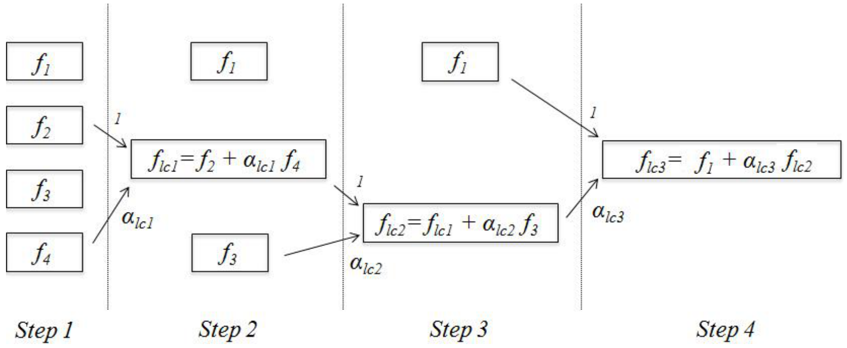


Fig. 1. Example of greedy approach steps

Table 1. XM2VTS database properties

	# Sample	# Positive	# Negative
Validation Set	40600	600	40000
Test Set	112200	400	111800

Table 2. pAUC of each classifier, calculated in step 1 and step 2 in fig. 1

(a) Step 1

	f_1	f_2	f_3	f_4
pAUC	0.093	0.095	0.094	0.096

(b) Step 2

	f_{lc1}	f_1	f_3
pAUC	0.097	0.093	0.094

The greedy approach is a suboptimal method which uses $K-1$ steps of a simpler algorithm, obtaining the optimum weights only after the $K-1$ steps. In particular, in each step two dichotomizers are combined using the algorithm described in section 3, finding the optimal weight for that combination. After the first weight is computed, the number of dichotomizers decreases from K to $K-1$. After that, there is the choice of the two dichotomizers for the next step. This procedure is repeated until all the dichotomizers have been combined.

Using the greedy approach is equivalent to find a suboptimal solution by making a locally optimal choice. Therefore, in each iteration, the choice of two dichotomizers that should be combined, plays an important role. In fact it is a fundamental issue that can affect the performance of the algorithm.

In the proposed method, we consider an approach based on the best performance of the individual dichotomizer in terms of pAUC. Therefore, at each step, the algorithm chooses the two dichotomizers with the maximum pAUC values.

Figure 1 and table 2 show the steps of the algorithm considering the interval $FPR = (0, 0.1)$, and assuming the linear combination of 4 classifiers. First of all, for each classifier the pAUC value is computed (tab. 2(a)). Then, the pair that corresponds to the two classifiers with higher pAUCs, in the example f_2 and f_4, is used to find the α_{lc1} for the first linear combination. The vector of pAUCs is updated (tab. 2(b)), considering the new classifier f_{lc1}. Then, these steps are repeated until there are no more classifiers to be combined.

In order to recover the weight for each of the classifiers, a combination tree is built during the evaluation of the α_{opt} (fig. 1). The original classifiers constitute the leaves of the tree and, each time a pair of classifiers is combined, a parent node is added associated to the two combined classifiers. The edges are labeled with the weights assigned to each classifier. At the end of the computation, the weight of each classifier can be easily recovered by traversing the tree from the leaf up to the root and multiplying all the values found on the edges. In the example shown in figure 1, the final combination of classifiers is:

$$\begin{aligned} f_{lc3} = f_1 + \alpha_{lc3} f_{lc2} &= f_1 + \alpha_{lc3}(f_{lc1} + \alpha_{lc2} f_3) \\ = f_1 + \alpha_{lc3}(f_2 + \alpha_{lc1} f_4 + \alpha_{lc2} f_3) &= f_1 + \alpha_{lc3} f_2 + \alpha_{lc3}\alpha_{lc1} f_4 + \alpha_{lc3}\alpha_{lc2} f_3 \end{aligned}$$

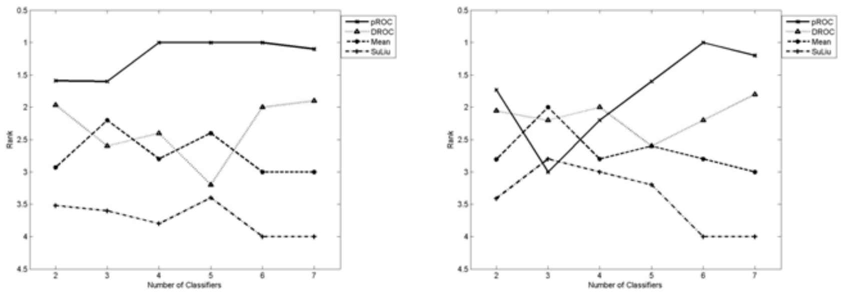
and the final weight vector is given by: $\alpha_{opt} = (1 \quad \alpha_{lc3} \quad \alpha_{lc2}\alpha_{lc3} \quad \alpha_{lc1}\alpha_{lc3})$.

5 Experimental Results

In order to evaluate the performance of the pROC algorithm proposed, the experiments are performed on the public-domain biometric dataset XM2VTS [10], characterized by 8 matchers, using the partition of the scores into training and test set proposed in [10] and showed in table 1. The XM2VTS is a multimodal database containing video sequences and speech data of 295 subjects recorded in four sessions in a period of 1 month. In order to assess its performance the Lausanne protocol has been used to randomly divide all the subjects into positive and negative classes: 200 positive, 25 evaluation negatives and 70 test negatives. All the details about the procedure used to obtain the final dichotomizers are described in [10].

The combination rule proposed (pROC) is compared with other algorithms: a method proposed by Su and Liu in [11], that provides a linear combination to maximize the AUC assuming a normal distribution for both positive and negative samples, and the DROC method proposed in [9] that maximizes the AUC considering the pairs of positive and negative samples that contribute to the AUC values without any assumption on their distribution. It is worth to note that both of the algorithms considered for the comparison evaluate a weight vector such that the combination maximizes the AUC. Furthermore, another considered combination rule is the average of the outcomes of the dichotomizers which is independent from any direct maximization of the metric and from any distribution.

For each considered method the vector α of coefficients for the linear combination is evaluated on the validation set, and then applied to the test set. The results are analyzed in term of partial AUC, considering the false positive ranges: $FPR_{0.1} = (0, 0.1)$, $FPR_{0.05} = (0, 0.05)$ and $FPR_{0.01} = (0, 0.01)$.



(a) (b)

Fig. 2. Mean of rank on validation set (a) and test set (b), with $FPR_{0.1}$. Note that in both figures the y-axis is reversed.

The number of combined dichotomizers varies from 2 to 7. For each of those experiments we obtain different number of possible combinations that are independent from each other. Therefore, we use an approach based on giving a rank to each method compared to the others, for each independent experiment. Let us consider the pAUC values $\{pAUC_{ij}\}_{M \times L}$, for $i = 1, \ldots, M$ with M the number of combinations, and for $j = 1, \ldots, L$ with L number of combination rules that are compared. For each row we assign a rank value r_j^i from 1 to L to each column depending on the pAUC values: the highest pAUC gets rank 1, the second highest the rank 2, and so on until L (in our case $L = 4$). If there are tied pAUCs, the average of the ranks involved is assigned to all pAUCs tied for a given rank. Only in this case it is appropriate to average the obtained ranks on the number of combinations:

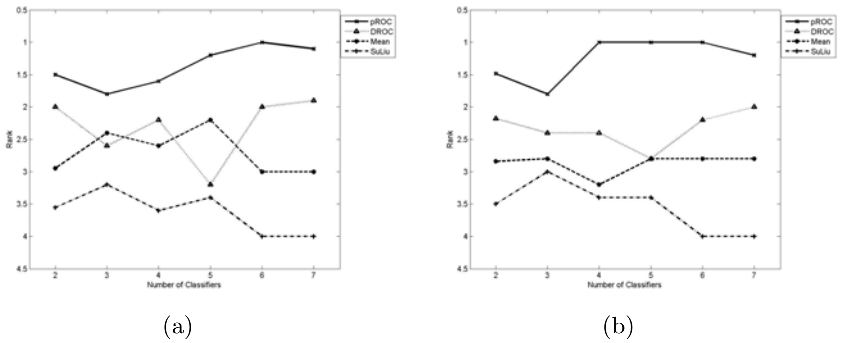
$$\bar{r}_j = \frac{1}{M} \sum_{i=1}^{M} r_j^i \tag{15}$$

Figures 2(a)-4(a) and figures 2(b)-4(b) show the results on the validation set and test set, respectively, varying the FPR ranges. The higher the curve, i.e. the lower the value, the better the related method.

Analyzing the results, we can observe a very good generalization of the algorithm pROC except for $FPR_{0.1}$, where its behavior is comparable with the one of the other methods. Notwithstanding that, the algorithm performs well on the most part of the experiments. Decreasing the FPR range, pROC performance are much better since the algorithm is more adapt to the problem.

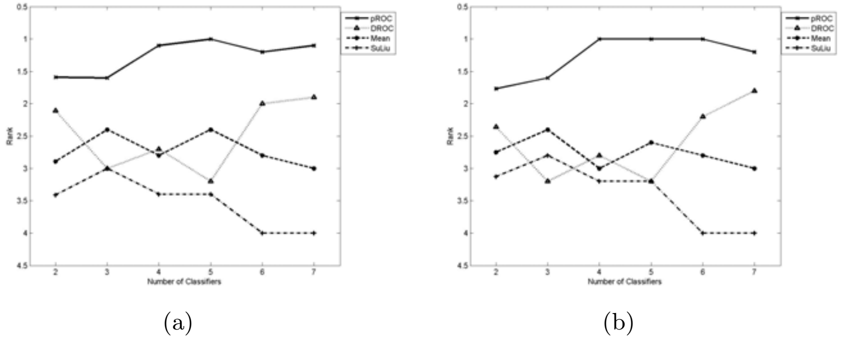
Moreover, it is shown the difference between the two kinds of maximization: AUC-based and pAUC-based. In particular, it is worth to note that methods designed to maximize the AUC (DROC and SuLiu) do not maximize the pAUC. In fact, DROC and SuLiu maximize the performance measure considering all the range of FPR, while pROC considers only a particular range of it.

Furthermore, assuming a normal distribution of negative and positive samples in SuLiu, does not perform as good as the average rule, and as DROC



(a) (b)

Fig. 3. Mean of rank on validation set (a) and test set (b), with $FPR_{0.05}$



(a) (b)

Fig. 4. Mean of rank on validation set (a) and test set (b), with $FPR_{0.01}$

method which is independent from any distributions. Such difference is due to the fact that the computation of the weight is affected more when the assumed distribution model is not close to the real one. In addition to the fact that the AUC-based method SuLiu does not maximize the pAUC, it has also less performance than a method that is independent from any kind of maximization (the average method).

6 Conclusions

In this paper, we have proposed a new linear combination method aims to improve the partial Area under the ROC curve (pAUC) in a two-class classification problem, since little attention has been given to the use of pAUC in machine learning and specifically as a performance measure in combining classifiers.

The algorithm designed to maximize the pAUC is based on the dependence of the metric on the coefficients vector α used for the linear combination of dichotomizers. The algorithm has been implemented for a two dichotomizers combination, then extended to the combination of $K > 2$ dichotomizers.

The results obtained have shown good performance of pROC method compared with other algorithms. In particular, it has been noticed that maximizing the total AUC is not so effective for the maximization of the partial AUC, in fact maximizing the metric on all the range of FPR is not equivalent to maximize the metric in a portion on that range. Moreover, methods that assume a particular distribution model for negative and positive samples, are not able to perform as good as method that do not have any assumption.

Future work regards an analysis on the possible rules that can be used in the greedy approach in order to choose the dichotomizers to combine at each step. It will be interesting to note if the performance will change and how.

References

1. Huang, J., Ling, C.X.: Using AUC and accuracy in evaluating learning algorithms. IEEE Trans. on Knowledge and Data Engineering 17, 299–310 (2005)
2. Cortes, C., Mohri, M.: AUC optimization vs. error rate minimization advances. In: Neural Information Processing Systems. MIT Press, Cambridge (2003)
3. Dodd, L.E., Pepe, M.S.: Partial AUC estimation and regression. Biometrics 59, 614–623 (2003)
4. Jiang, Y., Metz, C.E., Nishikawa, R.M.: A receiver operating characteristic partial area index for highly sensitive diagnostic tests. Radiology 201, 745–750 (1996)
5. McClish, D.K.: Analyzing a portion of the ROC curve. Medical Decision Making 9, 190–195 (1989)
6. Bradley, A.P.: The use of the area under the roc curve in the evaluation of machine learning algorithms. Patt. Recogn. 30, 1145–1159 (1997)
7. Hanley, J.A., McNeil, B.J.: The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143, 29–36 (1982)
8. Nandakumar, K., Dass, S.C., Jain, A.K.: Likelihood ratio-based biometric score fusion. IEEE Trans. on Patt. Anal. and Mach. Intell. 30, 342–347 (2008)
9. Marrocco, C., Molinara, M., Tortorella, F.: Optimal linear combination of dichotomizers via auc. In: Proceed. of the 22nd Intern. Conf. on Mach. Learn. - Workshop on ROC Analysis in Mach. Learn., pp. 778–785 (2005)
10. Poh, N., Bengio, S.: Database, protocol and tools for evaluating score-level fusion algorithms in biometric authentication. Patt. Recogn. 39, 223–233 (2006)
11. Su, J.Q., Liu, J.S.: Linear combinations of multiple diagnostic markers. Journal of the Americ. Stat. Assoc. 88, 1350–1355 (1993)

Ihara Coefficients: A Flexible Tool for Higher Order Learning

Peng Ren[1], Tatjana Aleksić[2], Richard C. Wilson[1], and Edwin R. Hancock[1]

[1] Department of Computer Science, The University of York,
York, YO10 5DD, UK
{pengren,wilson,erh}@cs.york.ac.uk
[2] University of Kragujevac, Faculty of Science, 34000 Kragujevac, Serbia
taleksic@kg.ac.rs

Abstract. The aim of this paper is to seek a compact characterization of irregular unweighted hypergraphs for the purposes of clustering. To this end, we propose a novel hypergraph characterization method by using the Ihara coefficients, i.e. the characteristic polynomial coefficients extracted from the Ihara zeta function. We investigate the flexibility of the Ihara coefficients for learning relational structures with different relational orders. Furthermore, we introduce an efficient method for computing the coefficients. Our representation for hypergraphs takes into account not only the vertex connections but also the hyperedge cardinalities, and thus can distinguish different relational orders, which is prone to ambiguity in the hypergraph Laplacian. In experiments we demonstrate the effectiveness of the proposed characterization for clustering irregular unweighted hypergraphs and its advantages over the spectral characterization of the hypergraph Laplacian.

1 Introduction

Hypergraph-based methods have recently been widely used for representing and processing relational structures where the relations present are not simply pairwise. Specific applications of hypergraph related methods in visual processing include the algorithms described in [3][6]. One common feature of these methods is that they exploit domain specific and goal directed representations, and do not lend themselves to generalization. The reason for this lies in the difficulty in formulating a hypergraph in a mathematically uniform way for computation. However, to be easily manipulated, hypergraphs must be represented in a mathematically consistent way, using structures such as matrices or vectors. One possible method for establishing hypergraph matrix representations is to transform a hypergraph into a graph and then use the associated graph adjacency matrix or Laplacian matrix as the matrix representation of the hypergraph. Agarwal *et al.* [1] have made a review of the possible graph representations for a hypergraph and revealed their relationships with each other in machine learning. Each of these methods assume that there is a weight attached to each hyperedge. The edges in the graph representation are weighted in a manner determined by the corresponding hyperedge weights. As far as unweighted hypergraphs are concerned, the literature mainly focuses on using tensor representations [7][9]. The tensor representations consider all possible permutations of a subset of vertices and establish hyperedges with cardinality

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 670–679, 2010.

consistent with the relational order. Therefore, tensors can only represent regular hypergraphs, and are not suited for irregular hypergraphs. Ren *et al.* [5] have proposed an improved hypergraph Laplacian based on developments of Zhou *et al.*'s method [10] and apply it to clustering hypergraphs. Although this method is suitable for unweighted irregular hypergraphs, it is based on a relatively impoverished spectral characterization and overlooks much of the detail of hypergraph-structure. Recently, Ren *et al.* [4] have attempted to represent hypergraphs using characteristics from the Ihara zeta function. However, this work does not indicate in which cases the characteristics are superior to spectral methods, neither does it investigate the flexibility of these features.

In this paper, we characterize irregular unweighted hypergraphs using Ihara coefficients. The proposed hypergraph representation proves to be a flexible tool in learning the structure of irregular unweighted hypergraphs with different relational orders. Our contributions are two-fold. First, we propose a vectorial representation, which naturally avoids the ambiguity induced by the matrix representations such as the hypergraph Laplacian, for irregular unweighted hypergraphs. We construct pattern vectors using the Ihara coefficients, i.e. the characteristic polynomial coefficients extracted from Ihara zeta function for hypergraphs. Second and more importantly, we propose an efficient method for computing the Ihara coefficient set, which renders the computation of the coefficients tractable. We use the pattern vectors consisting of Ihara coefficients for clustering hypergraphs extracted from images of different object views and demonstrate their effectiveness in hypergraph characterization.

2 Hypergraph Laplacian

A hypergraph is a generalization of a graph. Unlike the edge of a graph, which can connect only two vertices, the hyperedge in a hypergraph can connect any number of vertices. A hypergraph is normally defined as a pair $H(V, E_H)$ where V is a set of elements, called nodes or vertices, and E_H is a set of non-empty subsets of V called hyperedges. The representation of a hypergraph in the form of sets, concretely captures the relationship between vertices and hyperedges. However, it is difficult to manipulate this form in a computationally uniform way. Thus one alternative representation of a hypergraph is in the form of a matrix. For a hypergraph $H(V, E_H)$ with I vertices and J hyperedges, we establish an $I \times J$ matrix $\boldsymbol{H}$ which is referred to as the incidence matrix of the hypergraph. $\boldsymbol{H}$ has element $h_{i,j}$ 1 if $v_i \in e_j$ and 0 otherwise.

The incidence matrix can be more easily manipulated than its equivalent set representation. To obtain a vertex-to-vertex representation, we need to establish the adjacency matrix and Laplacian matrix for a hypergraph. To this end, a graph representation for the hypergraph is required. Agarwal *et al.* [1] have classified the graph representations for a hypergraph into two categories, namely a) the clique expansion and b) the star expansion. The clique expansion represents a hypergraph by constructing a graph with all the pairs of vertices within a hyperedge connecting each other. The star expansion represents a hypergraph by introducing a new vertex to every hyperedge and constructing a graph with all vertices within a hyperedge connecting the newly introduced vertex. The common feature of these methods is that each edge in a graph representation is weighted in terms of the corresponding hyperedge weight subject to certain conditions.

For example, the normalized Laplacian matrix $\hat{\boldsymbol{L}}_H = \boldsymbol{I} - \boldsymbol{D}_v^{-1/2}\boldsymbol{H}\boldsymbol{D}_e\boldsymbol{H}^T\boldsymbol{D}_v^{-1/2}$ introduced in [10] is obtained from the star expansion of a hypergraph, and its individual edges are weighted by the quotient of the corresponding hyperedge weight and cardinality. Here $\boldsymbol{D}_v$ is the diagonal vertex degree matrix whose diagonal element $d(v_i)$ is the summation of the ith row of $\boldsymbol{H}$, $\boldsymbol{D}_e$ is the diagonal vertex degree matrix whose diagonal element $d(e_j)$ is the summation of the jth column of $\boldsymbol{H}$, and $\boldsymbol{I}$ is a $|V| \times |V|$ identity matrix. In this case, even edges derived from an unweighted hyperedge are assigned a nonunit weight. On the other hand, rather than attaching a weight to each edge in the graph representation, the adjacency matrix and the associated Laplacian matrix for an irregular unweighted hypergraph can be defined as $\boldsymbol{A}_H = \boldsymbol{H}\boldsymbol{H}^T - \boldsymbol{D}_v$ and $\boldsymbol{L}_H = \boldsymbol{D}_v - \boldsymbol{A}_H = 2\boldsymbol{D}_v - \boldsymbol{H}\boldsymbol{H}^T$ respectively [5]. In practice, these two definitions are obtained in terms of the clique expansion without attaching a weight to a graph edge. The eigenvalues of $\boldsymbol{L}_H$ are referred to as the hypergraph Laplacian spectrum and can be used in a straightforward way as hypergraph characteristics.

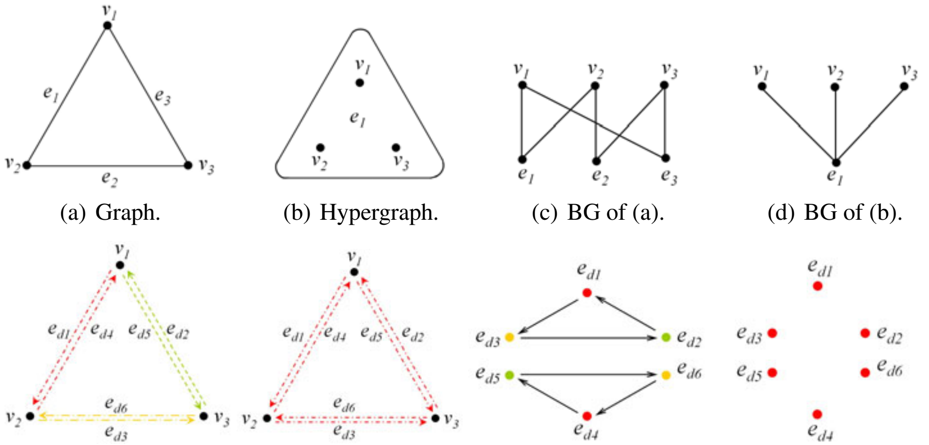
Although the vertex-to-vertex matrix representations for hypergraphs described above naturally reduce to those for graphs when the relational order is two, there are deficiencies for these representations in distinguishing relational structures. When relational structures have the same vertex cardinality but different relational orders, these vertex-to-vertex matrix representations become ambiguous. For example, for the graph in Fig. 1(a) and the hypergraph in Fig. 1(b), the adjacency matrices of the two hypergraphs are identical, and so are the associated Laplacian matrices. The adjacency matrix and Laplacian matrix are as follows:

$$\boldsymbol{A}_H = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \qquad \boldsymbol{L}_H = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}$$

It is clear that the unweighted adjacency matrix and Laplacian matrix can not distinguish these two hypergraphs. The reason for this deficiency is that the adjacency matrix and the Laplacian matrix only record the adjacency relationships between pairs of nodes and neglect the cardinalities of the hyperedges. In this regard they induce certain information loss in representing relational structures and can not always distinguish between pairwise relationships and high order relationships for the same set of vertices. The normalized Laplacian matrix for Fig. 1(a) and 1(b) are $\hat{\boldsymbol{L}}_{H1}$ and $\hat{\boldsymbol{L}}_{H2}$ respectively.

$$\hat{\boldsymbol{L}}_{H1} = \begin{pmatrix} 2/3 & -1/3 & -1/3 \\ -1/3 & 2/3 & -1/3 \\ -1/3 & -1/3 & 2/3 \end{pmatrix} \qquad \hat{\boldsymbol{L}}_{H2} = \begin{pmatrix} 1/2 & -1/4 & -1/4 \\ -1/4 & 1/2 & -1/4 \\ -1/4 & -1/4 & 1/2 \end{pmatrix}$$

Since $\hat{\boldsymbol{L}}_{H2} = \frac{3}{4}\hat{\boldsymbol{L}}_{H1}$, the eigenvalues of $\hat{\boldsymbol{L}}_{H2}$ are found by scaling those of $\hat{\boldsymbol{L}}_{H1}$ by a factor $3/4$, and both matrices have the same eigenvectors. Thus the normalized Laplacian matrices for different hypergraphs may yield spectra that are just scaled relative to each other. This hinders the hypergraph characterization when the eigenvectors are used. One important reason for the limited usefulness of the above hypergraph matrix representations is that they result in information loss when relational orders of varying degree are present. To overcome this deficiency, we use characteristic polynomials extracted from the Ihara zeta function as a means of representing hypergraphs. In the next



(a) Graph. (b) Hypergraph. (c) BG of (a). (d) BG of (b).

(e) Digraph of (a). (f) Digraph of (b). (g) OLG of (a). (h) OLG of (b).

Fig. 1. Hypergraph examples and their graph representations

section, we commence by showing that the Ihara zeta function can be used to represent this type of relational structure in hypergraphs. We use the Ihara coefficients, i.e. the characteristic polynomial coefficients extracted from the Ihara zeta function, as hypergraph characteristics. We show that the Ihara coefficients not only encode the relational structural in a consistent way but also overcome the deficiencies listed above.

3 Ihara Zeta Function from Graphs to Hypergraphs

The rational expression of the Ihara zeta function for a graph is as follows [2]:

$$Z_G(u) = \left(1-u^2\right)^{\chi(G)} \det\left(\boldsymbol{I}_{|V(G)|} - u\boldsymbol{A} + u^2\boldsymbol{Q}\right)^{-1}, \tag{1}$$

where $\chi(G) = |V| - |E|$, $\boldsymbol{A}$ is the adjacency matrix of the graph, and $\boldsymbol{Q} = \boldsymbol{D} - \boldsymbol{I}_{|V(G)|}$ where $\boldsymbol{I}_{|V(G)|}$ is the identity matrix and $\boldsymbol{D}$ is the degree matrix, which can be generated by placing the column sums as the diagonal elements while setting the off-diagonal elements to zero.

To formulate the Ihara zeta function for a hypergraph in a similar form with (1), the bipartite graph representation of the hypergraph is needed. To this end, we use a dual representation in which each hyperedge is represented by a new vertex. The new vertex is incident to each of the original vertices in the corresponding hyperedge. The union of the new vertex set and the original vertex set constitute the vertex set of the associated bipartite graph. The new vertices corresponding to hyperedges are on one side and the original hypergraph vertices on the other side. Thus the bipartite graph and star expansion for a hypergraph share the same form, although they are defined for different purposes. For instance, the bipartite graphs associated with the example hypergraphs in Figs. 1(a) and 1(b) are shown in Figs. 1(c) and 1(d) respectively (BG stands for bipartite graph).

The Ihara zeta function of the hypergraph $H(V, E_H)$ can be expressed in a rational form as follows:

$$\zeta_H(u) = (1-u)^{\chi(BG)} \det\left(\boldsymbol{I}_{|V(H)|+|E_H(H)|} - \sqrt{u}\boldsymbol{A}_{BG} + u\boldsymbol{Q}_{BG}\right)^{-1}, \qquad (2)$$

where $\chi(BG)$is the Euler number of the associated bipartite graph, $\boldsymbol{A}_{BG}$ is the adjacency matrix of the associated bipartite graph, and $\boldsymbol{Q}_{BG} = \boldsymbol{D}_{BG} - \boldsymbol{I}_{|V(H)|+|E_H(H)|}$. Further details on the arguments leading from (1) to (2) can be found in [8].

The adjacency matrix of the associated bipartite graph can be formulated using the incidence matrix $\boldsymbol{H}$ of $H(V, E_H)$:

$$\boldsymbol{A}_{BG} = \begin{bmatrix} \boldsymbol{0}_{|E_H(H)|\times|E_H(H)|} & \boldsymbol{H}^T \\ \boldsymbol{H} & \boldsymbol{0}_{|V(H)|\times|V(H)|} \end{bmatrix}. \qquad (3)$$

The hypergraph Ihara zeta function in the form of (2) provides an alternative method for the function value computation, as well as an efficient method of computing the Ihara coefficients, which will be discussed later on in Section 5.

4 Determinant Expression for Hypergraph Zeta Function

Although the Ihara zeta function can be evaluated efficiently using (2), the task of enumerating the coefficients of the polynomial appearing in the denominator of the Ihara zeta function is difficult, except by resorting to software for symbolic calculation. To efficiently compute these coefficients, a different strategy is adopted. The hypergraph is first transformed into an oriented line graph. The Ihara zeta function is then the reciprocal of the characteristic polynomial for the adjacency matrix of the oriented line graph. Our novel contribution here is to use the existing ideas from hypergraph theory to develop a new hypergraph representation, which can be used in machine learning to distinguishing hypergraphs with the same vertex set but different relational orders.

4.1 Oriented Line Graph

To establish the oriented line graph associated with the hypergraph $H(V, E_H)$, we commence by constructing a $|e_i|$-clique, i.e. clique expansion, by connecting each pair of vertices in the hyperedge $e_i \in E_H$ through an edge. The resulting clique expansion graph is denoted by $GH(V, E_G)$. For $GH(V, E_G)$, the associated symmetric digraph $DGH(V, E_d)$ can be obtained by replacing each edge of $GH(V, E_G)$ by an arc (oriented edge) pair in which the two arcs are inverse to each other. For the example hypergraphs in Figs. 1(a) and 1(b), their $DGH(V, E_d)$ are shown in Figs. 1(e) and 1(f) respectively, where the oriented edges derived from the same hyperedge are colored the same while from different hyperedges are colored differently. Finally, the oriented line graph of the hypergraph can be established based on the symmetric digraph. The vertex set V_{ol} and edge set E_{ol} of the the oriented line graph are defined as follows [8]:

$$V_{ol} = E_d(DGH); \quad E_{ol} = \{(e_d(u,v), e_d(v,w)) \in E_d \times E_d \;;\; u, w \not\subset E_H\}. \qquad (4)$$

One observation that needs to be made here is that the adjacency matrix $\boldsymbol{A}_H$ and Laplacian matrix $\boldsymbol{L}_H$ for a hypergraph introduced in Section 2 are actually those of the graph established on the clique expansion, but without an edge-weight attachment. These matrix representations can induce ambiguity when representing relational structures with different relational orders. This point is illustrated by the two example hypergraphs in Figs. 1(a) and 1(b) which have the same clique graph and thus the same adjacency matrix and Laplacian matrix. The reason for this is that the clique expansion only records adjacency relationships between pairs of nodes and can not distinguish whether or not two edges in the clique are derived from the same hyperedge. Thus the clique graph representations for hypergraph result in loss of information concerning relational order. However, the Ihara zeta function overcomes this deficiency by avoiding the interaction between two edges derived from the same hyperedge. This is due to the constraint in (4) that the connecting oriented edge pair in the same clique of DGH can not establish an oriented edge in the oriented line graph. According to these properties, the example hypergraphs with the same adjacency matrix and Laplacian matrix in Figs. 1(a) and 1(b) produce oriented line graphs with totally different structures as shown in Figs. 1(g) and 1(h) respectively (OLG stands for oriented line graph), where the constraint in (4) prevents connections between any nodes with the same color in Figs. 1(g) and 1(h). The adjacency matrix $\boldsymbol{T}_H$ of the oriented line graph is the Perron-Frobenius operator of the original hypergraph. For the (i, j)th entry of $\boldsymbol{T}_H$, $\boldsymbol{T}_H(i, j)$ is 1 if there is one edge directed from the vertex with label i to the vertex with label j in the oriented line graph, otherwise it is 0. Unlike the adjacency matrix of an undirected graph, the Perron-Frobenius operator for a hypergraph is not a symmetric matrix. This is because of the constraint described above that arises in the construction of oriented edges. Specifically, it is the fact that the arc pair with two arcs that are derived from the same hyperedge in the original hypergraph is not allowed to establish an oriented edge in the oriented line graph that causes the asymmetry of $\boldsymbol{T}_H$.

4.2 Characteristic Polynomial

With the oriented line graph to hand, the Ihara zeta function for a hypergraph can be written in the form of a determinant using the Perron-Frobenius operator [8]:

$$\zeta_H(u) = \det(\boldsymbol{I}_H - u\boldsymbol{T}_H)^{-1} = (c_0 + c_1 u + \cdots + c_{M-1}u^{M-1} + c_M u^M)^{-1}, \quad (5)$$

where M is the highest order of the polynomial. The polynomial coefficients $c_0, c_2, \ldots, c_M$ are referred to as the Ihara coefficients. From (5) we can see that M is the dimensionality of the square matrix $\boldsymbol{T}_H$. To establish pattern vectors from the hypergraph Ihara zeta function for the purposes of characterizing hypergraphs in machine learning, it is natural to consider taking function samples as the elements. Although the function values at most of the sampling points will perform well in distinguishing hypergraphs, there is the possibility of sampling at poles giving rise to meaningless infinities. Hence, the pattern vectors consisting of function samples are potentially unstable representations of hypergraphs, since the distribution of poles is unknown beforehand. The characteristic polynomial coefficients, i.e. the Ihara coefficients, do not give rise to infinities. From (5), it is clear that each coefficient can be derived from the elementary symmetric polynomials of the eigenvalue set $\{\lambda_1, \lambda_2, \lambda_3 \ \dots\}$ of $\boldsymbol{T}_H$ as $c_r = (-1)^r \sum_{k_1<k_2< \dots <k_r} \lambda_{k_1} \lambda_{k_2} \dots \lambda_{k_r}$.

Furthermore, the Ihara coefficients relate strongly to the hypergraph-structure since the Ihara zeta function records information about prime cycles in the hypergraphs. We can construct pattern vectors using a dominant subset of the Ihara coefficients $\boldsymbol{v} = [c_{r1} \ c_{r2} \ \dots \ c_{rN}]^T$ for a hypergraph and then apply them to clustering hypergraphs.

5 Numerical Computation

The formation of $\boldsymbol{T}_H$ and its eigen-decomposition tend to be computationally expensive for practical problems, because the matrix $\boldsymbol{T}_H$ are usually of big size. To overcome the deficiency of computing the Ihara coefficients using (5), we develop a straightforward yet efficient method which starts from the associated bipartite graph. Instead of constructing the oriented line graph for a hypergraph, we establish the oriented line graph for the bipartite graph. Considering the rational expression (2) based on the associated bipartite graph, we have:

$$\zeta_H^{-1}(u) = Z_{BG}^{-1}(\sqrt{u}) = \det(\boldsymbol{I}_{BG} - \sqrt{u}\boldsymbol{T}_{BG}), \tag{6}$$

where $\boldsymbol{T}_{BG}$ is the Perron-Frobenius operator of the associated bipartite graph, of which the Ihara zeta function (according to its original definition [2]) is represented as:

$$Z_{BG}^{-1}(u) = \prod_{p \in P_{BG}} \left(1 - u^{|p|}\right) = \left(1 - u^{|p_1|}\right)\left(1 - u^{|p_2|}\right)\left(1 - u^{|p_3|}\right) \cdots . \tag{7}$$

where p_i is the ith prime cycle in the set P_{BG} of prime cycle equivalence classes of the bipartite graph. Note that every cycle in a bipartite graph has an even length, i.e. $|p_i|$ is always an even number for a bipartite graph. Let $\{\tilde{c}_0, \tilde{c}_1, \tilde{c}_2, \tilde{c}_3, \tilde{c}_4, \tilde{c}_5, \tilde{c}_6 \dots\}$ denote the Ihara coefficient set of the bipartite graph. It is clear that $Z_{BG}^{-1}(u)$ is a polynomial with the odd coefficients equal to zeros:

$$\begin{aligned} Z_{BG}^{-1}(u) &= \det(\boldsymbol{I}_{BG} - u\boldsymbol{T}_{BG}) = \tilde{c}_0 + \tilde{c}_1 u + \tilde{c}_2 u^2 + \tilde{c}_3 u^3 + \tilde{c}_4 u^4 + \tilde{c}_5 u^5 + \tilde{c}_6 u^6 + \cdots \\ &= \tilde{c}_0 + \tilde{c}_2 u^2 + \tilde{c}_4 u^4 + \tilde{c}_6 u^6 + \cdots . \end{aligned} \tag{8}$$

Taking $\sqrt{u}$ as the argument of the bipartite graph Ihara zeta function instead of u:

$$\begin{aligned} \zeta_H^{-1}(u) &= Z_{BG}^{-1}(\sqrt{u}) = \det(\boldsymbol{I}_{BG} - \sqrt{u}\boldsymbol{T}_{BG}) = \left(1 - (\sqrt{u})^{|p_1|}\right)\left(1 - (\sqrt{u})^{|p_2|}\right) \cdots \\ &= \tilde{c}_0 + 0\sqrt{u} + \tilde{c}_2(\sqrt{u})^2 + 0(\sqrt{u})^3 + \tilde{c}_4(\sqrt{u})^4 + 0(\sqrt{u})^5 + \tilde{c}_6(\sqrt{u})^6 + \cdots \\ &= \tilde{c}_0 + \tilde{c}_2 u + \tilde{c}_4 u^2 + \tilde{c}_6 u^3 + \cdots = c_0 + c_1 u + c_2 u^2 + c_3 u^3 + \cdots . \end{aligned} \tag{9}$$

As we can see in (9), the Ihara coefficients of a hypergraph can be efficiently obtained by selecting just the even-indexed Ihara coefficients of the associated bipartite graph. This is much more efficient than the computation based on the oriented line graph of the hypergraph, because $\boldsymbol{T}_{BG}$ is much smaller in size than $\boldsymbol{T}_H$, especially for large hypergraphs. The size of the Perron-Frobenius operator of an irregular hypergraph tends to be difficult to enumerate. Here we thus use the K-regular hypergraph, i.e. hypergraph with every hyperedge containing K vertices, for analyzing the computational complexity of the Perron-Frobenius operators $\boldsymbol{T}_H$ and $\boldsymbol{T}_{BG}$. Suppose there are in total N hyperedges in the K-regular hypergraph. To obtain $\boldsymbol{T}_H$, the clique expansion and its digraph of the K-regular hypergraph need to be established according to the transform introduced in Section 4.1. This procedure produces an oriented line graph with $K(K-1)N$ vertices and a Perron-Frobenius operator of size $(K-1)KN \times (K-1)KN$. To obtain $\boldsymbol{T}_{BG}$, the bipartite graph and its digraph of the K-regular hypergraph need to be established. This procedure produces an oriented line graph with $2KN$ vertices and a Perron-Frobenius operator of size $2KN \times 2KN$. For regular hypergraphs K is not less than 2, and the relation always holds for $2KN < (K-1)KN$. As a result, the size of $\boldsymbol{T}_{BG}$ is smaller than that of $\boldsymbol{T}_H$. The computational complexity of obtaining the Ihara coefficients is governed by the eigen-decomposition of the Perron-Frobenius operator. This requires $O(n^3)$ operations where n is the size of the Perron-Frobenius operator. Therefore, the computational overheads of eigen-decomposition on $\boldsymbol{T}_{BG}$ are lower than those of $\boldsymbol{T}_H$.

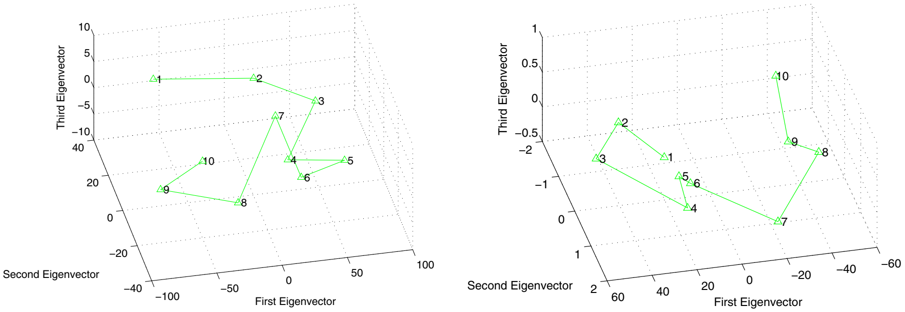
6 Experimental Evaluation

To establish hypergraphs on the visual objects, we first extract feature points using the Harris detector as the vertices of hypergraphs. Let $\boldsymbol{c}(v_i)$ denote the spatial coordinate of the feature point v_i in an image, and $I(v_i)$ denote the intensity of v_i. For each image, we construct the hypergraph using the method introduced in [5], where the element $H(i,j)$ of incidence matrix is 1 if $\|\boldsymbol{c}(v_i) - \boldsymbol{c}(v_j)\| \leq Th_{j1}$ and $| I(v_i) - I(v_j) | \leq Th_{j2}$, and 0 otherwise. Here Th_{j1} is the neighborhood threshold set to 1/4 the size of the image and Th_{j2} is the similarity threshold determined by the standard deviation of the intensities of neighboring feature points.

We first test the Ihara coefficient pattern vector in the form of $v_H = [c_3,\ c_4,\ \ln(|c_{M-3}|), \ln(|c_{M-2}|), \ln(|c_{M-1}|),\ \ln(|c_M|)]^T$ in characterizing within-class hypergraphs. We establish hypergraphs on ten images of a model house in the Chalet data set [5]. The images are taken consecutively as the camera pans around the model house in regular angular increments. Fig. 2 shows the PCA projections of the hypergraphs based on the truncated Laplacian spectrum, i.e. the leading six nonzero Laplacian eigenvalues, and the Ihara coefficients. The Laplacian spectra produce an erratic trajectory. The Ihara coefficients produce a much smoother trajectory and the neighboring images in the sequence are generally Euclidean neighbors in the eigenspace.

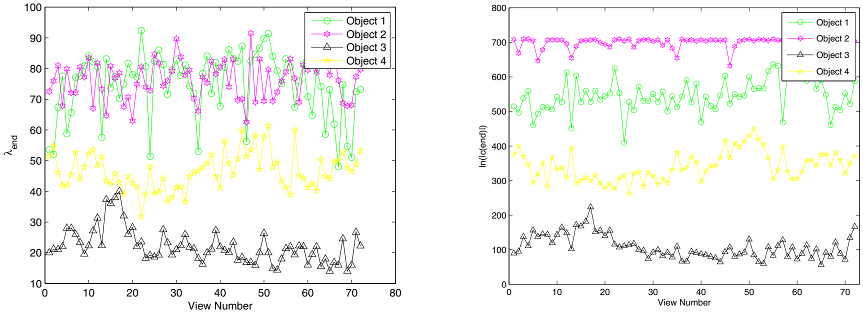
We then illustrate the largest Laplacian eigenvalue and the final Ihara coefficient for hypergraphs extracted from four objects in the COIL dataset [5]. The Ihara coefficients give clearer class separability than the Laplacian eigenvalues.

Finally we test the Ihara coefficients for clustering both unweighted graphs and unweighted hypergraphs. The graphs and hypergraphs are extracted from the images in the COIL dataset. We establish a Delaunay graph on the feature points of each image, and construct the pattern vectors in the form of $v_{Gs} = [c_3,\ c_4,\ \ln(|c_{2M}|)]^T$ for graphs. We evaluate the clustering performance obtained with different numbers of object classes. After performing PCA on the pattern vectors both for graphs and hypergraphs, we locate the clusters using the K-means method and calculate the Rand index, which is



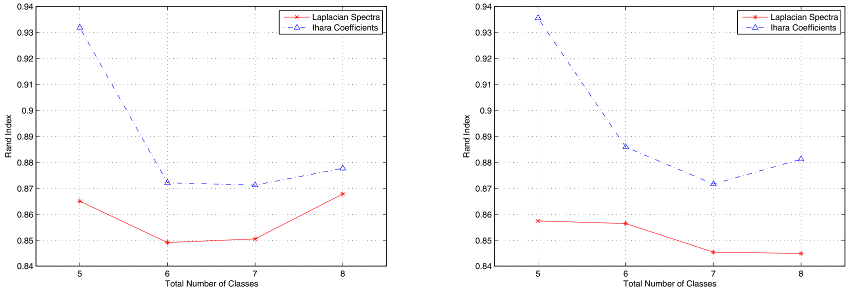
(a) Truncated Laplacian spectra. (b) Ihara coefficients.

Fig. 2. Within-class trajectory



(a) Leading nonzero Laplacian eigenvalue. (b) Ihara coefficients.

Fig. 3. Ihara coefficient plot



(a) Graph. (b) Hypergraph.

Fig. 4. Rand index

plotted as a function of class number in Fig. 4. We use Laplacian spectra for graphs and hypergraphs for comparison. From this set of experiments it is clarified that for both graphs and hypergraphs, the Ihara coefficients outperform the Laplacian spectra.

7 Conclusion

We have pointed out the deficiency of the vertex-to-vertex matrix representations for learning hypergraph-structure and applied the Ihara coefficients to hypergraph characterization to overcome these problems. The Ihara coefficients are a flexible tool which can be computed in a consistent manner for both graphs and hypergraphs. They can effectively overcome the ambiguity in distinguishing high order relational structures when matrix representations fail to work. Furthermore, we have proposed an efficient method for computing the Ihara coefficient set. Experimental results show that the Ihara coefficients are superior to spectral methods, both for graphs and hypergraphs.

Acknowledgments

We acknowledge the financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). Tatjana Aleksić is supported by Grant 144015G of the Serbian Ministry for Science and the British Scholarship Trust. Edwin R. Hancock is supported by a Royal Society Wolfson Research Merit Award.

References

1. Agarwal, S., Branson, K., Belongie, S.: Higher-order learning with graphs. In: ICML (2006)
2. Bass, H.: The ihara-selberg zeta function of a tree lattice. International Journal of Mathematics 6, 717–797 (1992)
3. Bretto, A., Cherifi, H., Aboutajdine, D.: Hypergraph imaging: an overview. Pattern Recognition 35(3), 651–658 (2002)
4. Ren, P., Aleksić, T., Wilson, R.C., Hancock, E.R.: Hypergraphs, characteristic polynomials and the ihara zeta function. In: CAIP (2009)
5. Ren, P., Wilson, R.C., Hancock, E.R.: Spectral embedding of feature hypergraphs. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 308–317. Springer, Heidelberg (2008)
6. Rota-Bullo, S., Albarelli, A., Pelillo, M., Torsello, A.: A hypergraph-based approach to affine parameters estimation. In: ICPR (2008)
7. Shashua, A., Zass, R., Hazan, T.: Multi-way clustering using super-symetric non-negtive tensor factorization. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 595–608. Springer, Heidelberg (2006)
8. Storm, C.K.: The zeta function of a hypergraph. Electronic Journal of Combinatorics 13 (2006)
9. Zass, R., Shashua, A.: Probabilistic graph and hypergraph matching. In: CVPR (2008)
10. Zhou, D., Huang, J., Scholkopf, B.: Learning with hypergraphs: Clustering, classification, and embedding. In: NIPS (2007)

A New Spectral Bound on the Clique Number of Graphs

Samuel Rota Bulò and Marcello Pelillo

Dipartimento di Informatica - University of Venice - Italy
{srotabul,pelillo}@dsi.unive.it

Abstract. Many computer vision and patter recognition problems are intimately related to the maximum clique problem. Due to the intractability of this problem, besides the development of heuristics, a research direction consists in trying to find good bounds on the clique number of graphs. This paper introduces a new spectral upper bound on the clique number of graphs, which is obtained by exploiting an invariance of a continuous characterization of the clique number of graphs introduced by Motzkin and Straus. Experimental results on random graphs show the superiority of our bounds over the standard literature.

1 Introduction

Many problems in computer vision and pattern recognition can be formulated in terms of finding a completely connected subgraph (i.e. a *clique*) of a given graph, having largest cardinality. This is called the maximum clique problem (MCP). One popular approach to object recognition, for example, involves matching an input scene against a stored model, each being abstracted in terms of a relational structure [1,2,3,4], and this problem, in turn, can be conveniently transformed into the equivalent problem of finding a maximum clique of the corresponding *association graph*. This idea was pioneered by Ambler et. al. [5] and was later developed by Bolles and Cain [6] as part of their local-feature-focus method. Now, it has become a standard technique in computer vision, and has been employing in such diverse applications as stereo correspondence [7], point pattern matching [8], image sequence analysis [9]. Other interesting applications of the maximum clique problem arise in the context of cluster analysis, where graph-theoretical methods have long proven to be especially effective [10,11,12], and in the context of category learning and knowledge discovery [13,14]. Furthermore, clique finding is also linked with the learning of graphical structure by the Hammersley-Clifford theorem [15].

From a computational point of view, the maximum clique problem (MCP) belongs to the class of NP-Complete problems, whose intractability forces us to fall back on approximation methods. Unfortunately, even approximating the MCP is intractable [16]. Due to this pessimistic state of affairs, much attention has gone into developing efficient heuristics for the MCP, for which no formal guarantee of performance may be provided, but are nevertheless useful in practical

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 680–689, 2010.

applications. We refer to Bomze et al. [17] for a survey concerning algorithms, applications, and complexity issues of this important problem.

Another interesting direction of research consists in trying to bound the clique number of a graph. In the literature we find both upper and lower bounds [18]. The former however are in general more interesting because any heuristics for the MCP can be used to generate lower bounds. In this paper we propose a new spectral upper bound by exploiting an invariance of a continuous characterization of the clique number of graphs introduced by Motzkin and Straus [19], and we present an algoritm for efficiently computing the bound. Experiments on random graphs demonstrate the effectiveness of our result. The bound proposed here can be used in the bounding phase of branch-and-bound style algorithms for finding maximal cliques, with applications in such problems as graph matching [1,20] and clustering [12] (see also [21] for the use of bounds in graph matching problems).

2 Bounds on the Clique Number of Graphs

Let $G = (V, E)$ be a (undirected) graph, where $V = \{1, \dots, n\}$ is the vertex set and $E \subseteq \binom{V}{2}$ is the edge set, with $\binom{V}{k}$ denoting the set of all k-element subsets of V. A *clique* of G is a subset of mutually adjacent vertices in V. A clique is called *maximal* if it is not contained in any other clique. A clique is called *maximum* if it has maximum cardinality. The maximum size of a clique in G is called the *clique number* of G and is denoted by $\omega(G)$.

Several spectral bounds on the clique number of graphs have been inspired by a theorem due to Motzkin and Straus [19]. This result establishes a link between the problem of finding the clique number of a graph G and the problem of optimizing the Lagrangian of G over the simplex Δ, where the Lagrangian of a graph $G = (V, E)$ is the function $L_G : \mathbb{R}^n \to \mathbb{R}$ defined as

$$L_G(\mathbf{x}) = \sum_{\{i,j\} \in E} x_i x_j \ ,$$

and the *standard simplex* Δ is the set of nonnegative n-dimensional real vectors that sum up to 1, i.e., $\Delta = \{\mathbf{x} \in \mathbb{R}^n_+ : \sum_{i=1}^n x_i = 1\}$.

Theorem 1 (Motzkin-Straus). *Let G be a graph with clique number $\omega(G)$, and $\mathbf{x}^*$ a maximizer of L_G over Δ then*

$$L_G(\mathbf{x}^*) = \frac{1}{2}\left[1 - \frac{1}{\omega(G)}\right] .$$

Assuming S a maximum clique of G, Motzkin and Straus additionally proved that the *characteristic vector* $\mathbf{x}^S$ of S defined as

$$x_i^S = \begin{cases} \frac{1}{|S|} & i \in S \\ 0 & i \notin S \end{cases}$$

is a global maximizer of L_G over Δ.

Before reviewing some existing bounds on ω, we briefly introduce some concepts from spectral graph theory. The *spectral radius* $\rho(G)$ of a graph G is the largest eigenvalue of the adjacency matrix of G. An eigenvector of unit length having $\rho(G)$ as eigenvalue will be called *Perron eigenvector* of G. The Perron eigenvector is always nonnegative and it may not be unique unless the multiplicity of the largest eigenvalue is exactly 1. By definition, the spectral radius ρ and an associated Perron eigenvector $\mathbf{x}_P$ of a graph G satisfy the eigenvalue equation

$$A_G\mathbf{x}_P = \rho\mathbf{x}_P \,,$$

which can be equivalently expressed in terms of the graph Lagrangian L_G as follows

$$\nabla L_G(\mathbf{x}_P) = \rho\mathbf{x}_P \,,$$

where ∇ is the standard gradient operator. Since G is undirected and hence, A_G is symmetric, a useful variational characterization of ρ and $\mathbf{x}_P$ is given by the following constrained program,

$$\rho = \max_{\mathbf{x}\in S_2} \mathbf{x}^T A_G \mathbf{x} = 2\max_{\mathbf{x}\in S_2} L_G(\mathbf{x}) \,, \tag{1}$$

where $S_k = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_k^k = 1\}$. Note that the eigenvectors of A_G are the critical points of this maximization problem. A further alternative characterization of the spectral radius and Perron eigenvector, that will be useful in the sequel, consists in maximizing the *Rayleigh quotient*, i.e.,

$$\rho = \max_{\mathbf{x}\in\mathbb{R}^n} \frac{\mathbf{x}^T A_G \mathbf{x}}{\mathbf{x}^T\mathbf{x}} = 2\max_{\mathbf{x}\in\mathbb{R}^n} \frac{L_G(\mathbf{x})}{\mathbf{x}^T\mathbf{x}} \,. \tag{2}$$

Note that every eigenvector associated to ρ is a maximizer in (2), whereas in (1) only a Perron eigenvector is a global maximizer.

We present now two upper bounds for ω that turned out to be the tightest ones in a paper of Budinich [18], where different bounds have been compared on random graphs. For a review of further spectral bounds we refer to [18,22].

The fist upper bound can be obtained by exploiting both the Motzkin-Straus theorem and (2).

Theorem 2. *Let G be an undirected graph with clique number $\omega(G)$ and spectral radius ρ. Then*

$$\omega(G) \leq \rho + 1 \,. \tag{B1}$$

Proof. Let $\mathbf{x}_\omega$ be the characteristic vector of a maximum clique of G, then $\mathbf{x}_\omega^T\mathbf{x}_\omega = 1/\omega(G)$ and by the Motzkin-Straus theorem $\mathbf{x}_\omega^T A_G \mathbf{x}_\omega = 1 - 1/\omega(G)$. By (2) we have that

$$\frac{\mathbf{x}_\omega^T A_G \mathbf{x}_\omega}{\mathbf{x}_\omega^T\mathbf{x}_\omega} = \frac{1-\frac{1}{\omega(G)}}{\frac{1}{\omega(G)}} = \omega(G) - 1 \leq \rho \,,$$

from which the property derives.

This bound can also be derived as a straightforward implication of the result of Wilf [23]. The second bound is due to Amin and Hakimi [24]:

Theorem 3. *Let G be an undirected graph with adjacency matrix A_G and clique number $\omega(G)$. Moreover, let N_{-1} be the number of eigenvalues of A_G that are less or equal to -1. Then*

$$\omega(G) \leq N_{-1} + 1 \,. \tag{B2}$$

3 The $\boldsymbol{\eta}$-Bound

We will introduce a new class of upper bounds generalizing (B1), where we exploit the fact that the maximizers of the Motzkin-Straus formulation are invariant with respect to shifts of the adjacency matrix of a graph G, whereas the maxima and the spectrum of the shifted matrix are not. Our intuition is that we can tighten (B1) by opportunely shifting the adjacency matrix of G.

We define

$$\phi_G(t, \mathbf{x}) = \mathbf{x}^T \left[A_G + (t-1)\mathbf{1}\mathbf{1}^\top\right] \mathbf{x} \,,$$

where $\mathbf{1}$ is an opportunely sized column vector of all 1's. Then by the Motzkin-Straus theorem we have

$$\max_{\mathbf{x}\in\Delta} \phi_G(t, \mathbf{x}) = \mathbf{x}^T \left[A_G + (t-1)\mathbf{1}\mathbf{1}^\top\right] \mathbf{x} = t - \frac{1}{\omega(G)} \,.$$

We will denote with $\phi_G(t)$ the leading eigenvalue of $A_G + (t-1)\mathbf{1}\mathbf{1}^\top$, i.e.,

$$\phi_G(t) = \max_{\mathbf{x}\in S_2} \phi_G(t, \mathbf{x}) \,,$$

and with $\Phi_G(t)$ the set of eigenvectors associated to $\phi_G(t)$, i.e.,

$$\Phi_G(t) = \arg\max_{\mathbf{x}\in S_2} \phi_G(t, \mathbf{x}) \,.$$

Theorem 4 (t-bound). *Let G be a graph with adjacency matrix A_G and clique number $\omega(G)$. Then for any $t > 0$*

$$\omega(G) \leq \frac{\phi_G(t) + 1}{t}$$

Proof. Let $\mathbf{x}_\omega$ be the characteristic vector of a maximum clique of G. Then

$$\phi_G(t) \geq \phi_G\left(t, \frac{\mathbf{x}_\omega}{\|\mathbf{x}_\omega\|_2}\right) = \frac{\phi_G(t, \mathbf{x}_\omega)}{\mathbf{x}_\omega^T \mathbf{x}_\omega} = \frac{t - \frac{1}{\omega(G)}}{\frac{1}{\omega(G)}} = \omega(G)t - 1 \,, \tag{3}$$

from which the result follows.

Theorem 4 introduces a class of upper bounds that contains (B1) as the special case $t = 1$. Let us define the *t-bound* as

$$\eta_G(t) = \frac{\phi_G(t) + 1}{t}.$$

Of course the more interesting t-bound is the tightest one, which will be called *η-bound* and denoted by $\eta(G)$, i.e.,

$$\eta(G) = \inf_{t>0} \eta_G(t).$$

Note that $\eta(G)$ is well defined, because by Theorem 4 it is lower bounded by the clique number of G.

4 Computation of the η-Bound

This section is dedicated to showing that the computation of $\eta(G)$ is not difficult, although not obvious at first glance, and we will provide an efficient algorithm for its computation.

Proposition 1. *Let $s > t > 0$. For any $\mathbf{x}(s) \in \Phi_G(s)$ and $\mathbf{x}(t) \in \Phi_G(t)$ we have*

$$(s-t)\mathbf{x}(t)^T\mathbf{1}\mathbf{1}^\top\mathbf{x}(t) \le \phi_G(s) - \phi_G(t) \le (s-t)\mathbf{x}(s)^T\mathbf{1}\mathbf{1}^\top\mathbf{x}(s)\,.$$

Proof

$$\begin{aligned}\phi_G(s) - \phi_G(t) &= \phi_G(s, \mathbf{x}(s)) - \phi_G(t)\\ &= \phi_G(t, \mathbf{x}(s)) + (s-t)\mathbf{x}(s)^T\mathbf{1}\mathbf{1}^\top\mathbf{x}(s) - \phi_G(t)\\ &\le \phi_G(t, \mathbf{x}(t)) + (s-t)\mathbf{x}(s)^T\mathbf{1}\mathbf{1}^\top\mathbf{x}(s) - \phi_G(t)\\ &= \phi_G(t) + (s-t)\mathbf{x}(s)^T\mathbf{1}\mathbf{1}^\top\mathbf{x}(s) - \phi_G(t)\\ &= (s-t)\mathbf{x}(s)^T\mathbf{1}\mathbf{1}^\top\mathbf{x}(s)\end{aligned}$$

$$\begin{aligned}\phi_G(s) - \phi_G(t) &= \phi_G(s, \mathbf{x}(s)) - \phi_G(t)\\ &\ge \phi_G(s, \mathbf{x}(t)) - \phi_G(t)\\ &= \phi_G(t, \mathbf{x}(t)) + (s-t)\mathbf{x}(t)^T\mathbf{1}\mathbf{1}^\top\mathbf{x}(t) - \phi_G(t)\\ &= \phi_G(t) + (s-t)\mathbf{x}(t)^T\mathbf{1}\mathbf{1}^\top\mathbf{x}(t) - \phi_G(t)\\ &= (s-t)\mathbf{x}(t)^T\mathbf{1}\mathbf{1}^\top\mathbf{x}(t)\end{aligned}$$

Proposition 2. *Let $s > t > 0$. For any $\mathbf{x}(s) \in \Phi_G(s)$ and $\mathbf{x}(t) \in \Phi_G(t)$ the following propositions hold*

1. *if $\phi_G(0, \mathbf{x}(s)) \ge -1$ then $\eta_G(s) \le \eta_G(t)$,*
2. *if $\phi_G(0, \mathbf{x}(t)) \le -1$ then $\eta_G(s) \ge \eta_G(t)$.*

Proof. If $\phi_G(0,\mathbf{x}(s)) \geq -1$ then

$$\begin{aligned}\eta_G(s)-\eta_G(t) &= \frac{\phi_G(s)+1}{s}-\frac{\phi_G(t)+1}{t}\\ &\leq \frac{\phi_G(t)+(s-t)\mathbf{x}(s)^T\mathbf{1}\mathbf{1}^\top\mathbf{x}(s)+1}{s}-\frac{\phi_G(t)+1}{t} \quad \text{(by Prop. 1)}\\ &= \frac{(s-t)\left[-\phi_G(t)-1+t\mathbf{x}(s)^T\mathbf{1}\mathbf{1}^\top\mathbf{x}(s)\right]}{ts}\\ &\leq \frac{(s-t)\left\{-\phi_G(t,\mathbf{x}(s))-1+t\mathbf{x}(s)^T\mathbf{1}\mathbf{1}^\top\mathbf{x}(s)\right\}}{ts}\\ &= \frac{(s-t)\left[-\phi_G(0,\mathbf{x}(s))-1\right]}{ts} \leq 0\,.\end{aligned}$$

While if $\phi_G(0,\mathbf{x}(t)) \leq -1$ then

$$\begin{aligned}\eta_G(s)-\eta_G(t) &= \frac{\phi_G(s)+1}{s}-\frac{\phi_G(t)+1}{t}\\ &\geq \frac{\phi_G(t)+(s-t)\mathbf{x}(t)^T\mathbf{1}\mathbf{1}^\top\mathbf{x}(t)+1}{s}-\frac{\phi_G(t)+1}{t} \quad \text{(by Prop. 1)}\\ &= \frac{(s-t)\left[-\phi_G(t)-1+t\mathbf{x}(t)^T\mathbf{1}\mathbf{1}^\top\mathbf{x}(t)\right]}{ts}\\ &= \frac{(s-t)\left[-\phi_G(0,\mathbf{x}(t))-1\right]}{ts} \geq 0\,.\end{aligned}$$

Theorem 5. *Let $s > t > 0$. For any $\mathbf{x}(s) \in \Phi_G(s)$ and $\mathbf{x}(t) \in \Phi_G(t)$ if*

$$\phi_G(0,\mathbf{x}(s)) \leq -1 \leq \phi_G(0,\mathbf{x}(t))\,,$$

then there exists $t \leq q \leq s$ such that $\eta(G) = \eta_G(q)$.

Moreover, if for any $q > 0$ and $\mathbf{x}(q) \in \Phi_G(q)$ we have $\phi_G(0,\mathbf{x}(q)) = -1$ then $\eta(G) = \eta_G(q)$.

Proof. By Proposition 2 it follows that

- for any $r < t$ we have $\eta(G) \leq \eta_G(t) \leq \eta_G(r)$;
- for any $r > s$ we have $\eta(G) \leq \eta_G(s) \leq \eta_G(r)$,

from which the first part of the result follows.

For the second part note that for any $r < q$ we have $\eta(G) \leq \eta_G(q) \leq \eta_G(r)$, while for any $r > q$ we have $\eta(G) \leq \eta_G(q) \leq \eta_G(r)$. Hence, $\eta_G(q) = \eta(G)$.

Proposition 3. *Let $\mathbf{x}(1) \in \Phi_G(1)$ be a Perron eigenvector of A_G. For any $\mathbf{x}(0) \in \Phi_G(0)$ we have*

$$\phi_G(0,\mathbf{x}(1)) \leq -1 \leq \phi_G(0,\mathbf{x}(0))\,.$$

Algorithm 1. Bisection search for computing $\eta(G)$

```
function $\eta$-BOUND($G$,$\epsilon$)
    $l = 0$
    $p = 0$
    $r = 1$
    Take any $\mathbf{x}(l) \in \Phi_G(l)$
    $\mathbf{x}(r) \leftarrow$ normalized Perron vector of $A_G$
    while $r - l > \epsilon$ do
        $p = (l + r)/2$                    ▷ or any other selection mechanism
        Take any $\mathbf{x}(p) \in \Phi_G(p)$
        if $\phi_G(0, \mathbf{x}(p)) < -1$ then
            $r \leftarrow p$
            $\mathbf{x}(r) \leftarrow \mathbf{x}(p)$
        else if $\phi_G(0, \mathbf{x}(p)) > -1$ then
            $l \leftarrow p$
            $\mathbf{x}(l) \leftarrow \mathbf{x}(p)$
        else
            return $\eta_G(p)$
        end if
    end while
    return $\eta_G(p)$
end function
```

Proof. It follows from (3) that $\phi_G(0) = \phi_G(0, \mathbf{x}(0)) \geq -1$.

Because of the nonnegativity of the Perron vector, trivially $\mathbf{x}(1)^T A_G \mathbf{x}(1) \leq \mathbf{x}(1)^T(\mathbf{1}\mathbf{1}^\top - I)\mathbf{x}(1)$, from which it follows that

$$\begin{aligned}\phi_G(0, \mathbf{x}(1)) + 1 = \\ = \mathbf{x}(1)^T \left[A_G - \mathbf{1}\mathbf{1}^\top\right] \mathbf{x}(1) + 1 \leq \mathbf{x}(1)^T \left[(\mathbf{1}\mathbf{1}^\top - I) - \mathbf{1}\mathbf{1}^\top\right] \mathbf{x}(1) + 1 = 0.\end{aligned}$$

Theorem 5 and Proposition 3 suggest an effective way of computing $\eta(G)$ by performing a section search (like the bisection search) in the interval (0,1]. Indeed, Theorem 5 allows us to bisect an interval having sign-discording values of $f(t) = \phi_G(0, \mathbf{x}(t)) + 1$ at the endpoints, and restrict the attention to the subinterval that preserves this property. Proposition 3, instead, entitles us to start the search procedure from the interval $[0, 1]$. Note that we can stop the search if we encounter an endpoint t, where $f(t) = 0$, as in this case $\eta_G(t)$ is our η-bound. Otherwise, the size of the interval is an indicator of the precision of the solution and we can stop as soon as this is small enough. Algorithm 1 reports an implementation that can be used for the computation of $\eta(G)$ with an arbitrary precision ϵ.

5 Experiments on Random Graphs

In this section, we evaluate the performance of our η-bound. We compare our bound against other spectral bounds, which were the best performing approaches

Table 1. Experiments on random graphs. The columns n, δ and ω are the order, density and average clique number of the random graphs, respectively. The results, expecting the last row, are expressed in terms of relative error.

Random graphs			Bound errors		
n	δ	ω	(B1)	(B2)	η
100	0.05	3.12	1.25	10.58	**0.79**
	0.10	3.96	1.99	9.26	**0.87**
	0.20	5.00	3.33	7.84	**1.07**
	0.30	6.13	4.17	6.52	**1.11**
	0.40	7.51	4.49	5.24	**1.08**
	0.50	9.11	4.58	4.19	**1.02**
	0.60	11.51	4.28	3.16	**0.91**
	0.70	14.55	3.85	2.33	**0.84**
	0.80	19.99	3.03	1.45	**0.64**
	0.90	30.69	1.94	0.61	**0.42**
	0.95	43.50	1.19	**0.16**	0.27
200	0.10	4.17	4.25	19.97	**1.45**
	0.50	11.00	8.19	7.71	**1.47**
	0.90	?	180.10	99.08	**68.45**

reviewed in the work of Budinich [18]. Specifically, we compare against bounds (B1) and (B2), which have been previously introduced.

Table 1 reports the results obtained on random graphs, where η is the column relative to our η-bound. The columns n, δ and ω are the order, density and average clique number of the random graphs, respectively. The results, except the last row, are expressed in terms of relative error, i.e. if $\overline{\omega}$ is the value of the bound then the relative error for the upper and lower bounds are $(\overline{\omega} - \omega)/\omega$ and $(\omega - \overline{\omega})/\omega$, respectively. In the last row, where the average clique number could not be computed, we reported the absolute value of the bounds. It is clear that, as expected, our η-bound improves (B1). Moreover, our bound outperforms also Amin's one on all instances excepting very dense graphs. Interestingly, it exhibits on average a remarkable improvement over the competitors by keeping an overall small relative error.

6 Conclusions

In this paper, we introduced a new spectral bounds on the clique number of graphs, called η-bound, which has been obtained by combining spectral graph theory with a result due to Motzkin and Straus. Specifically, we exploit an invariance of the Motzkin-Sraus formulation with respect to shifts of the adjacency matrix of graphs in order to tighten a well-known bound.

Finally, we tested our bounds on random graphs comparing them against state-of-the-art spectral approaches. The results outlined a marked improvement over the competitors.

References

1. Barrow, H., Burstall, R.M.: Subgraph isomorphism, matching relational structures and maximal cliques. Information Processing Letters 4(4), 83–84 (1976)
2. Chin, R.T., Dyer, C.R.: Model-based recognition in robot vision. Comput. Surveys 18(1), 67–108 (1986)
3. Pelillo, M., Siddiqi, K., Zucker, S.W.: Matching hierarchical structures using association graphs. IEEE Trans. Pattern Anal. Machine Intell. 21(11), 1105–1120 (1999)
4. Suetens, P., Fua, P., Hanson, A.J.: Computational strategies for object recognition. Comput. Surveys 24(1), 5–61 (1992)
5. Ambler, A.P., Barrow, H.G., Brown, C.M., Burstall, R.M., Popplestone, R.J.: A versatile computer-controlled assembly. In: Int. Joint Conf. on Artif. Intell., pp. 298–307 (1973)
6. Bolles, R.C., Cain, R.A.: Recognizing and locating partially visible objects: the local-feature-focus method. Int. J. Robotics Res. 1(n), 57–82 (1982)
7. Horaud, R., Skordas, T.: Stereo correspondence through feature grouping and maximal cliques. IEEE Trans. Pattern Anal. Machine Intell. 11(11), 1168–1180 (1989)
8. Ogawa, H.: Labeled point pattern matching by delaunay triangulation and maximal cliques. Pattern Recogn. 19(1), 35–40 (1986)
9. Radig, B.: Image sequence analysis using relational structures. Pattern Recogn. 17(1), 161–167 (1984)
10. Augustson, J.G., Minker, J.: An analysis of some graph theoretical cluster techniques. J. ACM 17(4), 571–588 (1970)
11. Jain, A.K., Dubes, R.C.: Algorithms for data clustering. Prentice-Hall, Englewood Cliffs (1988)
12. Pavan, M., Pelillo, M.: Dominant sets and pairwise clustering. IEEE Trans. Pattern Anal. Machine Intell. 29(1), 167–172 (2007)
13. Dmitry, D., Ari, R.: Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In: 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of the ACL, Association for Computational Linguistics, pp. 297–304 (2006)
14. Nina, M., Dana, R., Ram, S.: A new conceptual clustering framework. Machine Learning 56, 115–151 (2004)
15. Hammersley, J., Clifford, P.: Markov fields on finite graphs and lattices (1971)
16. Hastad, J.: Clique is hard to approximate within $n^{1-\varepsilon}$. In: Ann. Symp. Found. Comput. Sci., vol. 37, pp. 627–636 (1996)
17. Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M.: The maximum clique problem. In: Handbook of Combinatorial Optimization, vol. 1, pp. 1–74. Kluwer Academic Publishers, Boston (1999)
18. Budinich, M.: Exact bounds on the order of the maximum clique of a graph. Discr. Appl. Math. 127, 535–543 (2003)

19. Motzkin, T.S., Straus, E.G.: Maxima for graphs and a new proof of a theorem of Turán. Canad. J. Math. 17, 533–540 (1965)
20. Pelillo, M.: Replicator equations, maximal cliques, and graph isomorphism. Neural Computation 11(8), 1933–1955 (1999)
21. Schellewald, C.: A bound for non-subgraph isomorphism. In: Escolano, F., Vento, M. (eds.) GbRPR. LNCS, vol. 4538, pp. 71–80. Springer, Heidelberg (2007)
22. Lu, M., Liu, H., Tian, F.: Laplacian spectral bounds for clique and independence numbers of graphs. J. Combin. Theory Series B 97(5), 726–732 (2007)
23. Wilf, H.S.: The eigenvalues of a graph and its chromatic number. J. London Math. Soc. 42, 330–332 (1967)
24. Amin, A.T., Hakimi, S.L.: Upper bounds of the order of a clique of a graph. SIAM J. on Appl. Math. 22(4), 569–573 (1972)

Large Sample Statistics in the Domain of Graphs

Brijnesh J. Jain and Klaus Obermayer

Berlin Institute of Technology, Germany
{jbj,oby}@cs.tu-berlin.de

Abstract. One challenge in bridging the gap between structural and statistical pattern recognition consists in studying combinatorial structures like graphs using probabilistic methods. This contribution presents the structural counterparts of the first and second fundamental theorem in probability, (1) the law of large numbers and (2) the central limit theorem. In addition, we derive characterizations and uniqueness conditions for the mean of graphs. As a special case, we investigate the weighted mean of two graphs. The proposed results establish a sound statistical foundation for unsupervised structural pattern recognition methods.

1 Introduction

Central points such as the median and mean of a finite set of graphs find their applications in central clustering of graphs [5,9,10,11,19], graph quantization [13], frequent substructure mining [18] and multiple alignment of protein structures [14]. Because of their elementary importance, a thorough understanding of central points for a distribution of graphs is necessary in order to statistically justify and algorithmically improve existing unsupervised structural pattern recognition methods. For this reason, first theoretical results on central points in the domain of graphs have been established [4,6,12,15,17]. Compared to vector spaces, however, a fundamental understanding of the graph mean is still missing.

This paper aims at providing new insight to basic properties in large sample statistics of attributed graphs. We restate the strong law of large numbers for distributions on graphs presented in [15]. As novel results, we (1) propose a central limit theorem for distributions on graphs, (2) characterize the mean of graphs, (3) propose sufficient conditions for uniqueness of the mean of graphs, and (4) present properties of the weighted mean of two graphs. In order to derive these results an appropriate approach to represent graphs is necessary. The approach we suggest is to represent graphs as points in some Riemannian orbifold. An orbifold is a quotient of a manifold by a finite group action and therefore generalizes the notion of manifold. Using orbifolds we can derive an intrinsic metric that enables us to adopt integration locally to a Euclidean space.

The proposed approach has the following properties: First, it can be applied to finite combinatorial structures other than graphs like, for example, point patterns, sequences, trees, and hypergraphs can all be embedded isometrically into a Riemannian orbifold. For the sake of concreteness, we restrict our attention

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 690–697, 2010.

exclusively to the domain of graphs. Second, for graphs consisting of a single vertex with feature vectors as attributes, the proposed learning graph quantization (LGQ) coincides with LVQ.

This paper is organized as follows: Section 2 represents attributed graphs as points in an orbifold. Section 3 derives properties of the graph mean and Section 4 concludes.

2 Representation of Attributed Graphs

In order to do statistical data analysis, we need an appropriate representation of attributed graphs. We suggest to represent graphs as points in some Riemannian orbifold, since orbifolds allow us to apply useful concepts and techniques from differential geometry.

Let $\mathbb{E}$ be a d-dimensional Euclidean space. An *attributed graph* is a triple $X = (V, E, \alpha)$ consisting of a set V of *vertices*, a set $E \subseteq V \times V$ of *edges*, and an *attribute function* $\alpha : V \times V \rightarrow \mathbb{E}$, such that $\alpha(i,j) \neq \mathbf{0}$ for each edge and $\alpha(i,j) = \mathbf{0}$ for each non-edge. Attributes $\alpha(i,i)$ of vertices i may take any value from $\mathbb{E}$.

For simplifying the mathematical treatment, we assume that all graphs are of order n, where n is chosen to be sufficiently large. Graphs of order less than n, say $m < n$, can be extended to order n by including isolated vertices with attribute zero. For practical issues, it is important to note that limiting the maximum order to some arbitrarily large number n and extending smaller graphs to graphs of order n are purely technical assumptions to simplify mathematics. For pattern recognition problems, these limitations should have no practical impact, because neither the bound n needs to be specified explicitly nor an extension of all graphs to an identical order needs to be performed. When applying the theory, all we actually require is that the graphs are finite.

A graph X is completely specified by its *matrix representation* $\boldsymbol{X} = (\boldsymbol{x}_{ij})$ with elements $\boldsymbol{x}_{ij} = \alpha(i,j)$ for all $1 \leq i,j \leq n$. Let $\mathcal{X} = \mathbb{E}^{n \times n}$ be the Euclidean space of all $(n \times n)$-matrices with elements from $\mathbb{E}$ and let $\mathit{\Gamma}$ denote a subgroup of all $(n \times n)$-permutation matrices. Two matrices $\boldsymbol{X}, \boldsymbol{X}' \in \mathcal{X}$ are said to be equivalent, if there is a permutation matrix $P \in \mathit{\Gamma}$ such that $\boldsymbol{P}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{P} = \boldsymbol{X'}$. By

$$\mathcal{X}/\mathit{\Gamma} = \{[\boldsymbol{X}] \,:\, \boldsymbol{X} \in \mathcal{X}\}$$

we denote the quotient set consisting of all equivalence classes $[\boldsymbol{X}]$.

For notational convenience, we identify $\mathcal{X}$ with $\mathbb{E}^N$, where $N = n^2$ and consider vector- rather than matrix representations of graphs. By concatenating the columns of a matrix representation $\boldsymbol{X}$ of a graph X, we obtain a *vector representation* $\boldsymbol{x}$ of X.

Now we are in the position to take the final step towards representing graphs as points in a Riemannian orbifold. A *Riemannian orbifold of graphs* is a triple $\mathcal{Q} = (\mathcal{X}, \mathit{\Gamma}, \pi)$ consisting of an Euclidean space $\mathcal{X}$ with norm $\|\cdot\|$, a permutation group $\mathit{\Gamma}$ acting on $\mathcal{X}$, and an *orbifold chart*

$$\pi : \mathcal{X} \rightarrow \mathcal{X}_{\mathcal{Q}} = \mathcal{X}/\mathit{\Gamma}, \quad \boldsymbol{x} \mapsto [\boldsymbol{x}]$$

that projects each vector $\boldsymbol{x}$ to its orbit $[\boldsymbol{x}]$. We use capital letters X, Y, Z to denote graphs from $\mathcal{X}_{\mathcal{Q}}$ and write $\boldsymbol{x} \in X$ if $\pi(\boldsymbol{x}) = X$. Each vector $\boldsymbol{x} \in X$ is a *vector representation* of structure X and the set $\mathcal{X}$ of all vector representations is the *representation space* of $\mathcal{X}_{\mathcal{Q}}$.

The *intrinsic metric* of an orbifold $\mathcal{Q} = (\mathcal{X}, \Gamma, \pi)$ of graphs is of the form

$$d(X, X') = \min\left\{\|\boldsymbol{x} - \boldsymbol{x}'\| \,:\, \boldsymbol{x} \in X, \boldsymbol{x}' \in X'\right\}.$$

We call a pair $(\boldsymbol{x}, \boldsymbol{x}') \in X \times X'$ with $\|\boldsymbol{x} - \boldsymbol{x}'\| = d(X, X')$ an *optimal alignment* of X and X'. By $\mathcal{A}(X, X')$ we denote the set of all optimal alignments of X and X'. Note that the intrinsic metric is not a artificial construction for analytical purposes but rather appears in different guises as a common choice of proximity measure for graphs [2,3,8,20].

3 The Frechet Mean

In this section, we focus on the mean of a distribution on graphs. Unless otherwise stated, proofs of all results are delegated to [16].

3.1 The Frechet Mean Set of Graphs

Since it is unclear how to define the mean of graphs using a weighted sum or an integral of graphs, we present a definition based on the properties of the usual mean as suggested by Frechet [7]. The basic idea of Frechet to define central points in a metric space is essentially the same as for the concept of graph mean and median proposed by [4,6,12,15,17].

Suppose that $(\mathcal{Q}, d)$ is a metric orbifold of graphs with $\mathcal{Q} = (\mathcal{X}, \Gamma, \pi)$. We define the *Frechet function* as

$$F(Y) = \mathbb{E}_X\left[d(X, Y)^2\right] = \begin{cases} \displaystyle\int_{\mathcal{C}_{\mathcal{Q}}} d(X, Y)^2\, dP_{\mathcal{Q}}(X) & : \quad X \text{ is continuos} \\ \displaystyle\sum_{X \in \mathcal{C}_{\mathcal{Q}}} d(X, Y)^2\, P_{\mathcal{Q}}(X) & : \quad X \text{ is discrete} \end{cases},$$

where $P_{\mathcal{Q}}$ is a probability measure on the Borel sigma-field of $\mathcal{X}_{\mathcal{Q}}$ with support on a measurable subset $\mathcal{C}_{\mathcal{Q}}$ in the continuous case and a probability mass function in the discrete case. A *Frechet mean* is any element $M \in \mathcal{C}_{\mathcal{Q}}$ satisfying

$$F(M) = \inf_{Y \in \mathcal{C}_{\mathcal{Q}}} F(Y) < \infty.$$

The *Frechet mean set* $\mathcal{F}$ is the set of all Frechet means.

3.2 Characterization of Frechet Means

For characterizing a Frechet mean, we need the notion of Dirichlet fundamental domain. A *fundamental domain* of Γ in $\mathcal{X}$ is a closed subset $\mathcal{D} \subset \mathcal{X}$ with

$$\mathcal{X} = \bigcup_{\gamma \in \Gamma} \gamma(\mathcal{D})$$

and $\operatorname{int}(\gamma(\mathcal{D})) \cap \operatorname{int}(\gamma'(\mathcal{D})) = \emptyset$ for all $\gamma, \gamma' \in \Gamma$. Thus, the interior of a fundamental domain projects to the entire domain of graphs, where interior points of the fundamental domain are unique vector representations. A *Dirichlet fundamental domain* of $\boldsymbol{x} \in \mathcal{X}$ is a fundamental domain satisfying

$$\boldsymbol{x}' \in \mathcal{D}(\boldsymbol{x}) \;\Rightarrow\; \|\boldsymbol{x} - \boldsymbol{x}'\| \leq \|\boldsymbol{x} - \gamma(\boldsymbol{x}')\| \qquad \forall\, \gamma \in \Gamma.$$

Each Dirichlet fundamental domain $\mathcal{D}(\boldsymbol{x})$ is a convex polyhedral cone containing at least one vector representation of each graph. Two vector representations in $\mathcal{D}(\boldsymbol{x})$ projecting to the same graph always lie on the boundary of $\mathcal{D}(\boldsymbol{x})$. Since the boundary is of Lebesgue measure zero, we can regard the domain of graphs as being geometrically a polyhedral cone.

Theorem 1 shows that any vector representation $\boldsymbol{m}$ of a Frechet mean M is a population mean of the lifted probability distribution on its Dirichlet fundamental domain $\mathcal{D}(\boldsymbol{m})$.

Theorem 1 (Representation of a Frechet Mean). *Let $\mathcal{Q} = (\mathcal{X}, \Gamma, \pi)$ be an orbifold of graphs with intrinsic metric d and let $(\mathcal{X}_\mathcal{Q}, \Sigma_\mathcal{Q}, P_\mathcal{Q})$ be a probability space. Suppose that $M \in \mathcal{F}$ is a Frechet mean of $P_\mathcal{Q}$. Then any vector representation $\boldsymbol{m} \in \mathcal{X}$ that projects to M is of the form*

$$\boldsymbol{m} = \int_{\mathcal{D}(\boldsymbol{m})} \boldsymbol{x}\, dP(\boldsymbol{x}),$$

where $P(\boldsymbol{x}) = P_\mathcal{Q}(\pi(\boldsymbol{x}))$ on $\mathcal{D}(\boldsymbol{m})$.

From the definition of the Dirichlet fundamental domain follows that a vector representation $\boldsymbol{m}$ of a Frechet mean M is the population mean of the distribution on all vector representations $\boldsymbol{x}$ of X optimally aligned with $\boldsymbol{m}$.

3.3 Uniqueness of Frechet Mean

Next, we show under which assumptions the Frechet mean of graphs consists of a singleton. For this, we define the *injectivity radius* of a structure $X \in \mathcal{X}_\mathcal{Q}$ by

$$r_X = \min\left\{\|\boldsymbol{x} - \boldsymbol{x}'\| \,:\, \boldsymbol{x}' \in \operatorname{bd}(\mathcal{D}(\boldsymbol{x}))\right\},$$

where $\boldsymbol{x} \in X$ is a vector representation. The injectivity radius r_X measures the shortest distance from $\boldsymbol{x}$ to the boundary of its Dirichlet fundamental region $\mathcal{D}(\boldsymbol{x})$. The injectivity radius r_X is independent of the choice of vector representation ([16], Prop. 5). Thus, r_X is well-defined. The *injectivity angle* of X is

$$\alpha_X = \arcsin \frac{r_X}{l(X)}.$$

The injectivity angle is the smallest angle between a fixed vector representation $\boldsymbol{x}$ of X and a vector lying on a boundary of $\mathcal{D}(\boldsymbol{x})$. By definition, the injectivity angle is independent of the choice o vector representation of X.

The Frechet mean consists of a singleton, if the graphs are distributed within a circular right cone with sufficient narrow opening angle.

Theorem 2 (Uniqueness of Frechet Mean). *Let $\mathcal{Q} = (\mathcal{X}, \Gamma, \pi)$ be an orbifold of graphs with intrinsic metric d and let $(\mathcal{X}_\mathcal{Q}, \Sigma_\mathcal{Q}, P_\mathcal{Q})$ be a probability space. Suppose there is a structure $Z \in \mathcal{X}_\mathcal{Q}$ such that the support of $P_\mathcal{Q}$ is a measurable subset of the open circular cone*

$$\mathcal{C}_\Gamma\left(Z, \frac{\alpha_Z}{3}\right) = \left\{X \in \mathcal{X}_\mathcal{Q} \,:\, \measuredangle(Z, X) < \frac{\alpha_Z}{3}\right\}$$

with cone axis in direction of Z. Then the Frechet mean of $P_\mathcal{Q}$ is unique.

Suppose that $\boldsymbol{z}$ projects to Z and $\mathcal{C} \subseteq \mathcal{D}(\boldsymbol{z})$ projects to $\mathcal{C}_\Gamma\,(Z, \alpha_Z/3)$. Then from the proof of Theorem 2 follows that the elements of $\mathcal{C}$ are pairwise optimally aligned. Hence, we may identify open sets of $\mathcal{C}_\Gamma$ with open set of $\mathcal{C}$ and apply any mathematical result that holds locally in a Euclidean space. In particular, we directly obtain as a Corollary the Law of Large Numbers and the Central Limit Theorem.

3.4 A Strong Law of Large Numbers

Since the distribution $P_\mathcal{Q}$ is usually unknown and the underlying metric space often lacks sufficient mathematical structure, the Frechet function $F(Y)$ can neither be computed nor be minimized directly. Instead, we estimate a Frechet mean from empirical data. Suppose that $X_1, X_2, \ldots, X_N \in \mathcal{X}_\mathcal{T}$ is an independent and identically distributed random sample. We replace the Frechet function by the *empirical Frechet function*

$$\hat{F}_N(Y) = \frac{1}{N}\sum_{i=1}^{N} d(X_i, Y)^2$$

and approximate a Frechet mean by a global minimum of the empirical Frechet function. By $\hat{\mathcal{F}}_N$ we denote the set of *Frechet sample means* consisting of all global minima of $\hat{F}_N(Y)$.

As shown in [15], the strong law of large numbers for a distribution on graphs can be directly derived from [1].

Theorem 3. *Let $\mathcal{Q} = (\mathcal{X}, \Gamma, \pi)$ an orbifold of graphs with intrinsic metric d and let $(\mathcal{X}_\mathcal{Q}, \Sigma_\mathcal{Q}, P_\mathcal{Q})$ be a probability space. Suppose that the Frechet function F of $P_\mathcal{Q}$ is finite. Then for any $\varepsilon > 0$, there is a random variable $n(\omega, \varepsilon) \in \mathbb{N}$ and a $P_\mathcal{Q}$-null set $\mathcal{N}(\omega, \varepsilon)$ such that*

$$\hat{\mathcal{F}}_N \subseteq \mathcal{F}_\varepsilon = \left\{X \in \mathcal{X}_\mathcal{T} \,:\, \min_{M \in \mathcal{F}} d\,(X, M)^2 < \varepsilon\right\}$$

outside of $\mathcal{N}(\omega, \varepsilon)$ for all $N \geq n(\omega, \varepsilon)$. In particular, if the set $\mathcal{F} = \{\mu\}$ of Frechet means consists of a singleton μ, then every measurable selection, $\hat{\mu}_N$ from $\hat{\mathcal{F}}_N$ is a strongly consistent estimator of μ.

3.5 A Central Limit Theorem

Next, we want to derive a version of the central limit theorem for graphs. For this we introduce the following notations. The expression $X_N = o(Y_N)$ means that $X_N/Y_N \overset{P_{\mathcal{Q}}}{\to} 0$. In particular $X_N = o(Y_N)$ means that $X_N \overset{P_{\mathcal{Q}}}{\to} 0$.

Theorem 4. *Let $\mathcal{Q} = (\mathcal{X}, \Gamma, \pi)$ an orbifold of graphs with intrinsic metric d and let $(\mathcal{X}_{\mathcal{Q}}, \Sigma_{\mathcal{Q}}, P_{\mathcal{Q}})$ be a probability space. If*

$$\hat{F}_N(\hat{M}_N) \leq \inf_{Y \in \mathcal{X}_{\mathcal{Q}}} F(Y) + o_{P_{\mathcal{Q}}}\left(\frac{1}{N}\right)$$

and

$$\hat{M}_N \overset{P_{\mathcal{Q}}}{\to} M \in \mathcal{F},$$

where M is nonsingular. Then

$$\sqrt{N}\left(\hat{M}_N - M\right) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{m}) - o_{P_{\mathcal{Q}}}(1),$$

where $\boldsymbol{m}$ projects to M and $(\boldsymbol{x}_i, \boldsymbol{m}) \in \mathcal{A}(X_i, M)$ are optimal alignments. In particular, the sequence is $\sqrt{N}\left(\hat{M}_N - M\right)$ is asymptotically normal with mean zero and covariance matrix

$$\Sigma = \int_{\mathcal{D}(\boldsymbol{m})} (\boldsymbol{x} - \boldsymbol{m})(\boldsymbol{x} - \boldsymbol{m})^{\mathsf{T}} dP(\boldsymbol{x}).$$

3.6 The Frechet Mean of Two Structures

As a special case, we consider the Frechet function of the form

$$F(Y) = p \cdot d(X, Y)^2 + (1 - p) \cdot d(X', Y)^2,$$

where $p = P_{\mathcal{Q}}(X)$ and $1 - p = P_{\mathcal{Q}}(X')$ are the probabilities of the structures X and X', respectively. Regarding p and $1-p$ as weights rather than probabilities, the Frechet mean set of $F(Y)$ is an elementary component of competitive learning methods for central clustering and graph quantization [13].

Theorem 5. *Let $\mathcal{Q} = (\mathcal{X}, \Gamma, \pi)$ be an orbifold of graphs with intrinsic metric $d(\cdot|\cdot)$. Consider the Frechet function*

$$F(Y) = p \cdot d(X, Y)^2 + (1 - p) \cdot d(X', Y)^2,$$

where $X, X' \in \mathcal{X}_{\mathcal{Q}}$ and $p \in [0, 1]$. Then the following holds:

(P_1) Any vector representation $\boldsymbol{m}$ of a Frechet mean $M \in \mathcal{F}$ is of the form

$$\boldsymbol{m} = p \cdot \boldsymbol{x} + (1 - p) \cdot \boldsymbol{x}',$$

where $(\boldsymbol{x}, \boldsymbol{m}) \in \mathcal{A}(X, M)$ and $(\boldsymbol{x}', \boldsymbol{m}) \in \mathcal{A}(X', M)$ are optimal alignments.

(P_2) $M \in \mathcal{F}$ is a Frechet mean if and only if any vector representation $\boldsymbol{m} \in M$ is of the form

$$\boldsymbol{m} = p \cdot \boldsymbol{x} + (1 - p) \cdot \boldsymbol{x}',$$

where $(\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{A}(X, X')$ is an optimal alignment.

(P_3) A Frechet mean $M \in \mathcal{F}$ satisfies

$$\begin{aligned} d(X, M) &= p \cdot d(X, X') \\ d(X', M) &= (1 - p) \cdot d(X, X'). \end{aligned}$$

(P_4) $|\mathcal{F}| = 1$ with probability one.

Property (P_1) is a direct consequence of Theorem 1 and restated for sake of completeness. Property (P_2) states that the problem of determining an element of the Frechet mean of two structures is equivalent to finding an optimal alignment of X and X'. Properties (P_1) and (P_2) tell us how to construct a weighted mean. Property (P_3) shows that a Frechet mean of two structures is a weighted mean and therefore justifies, for example, the stochastic update rule of central clustering. The last property asserts that the Frechet mean consists of a singleton almost surely. At first this result may seem a useful achievement for a practical setting. A closer look at the proof, however, reveals that for most application problems, the given graphs lie in the set of Lebesgue measure zero for which no statement about uniqueness and non-uniqueness is given.

4 Conclusion

This contribution focused on large sample statistics for distribution on graphs. We derived structural versions of the two key results from probability theory, the law of large numbers and the central limit theory. In addition, we presented a characterization of Frechet means, sufficient conditions for uniqueness of the Frechet mean, and properties of the weighted Frechet mean of two graphs. The key idea to derive the proposed results is based on identifying graphs as points in some Riemannian orbifold. The results generalize corresponding results in Euclidean spaces. In addition, this work establishes a sound statistical basis for unsupervised structural pattern recognition methods such as PCA for structures, central clustering, and graph quantization. Furthermore, we gain new insight into the geometry of the graph domain, which in turn guides us to derive results for statistical and structural pattern analysis of graphs.

References

1. Bhattacharya, R., Bhattacharya, A.: Statistics on Manifolds with Applications to Shape Spaces. In: Perspectives in Mathematical Sciences, ISI, Bangalore (2008)
2. Caetano, T.S., et al.: Learning graph matching. In: ICCV 2007 Conf. Proc., pp. 1–8 (2007)
3. Cour, T., et al.: Balanced graph matching. In: NIPS 2006 Conf. Proc. (2006)

4. Ferrer, M.: Theory and algorithms on the median graph. application to graph-based classification and clustering, PhD Thesis, Univ. Aut‘onoma de Barcelona (2007)
5. Ferrer, M., et al.: Graph-Based k-Means Clustering: A Comparison of the Set Median versus the Generalized Median Graph. In: Jiang, X., Petkov, N. (eds.) CAIP 2009. LNCS, vol. 5702, pp. 342–350. Springer, Heidelberg (2009)
6. Ferrer, M., Valveny, E., Serratosa, F.: Median graphs: A genetic approach based on new theoretical properties. Pattern Recognition 42(9), 2003–2012 (2009)
7. Fréchet, M.: Les éléments aléatoires de nature quelconque dans un espace distancié. Annales de l'Institut Henri Poincaré 10(3), 215–310 (1948)
8. Gold, S., Rangarajan, A.: Graduated Assignment Algorithm for Graph Matching. IEEE Trans. PAMI 18, 377–388 (1996)
9. Gold, S., et al.: Learning with preknowledge: clustering with point and graph matching distance measures. Neural Comp. 8(4), 787–804 (1996)
10. Günter, S., Bunke, H.: Self-organizing map for clustering in the graph domain. Pattern Recognition Letters 23(4), 405–417 (2002)
11. Jain, B., Wysotzki, F.: Central Clustering of Attributed Graphs. Machine Learning 56, 169–207 (2004)
12. Jain, B., Obermayer, K.: On the sample mean of graphs. In: IJCNN 2008 Conf. Proc., pp. 993–1000 (2008)
13. Jain, B., Obermayer, K.: Graph Quantization, arXiv:1001.0921v1 [cs.AI] (2009)
14. Jain, B., et al.: Multiple alignment of contact maps. In: IJCNN 2009 Conf. Proc. (2009)
15. Jain, B., Obermayer, K.: Consistent Estimators of Median and Mean Graph. In: ICPR 2010 Conf. Proc. (2010)
16. Jain, B., Obermayer, K.: Supplementary material for the paper Large Sample Statistics in the Domain of Graphs (2010), http://user.cs.tu-berlin.de/~jbj/publication.html
17. Jiang, X., Munger, A., Bunke, H.: On Median Graphs: Properties, Algorithms, and Applications. IEEE Trans. PAMI 23(10), 1144–1151 (2001)
18. Mukherjee, L., et al.: Generalized median graphs and applications. Journal of Combinatorial Optimization 17, 21–44 (2009)
19. Schenker, A., et al.: Clustering of web documents using a graph model. In: Web Document Analysis: Challenges and Opportunities, pp. 1–16 (2003)
20. Umeyama, S.: An eigendecomposition approach to weighted graph matching problems. IEEE Trans. PAMI 10(5), 695–703 (1988)
21. van der Vaart, A.W.: Asymptotic Statistics. Cambridge University Press, Cambridge (2000)

Analysis of the Multi-Dimensional Scale Saliency Algorithm and Its Application to Texture Categorization

Pablo Suau and Francisco Escolano

Robot Vision Group, University of Alicante, Spain
{pablo,sco}@dccia.ua.es

Abstract. A new approach for multi-dimensional Scale Saliency (MDSS) was lately introduced. In this approach, the Scale Saliency algorithm by Kadir and Brady is extended to the multi-dimensional domain. The MDSS algorithm is based on alternative entropy and divergence estimation methods whose complexity does not increase exponentially with data dimensionality. However, MDSS has not been applied to any practical problem yet. In this paper we apply the MDSS algorithm to the texture categorization problem, and we provide further experiments in order to assess the suitability of different estimators to the algorithm. We also propose a new divergence measure based on the k-d partition algorithm.

1 Introduction

High level vision tasks usually rely on the results provided by image processing or feature extraction algorithms. The interest regions detected by feature extraction methods should satisfy several properties: they must be informative, distinguishable and invariant to a wide range of transformations[1]. The work in this paper is focused on the Scale Saliency algorithm by Kadir and Brady [1]. This algorithm is theoretically sound, due to the fact that it uses Information Theory in order to search the most informative regions on the image. Although its poor performance for matching problems [2], it has been shown to perform well in image categorization tasks [3]. Furthermore, it has been successfully applied before to this kind of problems [4,5].

The Scale Saliency algorithm [1] detects salient or unpredictable regions on an image. Shannon's entropy is used to measure the saliency of an image region. Given a pixel x, its entropy at scale s is computed from the grayscale intensity pdf of the circular region R_x of radius s, centered over x. The intensity pdf is approximated by means of an intensity histogram where $P_{d,s,x}$ is the probability that the intensity value $d \in D$ is found in R_x (in the case of a grayscale image, $D = \{0, \dots, 255\}$).

[1] Several authors prefer the term *covariant*, referring to image features that adapt to the transformation applied to the image.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 698–707, 2010.

The algorithm works as follows: firstly, entropy is estimated for all pixels x in the image, using all scales s in a range of scales between s_{min} and s_{max} (Eq. 1). Next, entropy peaks (local maxima in scale space) are selected (Eq. 2). Then, entropy peaks are weighted by means of a self-dissimilarity metric between scales (Eq. 3). Finally, a subset of the salient features is selected, in order of weighted entropy (Eq. 4). These selected features are the most salient features of the image.

$$H(s,x) = \sum_{d \in D} P_{d,s,x} \log_2 P_{d,s,x} \tag{1}$$

$$S = \{s : H(s-1,x) < H(s,x) > H(s+1,x)\} \tag{2}$$

$$W(s,x) = \frac{s^2}{2s-1} \sum_{d \in D} |P_{d,s,x} - P_{d,s-1,x}| \tag{3}$$

$$Y(s,x) = H(s,x)W(s,x) \ . \tag{4}$$

The application of the algorithm summarized above to higher dimensional data is straightforward. For instance, in RGB color images, where each pixel is assigned three different intensity values (corresponding to the three RGB channels), the local intensity pdf may be estimated from a 3D histogram. In general, for nD data, the same algorithm can be applied if entropy and self-dissimilarity are computed from nD histograms. Two problems arise from this extension to the multi-dimensional domain, due to the curse of dimensionality. Firstly, the complexity order of the algorithm increases exponentially with data dimensionality. And secondly, higher dimensional data yields sparser histograms, that are less informative. These issues make the use of the original Scale Saliency algorithm unfeasible in the case of $n \geq 4$ dimensions.

We previously introduced two extensions of the Scale Saliency algorithm to the multi-dimensional domain, based on entropy and self-dissimilarity (divergence between scales) estimation from entropic graphs [6] and from k-d partitions [7]. Our experiments show that up to 31 dimensions can be processed with MDSS, but i) apart from a repeatability test, we do not provide additional evidence of the suitability of the applied estimators to the Scale Saliency task, ii) the theoretical background of our k-d partition based divergence is not discussed, and iii) no practical application of the MDSS is reported. In this paper we address these three points. Firstly, in Sects. 2 and 3 we summarize the two different MDSS approaches (entropic graphs and k-d partition based), introducing a new k-d partition divergence estimation method. Then, in Sect. 4 we assess these approaches. Finally, in Sect. 5, we apply the MDSS algorithm to the texture categorization problem.

2 MDSS Based on k-Nearest Neighbour Graphs

In this approach, each pixel $x_i \in X$ is represented as a d-dimensional vector. The neighbourhood R_x of a pixel is represented by an undirected and fully connected graph $G = (V, E)$, being the nodes $v_i \in V$ the d-dimensional vectors representing

$x_i \in R_x$ and E the set of edges connecting each pair of nodes. The weight of each edge is the Euclidean distance in $\mathcal{R}^d$ between its two incident nodes. Entropy and divergence are estimated from the K-Nearest Neighbour Graph (KNNG), a subset of the fully connected graph, that connects each node to its k neighbours. From the KNNG, entropy is estimated by means of the measure defined by Kozachenko and Leonenko [8]

$$\hat{H}_{N,k} = \frac{1}{N}\sum_{i=1}^{N} \log\left((N-1)e^{-\psi(k)}B_d(\rho_{k,N-1}^{(i)})^d\right) \ , \tag{5}$$

where $|V| = N$, B_d is the volume of the d-dimensional unit ball, $\rho_{k,N-1}^{(i)}$ is the distance to the k-nearest neighbour of i when taking the rest of $N-1$ samples, and $\psi(z)$ is the digamma function.

In the case of self-dissimilarity between scales, the Friedman-Rafsky test is applied [6]. Let s be the scale in which an entropy peak was found. In order to weight that entropy value, we must calculate the dissimilarity with respect to scale $s-1$. Let X_s and X_{s-1} be the set of nodes of R_x at scales s and $s-1$, respectively. Since $X_{s-1} \subset X_s$ (new pixels are added to the previous ones as we increase the scale), the test only requires to build the KNNG from X_s and to count the amount of edges in this KNNG that connect a node from X_s/X_{s-1} to a node from X_{s-1}. One minus this number of edges is a consistent estimator of the Henze and Penrose divergence.

3 MDSS Based on the k-d Partition Algorithm

The second MDSS approach is based on the k-d partition algorithm by Stowell *et al.* [9]. As in the approach presented above, each pixel in R_x is represented as a d-dimensional vector. The d-dimensional feature space is recursively spit into cells following the data splitting method of the k-d tree algorithm. At each level, the data is spit by their sample median along one axis. Then, data splitting is applied to each subspace until an uniformity stop criterion is reached. The aim of this stop criterion is to produce cells with uniform empirical distribution, in order to best approximate the underlying pdf. The data partition yields a set $A = \{A_j\}$ of p cells, and then entropy estimation is given by

$$\hat{H} = \sum_{j=1}^{p} \frac{n_j}{n} \log\left(\frac{n}{n_j}\mu(A_j)\right) \ , \tag{6}$$

where $\mu(A_j)$ is the volume of the cell A_j, n_j is the number of samples in A_j and n is the the total number of samples in R_x.

Regarding the self-dissimilarity between scales, we propose a new divergence metric inspired by the k-d partition algorithm. Our k-d partition based divergence metric follows the spirit of the total variation distance [10], but may also be interpreted as a L1-norm distance. The total variation distance between two probability measures P and Q in the case of a finite alphabet is given by

$$\delta(P,Q) = \frac{1}{2}\sum_{x} |P(x) - Q(x)| \ . \tag{7}$$

Let $f(x)$ and $g(x)$ be two distributions, from which we gather a set X of n_x samples and a set O of n_o samples, respectively. If we apply the partition scheme of the k-d partition algorithm to the set of samples $X \bigcup O$, the result is a partition A of $X \bigcup O$, being $A = \{A_j | j = 1, \ldots, p\}$. In the case of $f(x)$, the probability of any cell A_j is given by $p(A_j) = \frac{n_{x,j}}{n_x} = p_j$ where $n_{x,j}$ is the number of samples from X in cell A_j. Conversely, in the case of $g(x)$ the probability of each cell A_j is given by $q(A_j) = \frac{n_{o,j}}{n_o} = q_j$ where $n_{o,j}$ is the number of samples from X in the cell A_j. Since both sample sets share the same partition A, and considering the set of cells A_j a finite alphabet, we can compute the total variation distance between $f(x)$ and $g(x)$ as

$$D(O||X) = \frac{1}{2}\sum_{j=1}^{p} |p_j - q_j| \ . \tag{8}$$

The latter distance metric can be used as a self-dissimilarity measure in Scale Saliency algorithm, since it satisfies $0 \leq D(O||X) \leq 1$. The minimum value $D(O||X) = 0$ is obtained when all the cells A_j contain the same proportion of samples from X and O. By the other hand, the maximum value $D(O||X) = 1$ is obtained when all the samples in any cell A_j were gathered from a single distribution.

4 Experimental Results

In this section we introduce additional experiments to those shown in [7]. These experiments in [7] were aimed to compare the computational time of both MDSS approaches and the quality of the extracted features. We demonstrated that the computational order decreased from exponential with respect to data dimensionality (due to the use of histograms in the original Kadir and Brady algorithm) to linear. The computational efficiency of the k-d partition approach is remarkably higher when compared to the rest of algorithms; it can process a 31-dimensional 256 × 256 image in less than four minutes. In the case of the quality of the extracted features, we applied a repeatability test in order to assess the stability of the extracted features over a wide range of transformations, using the image dataset proposed by Mikolajczyk *et al.* [2]. Colour information was used not only in the case of MDSS, but also in the case of the Kadir and Brady Scale Saliency. The results showed that none of the MDSS approaches performs better than the other one in all circumstances. Furthermore, both MDSS algorithms showed lower repeatability that the original Scale Saliency algorithm.

4.1 Entropy Estimation Bias

Firstly we assess the estimation bias of the two entropy estimation methods summarized above, using two types of distributions: Gaussian and uniform. The normal distribution $N(\mu, \sigma^2)$ has maximum entropy among all real-valued distributions with specified mean μ and standard deviation σ [11]. By the other hand, the uniform distribution on the interval $[a, b]$ is the maximum entropy distribution among all continuous distributions which are supported in the interval $[a, b]$ [11]. In both cases we measured the mean deviation (after 100 runs) from the theoretical entropy of the Gaussian and uniform distributions for increasing data dimensionality and a number of samples corresponding to the number of pixels taken by the MDSS algorithm between scales $s_{min} = 3$ and $s_{max} = 30$. The results are shown in Fig. 1. As one may expect, in general the estimation asymptotically improves when increasing the number of samples. Also, in all cases, increasing data dimensionality degrades the entropy estimation. None of the tested estimators performs better in all circumstances. The Leonenko *et al.* estimator approximates better the theoretical entropy of the Gaussian distribution, while the k-d partition estimation approximates better the theoretical entropy of the uniform distribution. It must be also noted that the Leonenko estimator does not require a high value of the parameter k; on the contrary, it yields better results for $k = 2$.

Despite these results, the Scale Saliency algorithm does not require an exact estimation of entropy, as long as the saliency estimator used follows the trend of Shannon's entropy as saliency increases. We performed an additional experiment in order to test the trend of the entropy estimation given by the k-d partition algorithm and the Leonenko *et al.* estimator. The experiment consisted in gathering N samples $x \in [0, 255]^d$ from a Gaussian and an uniform distribution, being N the number of pixels processed at $s_{max} = 30$ during the Scale Saliency algorithm. Then we computed the estimated entropy as we decreased the amount of samples, removing in each iteration the sample which is the furthest from the samples' center of mass and taking the mean after 100 runs. The experiment was repeated for different data dimensionalities. Although the results of the experiment are not shown here due to the lack of space, we summarize them here.

For Gaussian data, the k-d partition algorithm approximates better the trend of histogram based entropy estimation, even in the case of higher data

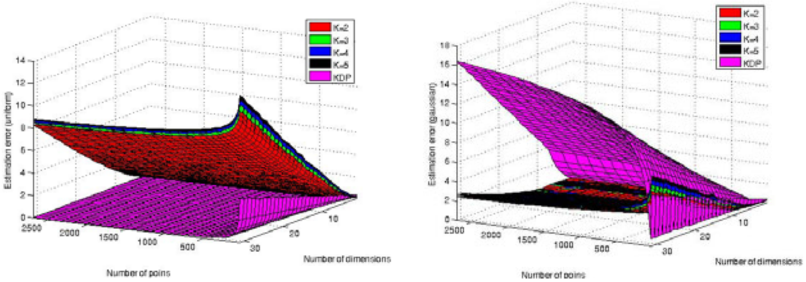


Fig. 1. K-d partition (KDP) and Leonenko *et al.* method (for $k = 2 \ldots 5$) estimation bias for an uniform distribution in the range $[-3, 3]^d$ (left) and a Gaussian distribution with zero mean and $\Sigma = I$ (right)

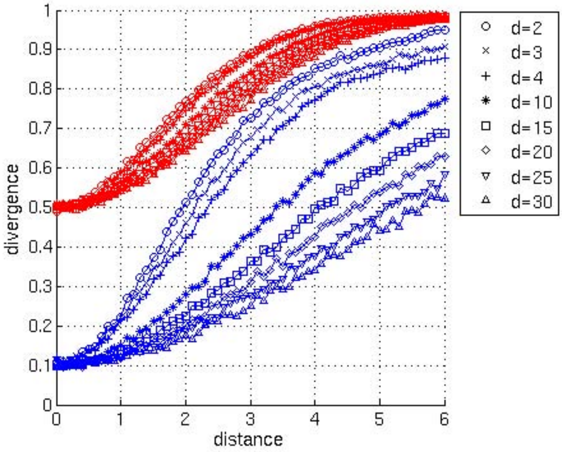


Fig. 2. Divergence estimation results using Friedman-Rafsky test (red) and k-d partition divergence (blue), for different data dimensionalities (d)

dimensionality. From $d = 3$, the Leonenko based estimation soon converges as N increases and, as a consequence, it has less discriminative power. For uniform data both estimators soon reach an asymptote; however, the k-d partition curve still approximates better to the shape of the histogram based curve.

4.2 Divergence Comparison

Now we compare the estimation results of our k-d partition based divergence with those of the Friedman-Rafsky test. Both methods were used to estimate the divergence of two sample sets gathered from two Gaussian distributions, starting with the same mean and variance, as we increased the distance between Gaussian centers until the probability that the samples overlap is low. The experiment was repeated for different data dimensionalities. The results are shown in Fig 2. In both cases, the divergence (y axis) increases with the distance between Gaussian centers (x axis). The values or Friedman-Rafsky test lie in the range $[0.5, 1]$. The range of values in the case of our k-d partition divergence is generally wider, but its results degrade for higher dimensionalities. However, even in the case of $d = 30$, the width of the range of values yielded by the k-d partition based divergence is similar to the one yielded by the Friedman-Rafsky test.

4.3 Number of Features

The amount of detected salient regions may have an effect on the quality and the repeatability of a feature extraction algorithm [2]. In the MDSS and the Kadir and Brady Scale Saliency algorithms we can set the percentage of most salient features to select, but not its final number due to the non-maximum suppression

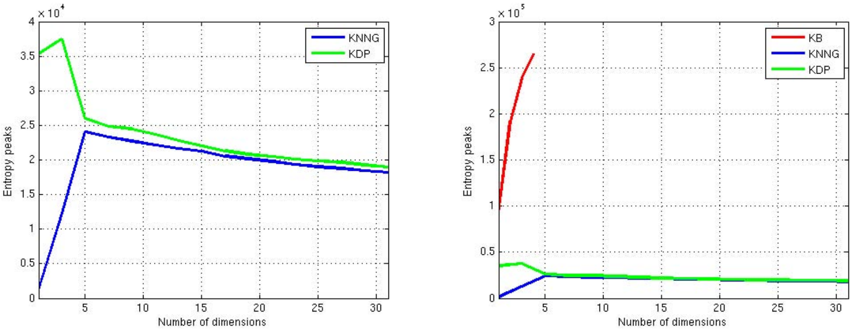


Fig. 3. Mean number of detected entropy peaks during Leonenko based MDSS (KNNG), k-d partition based MDSS (KDP) and Kadir and Brady Scale Saliency (KB) for increasing data dimensionality

step [1]. Thus, rather than performing a comparison of the two MDSS approaches based on the final number of detected features, our comparison was based on the number of entropic peaks found during the algorithm.

In Fig. 3 (left) we show the mean number of entropic peaks found during Leonenko and k-d partition based MDSS algorithms, using the images of the Bristol dataset, as we increased the number of dimensions (the number of layers used for entropy estimation). For a detailed description of the Bristol dataset see [7]. When data dimensionality is low, the results of the k-d partition based MDSS outperform those of the Leonenko based MDSS, providing a higher number of entropic peaks. For $d > 4$, although the number of detected entropic peaks is slightly higher in the case of the k-d partition based MDSS, the results are similar to those of the Leonenko based MDSS. Thus, both estimators may be considered equivalent in terms of number of detected features, and any of them could be applied to MDSS, if only this factor is relevant. In Fig. 3 (right) we compare the results of the MDSS and the Kadir and Brady Scale Saliency algorithms. It is unfeasible to apply the histogram based estimation for $d > 4$ due to the extremely high required computation time; thus, in Fig. 3 we are only showing partial comparison results. As can be seen, the Kadir and Brady Scale Saliency algorithm detects a higher amount of salient features in this range of data dimensions. This fact could be the cause of the better performance of this algorithm in the repeatability experiment in [7].

It must be noted that as the number of dimensions increase, the amount of entropy peaks decrease. This fact imposes a bound on the number of dimensions to which the MDSS can be applied. We tried, for instance, to apply the MDSS algorithm to 128D data images, in which a SIFT descriptor [12] was extracted for each image pixel, using a fixed scale. In most cases, the MDSS did not detect any entropy peak.

4.4 Choosing a MDSS Approach

Given the conclusions extracted during our experiments, and those in [7], the k-d partition entropy estimation algorithm should be preferred over the Leonenko estimator for MDSS implementation. Its computation time is remarkably lower, and it approximates better the trend of the Shannon's entropy for increasing saliency, even for high data dimensionality. Our new k-d partition based divergence also provides better estimation results than the KNNG based approach. The main drawback of both MDSS approaches is the low number of detected salient regions, that can decrease their performance in terms of repeatability. And although MDSS can cope with remarkably higher data dimensionality than the Kadir and Brady Scale Saliency algorithm, a bound on data dimensionality still exits, due to the fact that the number of features decrease as the number of dimensions increase.

5 A MDSS Application: Texture Categorization

In this section we show how the MDSS algorithm can be applied, in conjunction with the Lazebnik *et al.* [13] texture representation, to the to the texture categorization problem. In this problem, each image is showing one texture. We represent each texture image by a signature $S = \{(t_1, w_1), \ldots, (t_n, w_n)\}$, where t_i is a texton and w_i is its relative weight. The steps required to build the signature following the Lazebnik method are: i) firstly, image features are extracted from grayscale intensities of the image, and a descriptor is computed for each feature, ii) agglomerative clustering is applied to all the descriptors of an individual image, and iii) the textons are the center of these clusters, and their relative weight is computed as the number of descriptors in the clusters divided by the total number of descriptors in the image. The obtained signatures can be compared by means of the Earth Mover's Distance (see [13] for more detail). In our case we apply MDSS to build a signature for each texture image from 15D data: all the pixels in the image are processed by means of a Gabor filter bank, consisting of 15 Gabor filters with different orientations and wavelengths. The Kadir and Brady Scale Saliency algorithm can not cope with this high dimensional data.

In Fig. 4 we show the results of our texture retrieval experiment (along with the output of the MDSS and the Scale Saliency algorithms for two example texture images). In this experiment, that shows the performance of a given texture representation, all images in the Brodatz dataset[2] are used as query image once. For each image query, we select images from the database in increasing order of EMD. The result is a plot that shows the average recall of all query images (being recall the number of images from the class of the query image retrieved so far divided by the total number of images in that class) versus the number of closest images retrieved. In Fig. 4 we compared the performance of the grayscale Scale Saliency and k-d partition based MDSS for the case of different descriptors: using only RIFT (*kadirrift* and *kdpeerift*, respectively), only spin

[2] http://www.ux.uis.no/∼tranden/brodatz.html

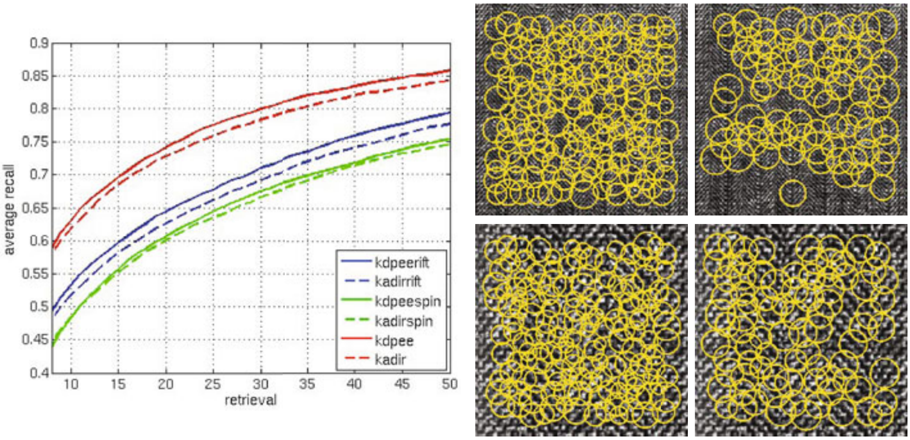


Fig. 4. Left: results of the texture categorization experiment. Right: output of the MDSS algorithm from 15D data (left) and the Scale Saliency algorithm from grayscale intensities (right), for two example texture images. In both cases the 150 most salient features (after non maximum suppression) were selected.

images (*kadirspin* and *kdpeespin*), and combining RIFT and spin images (*kadir* and *kdpee*). For a complete description of the RIFT and spin image descriptors, see [13]. In order to combine RIFT and spin images in the retrieval task, the total distance between two images is computed adding the normalized EMDs estimated for each individual descriptor.

Multi-dimensional data increased the performance of the texture retrieval task for each tested descriptor. However, its impact is not as noticeable as the impact of choosing an adequate descriptor. As can be seen in Fig. 4, the average retrieval is strongly affected by this last factor. The worst results are obtained for the case of spin images. RIFT increases the average recall, but the most significative improvement is achieved when combining both.

6 Conclusions and Future Work

The Scale Saliency algorithm by Kadir and Brady can be easily extended to process multi-dimensional data. However, its computational efficiency remarkably decreases with data dimensionality. We assess two approaches of MDSS based on different entropy and divergence metrics which computational order is linear with respect to data dimensionality. Our analysis shows that the k-d partition approach should be preferred over the graph based approach. We introduced a new divergence estimation method based on the k-d partition algorithm and the total variation distance, and we experimentally demonstrated its suitability. Finally, we showed a practical application of our approach in the context of texture categorization.

Our future work is addressed to evaluate the application of multi-dimensional data in other computer vision problems, like video processing or image retrieval. In the texture categorization context, we should also study the impact of using different Gabor filter banks, or even different input data. This is a combinatorial problem that may be treated with Machine Learning methods like feature selection.

References

1. Kadir, T., Brady, M.: Scale, Saliency and Image Description. Int. J. Comp. Vision 2, 83–105 (2001)
2. Mikolajczyk, K., Tuyelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. Int. J. Comp. Vision 65(1/2), 43–72 (2005)
3. Mikolajczyk, M., Leibe, B., Schiele, B.: Local features for object class recognition. In: Proc 10th IEEE Int. Conf. Comp. Vision, vol. 2, pp. 1792–1799 (2005)
4. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. Int. J. Comp. Vision 79(3), 299–318 (2008)
5. Newman, P., Ho, K.: SLAM-loop closing with visually salient features. In: Proc. IEEE Int. Conf. Rob. Aut., pp. 635–642 (2005)
6. Suau, P., Escolano, F.: Multi-dimensional scale saliency feature extraction based on entropic graphs. In: Proc. of the 4th Int. Symp. on Visual Computing, vol. 2, pp. 170–180 (2008)
7. Suau, P., Escolano, F.: A new feasible approach to multi-dimensional scale saliency. In: Proc. 11th Int. Conf. on Advanced Concepts for Intel. Vision Systems, pp. 77–88 (2009)
8. Kozachenko, L., Leonenko, N.: On statistical estimation of entropy of a random vector. Problems of Inf. Transm. 23, 95–101 (1987)
9. Stowell, D., Plumbley, M.D.: Fast multidimensional entropy estimation by k-d partitioning. IEEE Signal Processing Letters 16(6), 537–540 (2009)
10. Denuit, M., van Bellegenm, S.: On the stop-loss and total variation distances between random sums. Stat. and Prob. Letters 53, 153–165 (2001)
11. Cover, T., Thomas, J.: Elements of Information Theory. John Wiley & Sons, Chichester (1991)
12. Lowe, D.: Object recognition from local scale-invariant features. In: Proc. of the 7th IEEE Int. Conf. on Comp. Vision, vol. 2, pp. 1150–1157 (1999)
13. Lazebnik, S., Schmid, C., Ponce, J.: A sparse texture representation using local affine regions. IEEE Trans. Pat. Anal. Mach. Intel. 27, 1265–1278 (2005)

Interactive Image Retrieval Using Smoothed Nearest Neighbor Estimates*

Miguel Arevalillo-Herráez and Francesc J. Ferri

Departament d'Informàtica, Universitat de València,
Av. Vicent Andrés-Estellés, s/n, 46100 Burjassot (València), Spain
{miguel.arevalillo,francesc.ferri}@uv.es

Abstract. Relevance feedback has been adopted by most recent Content Based Image Retrieval systems to reduce the semantic gap that exists between the subjective similarity among images and the similarity measures computed in a given feature space. Distance-based relevance feedback using nearest neighbors has been recently presented as a good tradeoff between simplicity and performance. In this paper, we analyse some shortages of this technique and propose alternatives that help improving the efficiency of the method in terms of the retrieval precision achieved. The resulting method has been evaluated on several repositories which use different feature sets. The results have been compared to those obtained by the nearest neighbor approach in its standard form, suggesting a better performance.

Keywords: CBIR, image retrieval framework, relevance feedback.

1 Introduction

Content based image retrieval (CBIR) embraces a set of techniques which aim to recover pictures from large image repositories according to the interests of the user. Usually, a CBIR system represents each image in the repository as a set of features (usually related to color, texture and shape), and uses a set of distance functions defined over this feature space to estimate similarity between pictures. In this context, a query is usually composed of one or more sample pictures, and the task of the CBIR system is to retrieve the set of images which best matches this query. Indeed, the performance of such a system depends on both the feature space and the distance function used to estimate the similarity between pictures. In this direction, a large number of features and distance functions have been proposed in the past [1,2,3].

The assumption that subjective or semantic similarity is related to the similarity between low level features is implicit to this way of posing the retrieval problem. But since this does not hold true, the goal of most CBIR techniques is to reduce the existing gap between the semantics induced from the low level

* Work partially funded by Feder, GV and MEC through projects TIN2009-14205-C04-03, ACOMP/2010/287 and Consolider Ingenio 2010 CSD2007-00018.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 708–717, 2010.

features and the real high level meaningful semantics of the image. Relevance feedback has been adopted by most recent CBIR systems to reduce the so-called semantic gap [4]. When relevance feedback is used, the search is considered an iterative process in which the original query is refined interactively, to progressively obtain a more accurate result. At each iteration, the system retrieves a series of images according to a pre-defined similarity measure, and requires user interaction to mark the relevant and non relevant retrievals. This data is used to modify some system parameters and produce a new set of results, repeating the process until a satisfying enough result is obtained. In this context, the relationship between any image in the database and the user's desire is usually expressed in terms of a relevance value, which is aimed at directly reflecting the interest the user may have in the image and is to be refined at each iteration.

A large amount of relevance feedback algorithms use the user's selection to search for global properties which are shared by the relevant samples available at each iteration [4]. From a Pattern Recognition viewpoint, this can be seen as obtaining an appropriate estimate of the probability of (subjective) relevance. Many different approaches exist to model and progressively refine these estimates. But taking into account that this constitutes a small sample problem whose models cannot be reliably established because of the semantic gap, non-parametric distance-based methods using neighbors are particularly appealing in this context and may constitute an appropriate trade off [5,6,7]. The main idea in these methods is to assess relevance using distances to relevant and non-relevant neighbors of a given image. In this paper, we consider several already proposed algorithms and analyse their behavior to identify some major pitfalls and motivatedly arrive at several improvements to the original algorithms.

The remainder of the paper is organized as follows. The next section presents the model used, outlines the assumptions made, and presents the name conventions used throughout the article. Then, the nearest neighbor approach is outlined. After, some problems related to the application of this technique in the field of CBIR are analysed. Later, some improvements that tackle these problems are introduced, proposing an alternative formulation of the approach. Finally, the resulting algorithm is compared to the original one [5,6] and some final conclusions are drawn.

2 Relevance-Guided Interactive Image Retrieval

Let us assume we have a repository of images $\mathcal{X} = \{x_1, \cdots, x_m\}$ conveniently represented in a metric feature space, $\mathcal{F}$, whose associate distance measure is $d : \mathcal{F} \times \mathcal{F} \longrightarrow \mathbb{R}^{\geq 0}$. Usually, in the image retrieval context, the representation space is assumed to be the $D-$dimensional space $\mathbb{R}^D$, which may embrace multiple low level descriptors (e.g. color, texture or shape) and the distance d is constructed by combination from simple distance measures over each descriptor [1].

Let us assume that a particular user is interested in retrieving images from $\mathcal{X}$ related to a particular semantic concept. The user's interest can be modelled in the feature space as a probability function, $p(relevant|x)$, which carries in fact a hidden dependence on the given repository, $\mathcal{X}$.

Single point query approaches assume that this probability function can be appropriately represented by a single (ideal) point $c \in \mathcal{F}$ possibly along with a convenient axis or feature reweighting [8]. This approach can be extended to use a set of representative points $\mathcal{C} = \{c_1 \cdots c_\ell\}$ as in typical multipoint query setups [9].

The goal of the retrieval system at each relevance feedback iteration is to find a set of images from $\mathcal{X}$ that contains as much relevant images as possible using the available information. Single point methods use a distance measure to rank images while multiple point methods usually (linearly) combine rankings to each representative in $\mathcal{C}$.

The available information or user feedback is given by the set of images from $\mathcal{Q} \subset \mathcal{X}$ already seen by the user and marked either as relevant (positive), $\mathcal{Q}^+$ or as non-relevant (negative), $\mathcal{Q}^-$. Both disjoint subsets, $\mathcal{Q}^+$ and $\mathcal{Q}^-$, can be seen as samples corresponding to the distribution functions that determine $p(relevant|x)$ as in a typical two-class classification setting. The problem is that these samples are far from being truly representative both because of the small sample size case and the strong dependences introduced by the way in which new evidence is progressively taken into account.

3 Nearest Neighbor (NN) Approach

Nearest neighbors methods have been extensively used in the context of learning, vision and pattern recognition due to their well-known, convenient and well studied practical and asymptotic behavior [10,11].

In particular, the ratio of fraction of neighbors of a certain kind to the volume of the hypersphere containing them is known to be a good estimate of the corresponding probability distributions [12]. This fact has been used in the context of image retrieval [7] in the particular case of a single nearest neighbor to obtain estimates for the relevant and non-relevant classes as inversely proportional to the volume of the corresponding 1-Neighborhoods, $V_R(d_R(x))$ and $V_N(d_N(x))$, where the subscripts refer to the nearest relevant (R) and non-relevant (N) neighbors, respectively; and d_R and d_N are the corresponding distances to each neighbor from x.

From the separate estimates and obviating the exponent D and some constant terms in the volume formulae, the following expression can be arrived at [7]:

$$p(relevant|x) = \frac{d_N(x)}{d_R(x) + d_N(x)}$$

A simplified version of this estimate already used in this context [5] is given by:

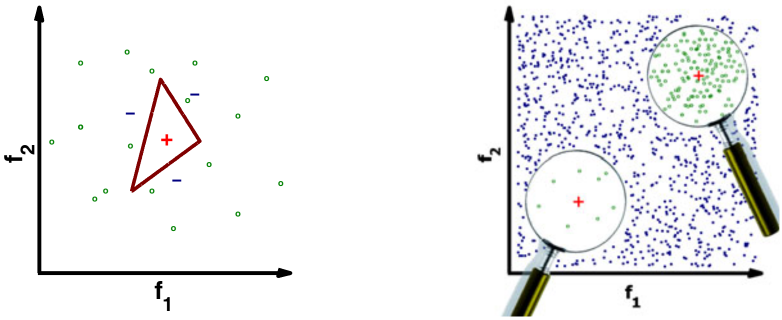
$$p(relevant|x) \propto \frac{d_N(x)}{d_R(x)} \tag{1}$$

where the symbol $\propto$ indicates that a convenient nondecreasing mapping is to be used. On the other hand, it is worth noting that in the image retrieval context what is in fact important is the ranking these estimates induce and not their absolute values. For convenience, normalized relevance scores are used in practice as e.g. $R(x) = 1 - e^{\frac{d_N(x)}{d_R(x)}}$ [5]. Note that all estimates above whether normalized or not, give rise to the same ranking and consequently are equivalent from the point of view of relevance feedback.

4 Considerations about NN Estimates in CBIR

Several problems arise when applying NN estimates in the context image retrieval with relevance feedback. A first problem already reported in [7] comes from the relative sizes of the $\mathcal{Q}^+$ and $\mathcal{Q}^-$ sets. In general, the number of relevant items is by far smaller than the number of non-relevant ones, even in the surroundings of the elements in $\mathcal{C}$. This causes that typically the number of elements in $\mathcal{Q}^+$ be also lower than in $\mathcal{Q}^-$. When a relevant selection is surrounded by non-relevant ones, the above rankings produce high values in a very small closed region around it. But from the images outside this region, the top ranked ones are those which are far from *both* relevant and non-relevant samples. This undesirable effect is illustrated in Figure 1 using the simplest ratio. The chances of this type of situation increase with the relevance feedback iterations, as areas around positive selections tend to be explored more in depth.

Very coupled with the first problem and already identified in [7] is the fact that NN density estimates become very unreliable under the small sample size case. The use of distances to k-th neighbors instead of using $k = 1$ has already been proposed to obtain (slightly) more stable estimates.



(a)

(b)

Fig. 1. Plus signs represent samples that the user has marked as relevant and minus signs those which have been marked as non relevant. Circles represent other existing images. (a) only the pictures within the frontier depicted would yield values of $d_N(x_i)/d_R(x_i)$ above 1. Since there are no images in this area, the most relevant samples may be the farthest from positive and negative samples. (b) When images in the repository (circles) are unevenly populated, some regions may dominate rankings. Top ranked images will be all in the close neighborhood of just one of the relevant sample (the one at the top right corner).

Another important problem not described in previous works is caused by differently populated regions in the feature space. As the ratio of distances is defined in a global way, densely populated regions with high relevance values will tend to dominate the ranking which may result in an undesirable effect in the general case of multipoint query. This problem is caused both by the possibly uneven distribution of images in the repository, $\mathcal{X}$ (as illustrated in Figure 1), and also because of the complex relationship between perceptual similarity and the distance used to find neighbors which may in turn be different in different regions of the feature space. That is, the probability of relevance may scale differently with distance in different regions.

5 Local Searching Using Smoothed NN Estimates

Even with the above mentioned problems, NN-based relevance feedback gives surprisingly good results in practice comparable to other state-of-the-art techniques in most practical cases [7]. Nevertheless and apart from other improvements related to more meaningful or robust representations (e.g. using dissimilarity spaces) or using hybridization techniques (e.g. with Bayesian relevance feedback), there is still room for improvement in the NN approach to relevance feedback itself.

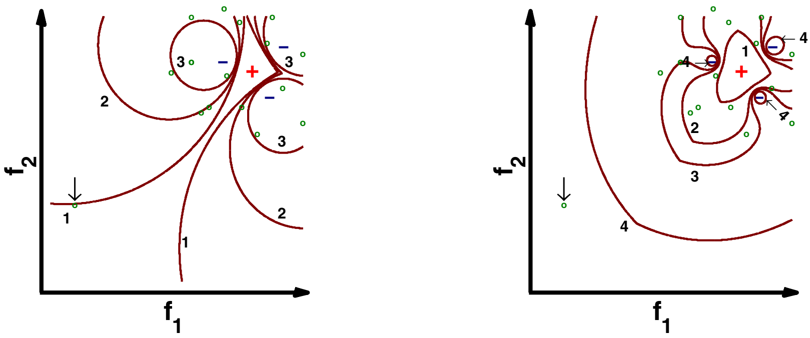
First, a conveniently smoothed NN estimate can be defined by increasing the importance of $\mathcal{Q}^+$. As all previously defined ratios of distances are equivalent with regard to the ranking they induce, we will consider the simple one in Eq. 1. This unnormalized ratio has the advantage of having a simple interpretation in terms of the error rate of the 1-NN classifier [13] and it has also been previously used to derive NN classification rules for imbalanced problems [14].

Using the unnormalized ratio and assuming that the distance to the closest element in $\mathcal{C}$ was available for every picture stored in the repository $\mathcal{X}$, a feasible strategy to smoother estimates would be to introduce a *Moderating Term* (MT) in Eq. 1 directly related to this distance. As these are not available (the elements in $\mathcal{C}$ are the unknowns of the problem that ideally represent the user's desire), a reasonable approximation is to consider the relevant selections in $\mathcal{Q}^+$ instead. To this end, the ratio in Eq. 1 is multiplied by the inverse of the distance to the closest relevant sample. In this simple and parameterless way, those points which are close to any of the elements in $\mathcal{Q}^+$ are rewarded against others which lay farther from them in the feature space. In particular, the following expression is used to compute the relevance scores:

$$R(x) = 1 - e^{-\frac{d_N(x)}{d_R(x)^2}} \tag{2}$$

This expression gives more importance to positive than to negative selections, a consistent approach to deal with the significant differences in the cardinality of the sets of relevant and non relevant selections. The effect of using this smoothed estimate is illustrated in Figure 2.

To deal with the problem of differently populated areas, instead of considering the relevance score, $R(x)$, to produce a global ranking, a set of r local searches



(a) (b)

Fig. 2. Effect of using smoothed and non smoothed estimates. Dots represent images in the repository. Plus and minus signs represent relevant and non relevant selections respectively. The lines are contours of equal score. These have been numbered in decreasing order of relevance. (a) shows these contours using non smoothed estimates; and (b) using smoothed estimates (Eq. 2). In (a), the picture pointed by the arrow would be the closest. In (b) this would be the farthest image of all.

(one per relevant selection q_i^+ in the set $\mathcal{Q}^+$) will be carried out. Each of these searches is performed using equation 2, but considering the picture q_i^+ as the only relevant sample. This results in a set of r independent rankings $R = \{R_1 \cdots R_r\}$, one for each local search. Finally, each picture is assigned a score which is inversely proportional to its best ranking position in the set of rankings R. This technique makes the approach robust against different density areas.

6 Empirical Evaluation

In order to evaluate the impact of the improvements introduced in this work, a comparative analysis of the results obtained with and without them in the original NN approach [5] without any other independent extensions is considered. To evaluate the independent effect of each of the mechanisms proposed we have compared the proposed smoothed NN estimate with local search to those obtained with: (a) the basic NN technique [5]; (b) the NN approach incorporating the local searches (LS) approach; (c) the NN approach adding only the moderation term (MT) technique to handle the surrounding problem. We will refer to these algorithms as the original, the original+LS and the original+MT respectively.

Exhaustive experimentation has been carried out using three well distinct repositories:

- The commercial collection “Art Explosion”, distributed by the company Nova Development. The 10 x 3 HS color histogram and six texture features have been computed for each picture in this database, namely Gabor

Convolution Energies, Gray Level Co-occurrence Matrix, Gaussian Random Markov Fields, the coefficients of fitting the granulometry distribution with a B-spline basis, and two versions of the Spatial Size distribution, one using a horizontal segment and another with a vertical segment [15].

- The subset of the Corel database used in [5]. This is composed of 30 000 images which were manually classified into 71 categories. The descriptors used are those provided in the KDD-UCI[1] repository, namely: A nine component vector with the mean, standard deviation and skewness for each hue, saturation and value in the HSV color space; a 16 component vector with the co-ocurrence in horizontal, vertical and the two diagonal directions; a 32 component vector with the 4×2 color HS histograms for each of the resulting sub-images after one horizontal and one vertical split; and a 32 component vector with the HS histogram for the entire image.
- A small repository which was intentionally assembled for testing, using some images obtained from the Web and others taken by the authors. The 1508 pictures it contains are classified as belonging to 29 different themes such as flowers, horses, paintings, skies, textures, ceramic tiles, buildings, clouds, trees, etc. In this case, the descriptors include a 10 x 3 HS color histogram and texture information in the form of two granulometric cumulative distribution functions [15].

The distances between features have been estimated using the histogram intersection [16] on the color histogram vectors and the Euclidean distance for the other descriptors, and they have been combined as specified in the original publication [5]. In particular, a relevance value is computed for each descriptor and the final score is calculated as a weighted linear combination.

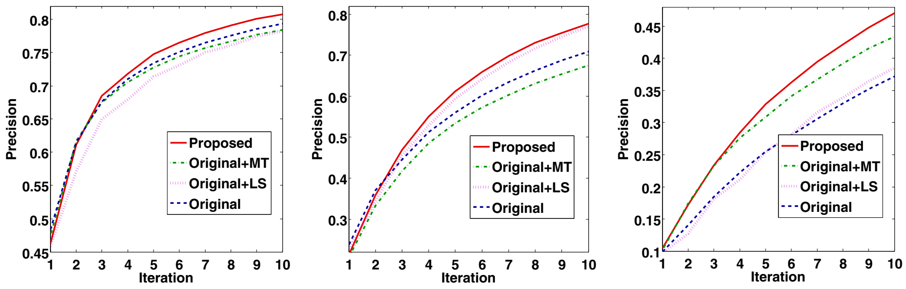
For experimental purposes, a similar setup to that used in [5,7] has been employed. The available categories have been used as concepts, and user judgments about similarity have been simulated considering that only pictures under the same category represent the same concept. To simulate a search, a category is initially chosen at random. At each iteration, automatic judgments are made on the first 50 images, and submitted to the system. Then, the algorithm processes the selection and returns a new image ranking which is judged again as part of an iterative procedure.

To obtain more reliable data, each technique has been evaluated with 500 searches on each repository, using the same categories and initial picture order for all algorithms. In all cases, we have forced the situation that there is at least one relevant sample in between the first 50 images in the initial order in which pictures are presented.

Results have been measured in terms of precision at a cutoff value of 50 (the proportion of relevant picture amongst the top 50 ranked). These results are graphically shown in figure 3.

The results show a significant improvement in retrieval precision when using the proposed technique, incorporating both the moderation term and the local

[1] Available in http://kdd.ics.uci.edu/databases/CorelFeatures



(a) (b) (c)

Fig. 3. Averaged retrieval precision measured on the first 50 retrieved images using a) Web, b) Corel, and c) Art databases

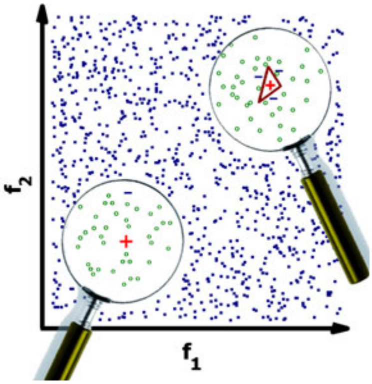


Fig. 4. When a local search strategy is adopted, the surrounded problem becomes more critical. Plus and minus signs represent samples that the user has marked as relevant and non relevant respectively, and each circle a picture in the database. The frontier delimits the area in which the expression $d_N(x_i)/d_R(x_i)$ yields a value greater than one, when a local search with respect to the relevant selection at the top right corner is performed.

searches approaches. Surprisingly enough, when these are used in isolation, worse results are obtained in some cases.

This is mainly due to the tight relation between the solutions to the two problems that we aim to solve. When only the local search strategy is adopted, the surrounding problem becomes more critical. This can easily be understood by considering the case illustrated in Figure 4. In this example, the original algorithm would not retrieve any of the pictures surrounding the positive selection at the top right corner (except the selection itself). However, if two local searches are performed and their results combined, half of the images retrieved (the ones which correspond to the relevant selection at the top right corner) will be those which yield a value of $d_N(x_i)/d_R(x_i)$ just below 1 (the farthest from positive and negative samples), very unlikely to be of any interest. Intuitively, neither option is correct. It is only when the moderation term is also introduced that pictures around the two positive selection are retrieved.

Similarly, if only the moderation term is introduced, the problem about differently populated areas becomes more noticeable. In the original technique, the larger effect of negative selections usually limits the amount of pictures with a large score around a single selection, helping diversity and playing a regulating effect with respect to density. In this sense, the reduction of the importance of non relevant selections caused by the moderating term may have a negative impact on the performance of the algorithm.

It is worth noting that the original NN approach may give even better results at the first and second iterations, e.g. on the Corel database. It is as the relevance feedback iterations progress that the improvements introduced become more and more important in all cases. The proposed method obtains the best results consistently and by a significant difference from the 4th iteration on. The only not significant difference is on Corel database and between the proposed method and the basic NN approach with local searches.

7 Concluding Remarks

We have presented an improved nearest neighbor based algorithm for CBIR. In particular, we have re-formulated the algorithm presented in [5] to make it more robust to differences in the densities of pictures in the feature space and the cardinality of the sets of relevant and non relevant selections. It has been observed that this new formulation allows for a significant increase in retrieval precision with respect to the original approach.

Note that in the formulation of the approach we have made no assumption on the feature space, the distance functions used for retrieval and the method used to combine these functions. Although a simple linear combination of the scores obtained for each descriptor has been used in the experimentation (as in [5]) and the use of a dissimilarity space has been suggested in [7], other strategies are also possible. In particular, combination methods which allow one to construct a single similarity measure from several distance functions (e.g. [17]) have been proposed. The integration of the nearest neighbor method with such approaches would make it possible to compute a single relevance score for each image by direct application of equation 2.

References

1. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys 40(2), 1–60 (2008)
2. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval: State of the art and challenges. ACM Transactions on Multimedia Computing, Communications and Applications 2(1), 1–19 (2006)

3. Smeulders, A., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE transactions on Pattern Analysis and Machine Intellingence 22(12), 1349–1379 (2000)
4. Zhou, X., Huang, T.: Relevance feedback for image retrieval: a comprehensive review. Multimedia systems 8(6), 536–544 (2003)
5. Giacinto, G., Roli, F.: Nearest-prototype relevance feedback for content based image retrieval. In: ICPR 2004: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR 2004), Washington, DC, USA, vol. 2, pp. 989–992. IEEE Computer Society, Los Alamitos (2004)
6. Giacinto, G., Roli, F.: Instance-based relevance feedback for image retrieval. In: Saul, L.K., Bottou, Y.W. (eds.) Advances in Neural Information Processing Systems, vol. 17, pp. 489–496. MIT, Cambridge (2005)
7. Giacinto, G.: A nearest-neighbor approach to relevance feedback in content based image retrieval. In: Proceedings of the 6th ACM international conference on Image and video retrieval (CIVR 2007, pp. 456–463. ACM Press, Amsterdam (2007)
8. Ciocca, G., Schettini, R.: A relevance feedback mechanism for content-based image retrieval. Information processing and management 35(1), 605–632 (1999)
9. Urban, J., Jose, J.M.: Evidence combination for multi-point query learning in content-based image retrieval. In: ISMSE 2004: Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering, pp. 583–586. IEEE Computer Society, Washington (2004)
10. Dasarathy, B. (ed.): Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos (1991)
11. Shakhnarovich, G., Darrell, T., Indyk, P.: Nearest-Neighbor Methods in Learning and Vision: Theory and Practice. In: Neural Information Processing. The MIT Press, Cambridge (2006)
12. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, Hoboken (2000)
13. Paredes, R., Vidal, E.: Learning prototypes and distances: A prototype reduction technique based on nearest neighbor error minimization. Pattern Recognition 39(2), 180–188 (2006)
14. Barandela, R., Valdovinos, R., Sanchez, J., Ferri, F.: The imabalanced training sample problem: Under or over sampling? In: Fred, A., Caelli, T.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.) SSPR&SPR 2004. LNCS, vol. 3138, pp. 806–814. Springer, Heidelberg (2004)
15. Soille, P.: Morphological Image Analysis: Principles and Applications. Springer, Berlin (2003)
16. Swain, M.J., Ballard, D.H.: Color indexing. Int. J. Comput. Vision 7(1), 11–32 (1991)
17. Arevalillo-Herráez, M., Domingo, J., Ferri, F.J.: Combining similarity measures in content-based image retrieval. Pattern Recognition Letters 29(16), 2174–2181 (2008)

Kernel Fusion of Multiple Histogram Descriptors for Robust Face Recognition

Chi-Ho Chan, Josef Kittler, and Muhammad Atif Tahir

Centre for Vision, Speech and Signal Processing,
University of Surrey, United Kingdom
{c.chan,j.kittler,m.tahir}@surrey.ac.uk

Abstract. A multiple kernel fusion method combining two multiresolution histogram face descriptors is proposed to create a powerful representation method for face recognition. The multi resolution histogram descriptors are based on local binary patterns and local phase coding to achieve invariance to various types of image degradation. The multikernel fusion is based on the computationally efficient spectral regression KDA. The proposed face recognition method is evaluated on FRGC 2.0 database yielding very impressive results.

Keywords: Local Binary Pattern, Local Phase Quantization, Kernel, Fusion, Linear Discriminant Analysis.

1 Introduction

Recognising faces under uncontrolled lighting conditions and blur either due to misfocus or motion is one of the most important challenges for practical face recognition systems. The problem is aggravated by a high dimensionality of the face data and a small sample size.

Most previous works on face recognition make use of raw image data as input to a linear transformation which maps the image to a point in a space, called face subspace. This point is defined by the coefficients of the face image projection into the associated bases, exemplified by [1][2], Eigenface [3] and Fisherface [4]. However, the performance of such methods degrades when the cropped face image is acquired in changing illumination or is degraded by blur.

In contrast, histogram-based features, such as the Local binary pattern histogram (LBPH)[5], Local Phase Quantisation histogram (LPQH)[6] and the histogram of Gabor Phase Patterns (HGPP) [7], have gained reputation as powerful and attractive texture descriptors showing excellent results in terms of accuracy and computational complexity in face recognition, as these features, which capture the information about the spatial relation of facial regions, are partially invariant to these degradation. In these methods, the face image is first partitioned into a large number of small regions from which pattern histograms, representing the local texture of face images, are extracted. The recognition is performed using the nearest-neighbour classifier. Chan et al. have extended the LBP histogram[8] and LPQ histogram[9] methods to provide a multiresolution

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 718–727, 2010.

representation which further exhibits robustness to face misalignment. These extensions have been demonstrated to achieve excellent results in Feret, XM2VTS and BANCA databases. It is well known that multiple cues enrich the representation of any object. This has been demonstrated also for faces. Face representations derived from the complementary sources of information presented in Table 1 have been shown to achieve better performance than single best representation. It is evident from Table 1 that with the exception of [10] the reviewed fusion studies always involve integrating information emanating from different image domains rather than using different face representations which are derived from a single domain. Moreover, most systems in Table 1 apply either a score fusion or a feature level fusion method. It is therefore of interest to investigate a novel mechanism -multiple kernel fusion- for combining different face representations computed from an intensity image. We focus on the an intensity domain, as it is more robust to changes in image acquisition conditions (camera, illumination).

Table 1. Summary of the fusion methods in different face recognition systems

Ref.	Image Domain	Face representation	Fusion Method
[11]	Infrared and Visible Images	Wavelet	Feature fusion
[12]	2.5D, Curvature, Visible Images	Gabor	Kernel fusion
[13]	Color Image	LBPH	Feature and Score fusions
[10]	Intensity Image	Gabor, LBPH	Score fusion
[14]	Color Image	Frequency feature	Score fusion
[15]	Color Image	Gabor, MLBPH, Frequency Feature	Score fusion
[16]	Global, intrinsic faces	Frequency Feature	Score fusion
[17]	Global, intrinsic faces	Gabor, Frequency Feature	Score fusion

This paper presents a computational and statistical framework for integrating two different descriptors, Multiscale LBPH and Multiscale LPQH for face recognition in 2D grey-scale image domain. These descriptors are selected because of their invariance to monotonic illumination changes and blur. The framework relies on the use of kernel-based statistical learning methods. These methods represent the data by means of a kernel function which is the non-linear function of similarities between pairs of face descriptors. One of the reason for the success of kernel methods is that the kernel function measures the similarity between query face image descriptors and those derived from the training set in an implicitly infinitely dimensional space. Each kernel therefore extracts a specific type of information from the training set, thus providing a partial description or view of the query image. A unique combined kernel obtained from the individual kernels formed by the two descriptors is then projected into the Fisher space for face recognition. Paper is organised as follows. In Section 2 we introduce the image descriptors adopted, as well as a computationally efficient kernel matching method. The problem of fusion is discussed in Section 3. The experimental results are presented in Section 4, leading to conclusions in Section 5.

2 Histogram Features

Local Binary Pattern

The LBP operator, shown in Equation 1, extracts information which is invariant to local monotonic grey-scale variations of the image. During the LBP operation, the value of current pixel, f_c, is applied as a threshold to each of the neighbours, $f_p(p = 0, \cdots, P-1)$ to obtain a binary number. A local binary pattern is obtained by first concatenating these binary bits and then converting the sequence into the decimal number. Using circular neighbourhoods and linearly interpolating the pixel values allows the choice of any radius, R, and number of pixels in the neighbourhood, P, to form an operator

$$LBP_{P,R}(\mathbf{x}) = \sum_{p=0}^{P-1} s(f_p - f_c)2^P \quad | \quad s(v) = \begin{cases} 1 & v \geq 0 \\ 0 & v < 0 \end{cases} \tag{1}$$

Local Phase Quantisation Pattern

The local phase quantisation(LPQ)[6] pattern is robust to blur effects. The phase information of LPQ can be extracted using the two dimensional windowed Fourier transform (2DWFT).

$$\mathbf{F_u}(\mathbf{x}) = \sum_{\mathbf{m} \in \mathcal{N}_x} \mathbf{w}(\mathbf{m} - \mathbf{x})\mathbf{f}(\mathbf{m})e^{-j2\pi\mathbf{u}^T\mathbf{m}} = \mathbf{E}_u^T \mathbf{f_x} \tag{2}$$

where $\mathbf{E}_u$, size $= 1 \times z^2$, is a basis vector of 2DWFT with frequency $\mathbf{u}$, and $\mathbf{f_x}$, size$= z^2 \times N$, is a vector containing image pixel values in $\mathcal{N}_x$ at each $\mathbf{x}$ location. The window function, $\mathbf{w}(\mathbf{x})$ is a rectangular function in this work. The transform is computed at four frequency points, $\mathbf{u} = [\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]$ where $\mathbf{u}_0 = [a, 0]^T$, $\mathbf{u}_1 = [0, a]^T$,$\mathbf{u}_2 = [a, a]^T$ and $\mathbf{u}_3 = [a, -a]^T$. a is a highest scalar frequency for which $\mathbf{W}_{\mathbf{u}_i} > 0$. Thus, only four exponential complex functions are needed as a filter bank to yield eight resultant complex images consisting of 4 filtered images of the real part and 4 images of the imaginary part of the transform. A whitening transform[6] is applied to decorrelate $\mathbf{F_u}(\mathbf{x})$ to improve the system performance. Each pixel of the resultant complex image can be encoded into a binary value shown in Equation (3) by applying the quadrant bit coding.

$$\begin{aligned} \mathbf{B}_{\mathbf{u}_i}^{Re}(\mathbf{x}) &= \begin{cases} 1 & \text{if } \mathbf{F}_{\mathbf{u}_i}^{Re}(\mathbf{x}) > 0 \\ 0 & \text{if } \mathbf{F}_{\mathbf{u}_i}^{Re}(\mathbf{x}) \leq 0 \end{cases} \\ \mathbf{B}_{\mathbf{u}_i}^{Im}(\mathbf{x}) &= \begin{cases} 1 & \text{if } \mathbf{F}_{\mathbf{u}_i}^{Im}(\mathbf{x}) > 0 \\ 0 & \text{if } \mathbf{F}_{\mathbf{u}_i}^{Im}(\mathbf{x}) \leq 0 \end{cases} \end{aligned} \tag{3}$$

This coding method assigns 2 bits for every pixel to represent the quadrant in which the phase angle lies. In fact, it also provides the quantisation of the Fourier phase feature. LPQ is a binary string obtained, for each pixel, by concatenating the real and imaginary quadrant-bit codes of the eight Fourier coefficients of $\mathbf{u}_i$. The binary string is then converted to the decimal number by Equation (4) to produce a LPQ pattern

$$\begin{aligned}\mathbf{LPQ}(\mathbf{x}) = \mathbf{B}^{Re}_{\mathbf{u}_0}(\mathbf{x}) + \mathbf{B}^{Im}_{\mathbf{u}_0}(\mathbf{x}) \times 2^1 + \cdots \\ + \mathbf{B}^{Re}_{\mathbf{u}_3}(\mathbf{x}) \times 2^{k-1} + \mathbf{B}^{Im}_{\mathbf{u}_3}(\mathbf{x}) \times 2^k\end{aligned} \tag{4}$$

In digital image processing, blur effects can be modelled by a discrete linear relationship defined by a convolution between the image intensity and a point spread function (PSF). In the Fourier transform, the phase of each harmonic of the blurred image is the sum of the phase of the original image and phase of the PSF. If the PSF of blur is a positive even function, it will act as a zero-phase low-pass filter. In other words, the LPQ representation is invariant to blur if the cut-off frequency of blur (PSF) is greater than that of the LPQ filter.

Multiscale Pattern Histogram

A multiresolution representation can be obtained by varying the filter size, $z \times z$, and combining the resulting pattern images. Such a representation [9][8] has been suggested for face recognition and the results reported for this application show that the accuracy is better than that of a single scale pattern method. As a multiresolution representation defined by a set of pattern operators of different filter size may give an unstable result because of noise, this problem can be minimised by using aggregate statistics, exemplified by histogram. There are several advantages in summarising the patterns in the form of histogram. First, the statistical summary can reduce the feature dimension from the image size to the number of histogram bins. Secondly, using histogram as a set of features is robust to image translation and rotation to a certain extent and therefore the sensitivity to mis-registration is reduced. Finally, although the effect of unstable pattern responses due to noise is attenuated by histogramming, it can further be reduced by controlling the number of histogram bins and /or projecting the histogram to other spaces.

In our approach, pattern operators defined on a neighbourhood Q, for an instance LBP or LPQ, at R scales, are first applied to a face image. This generates a grey level code for each pixel at every resolution. The resulting Pattern images are cropped to the same size and divided into non-overlapping sub-regions, $\mathbf{M}_0$, $\mathbf{M}_1$,..$\mathbf{M}_{J-1}$. The regional pattern histogram for each scale is computed as

$$\begin{aligned}\mathbf{h}_{r,j}(i) &= \sum_{\mathbf{x} \in \mathbf{M}_j} E(Q_r(\mathbf{x}) = i) \\ &\mid \quad i \in [0, L-1], j \in [0, J-1], \\ &\quad r \in [1, R], z = r \times 2 + 1 \\ E(v) &= \begin{cases} 1 & \text{when } v \text{ is true} \\ 0 & \text{otherwise} \end{cases}\end{aligned} \tag{5}$$

$E(v)$ is a Boolean indicator. r is the scale index and z is the width or height of the pattern filter. The set of histograms computed at different scales for each region $\mathbf{M}_j$ provides regional information. L is the number of histogram bins. By concatenating these histograms into a single vector, we obtain the final multiresolution regional face descriptor.

$$\mathbf{k}_j = [\mathbf{h}_{1,j}, \mathbf{h}_{2,j}, \cdots, \mathbf{h}_{R,j}] \tag{6}$$

KDA Using Spectral Regression (SR-KDA)

Kernel Discriminant Analysis is a non-linear extension of LDA which maps the original measurements into a higher dimensional space using the "kernel trick". If ν denotes a projective function into the kernel feature space, then the objective function for KDA is

$$\max_{\nu} \mathcal{J}(\nu) = \frac{\nu^T \mathbf{C}_b \nu}{\nu^T \mathbf{C}_t \nu} \tag{7}$$

where $\mathbf{C}_b$ and $\mathbf{C}_t$ denote the between-class and total scatter matrices in the feature space respectively. A solution to Equation 7 leads to the eigenvalue analysis problem $\mathbf{C}_b = \lambda \mathbf{C}_t$. It is proved in [18] that equation 7 is equivalent to

$$\max_{\alpha} \mathcal{J}(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{KAKw}}{\mathbf{w}^T \mathbf{KKw}} \tag{8}$$

where $\mathbf{w} = [\alpha_1, \alpha_2, \cdots, \alpha_m]^T$ is the eigen-vector satisfying $\mathbf{KAKw} = \lambda \mathbf{KKw}$. $\mathbf{A} = (\mathbf{A}_l)_{l=1,\cdots,n}$ is a $(m \times m)$ block diagonal matrix of labels arranged such that the upper block corresponds to positive examples and the lower one to negative examples of the class. $\mathbf{K}$ is an $m \times m$ kernel matrix such that $K(\mathbf{k}^{s_1}, \mathbf{k}^{s_2}) = \langle \Phi(\mathbf{k}^{s_1}), \Phi(\mathbf{k}^{s_2}) \rangle$, where $\Phi(\mathbf{k}^{s_1})$ and $\Phi(\mathbf{k}^{s_2})$ are the embeddings of data items $\mathbf{k}^{s_1}$ and $\mathbf{k}^{s_2}$. Each eigenvector $\mathbf{w}$ gives a projection function ν into the feature space.

It is shown in [19] that instead of solving the eigen-problem in KDA, the KDA projections can be obtained by the following two linear equations

$$\begin{aligned} \mathbf{A}\phi &= \lambda\phi \\ (\mathbf{K} + \delta\mathbf{I})\mathbf{w} &= \phi \end{aligned} \tag{9}$$

where ϕ is an eigenvector of $\mathbf{A}$, $\mathbf{I}$ is the identity matrix and $\delta > 0$ is a regularisation parameter. Eigen-vectors ϕ are obtained directly from the Gram-Schmidt method. Since $(\mathbf{K}+\delta\mathbf{I})$ is positive definite, the Cholesky decomposition, $(\mathbf{K} + \delta\mathbf{I}) = \mathbf{R}^T\mathbf{R}$ is used to solve the linear equations in Equation 9 and the obtained result, R is a upper triangular matrix. Thus, the solution of the linear system becomes

$$(\mathbf{K} + \delta\mathbf{I})\mathbf{w} = \phi \Leftrightarrow \begin{cases} \mathbf{R}^T\theta = \phi \\ \mathbf{Rw} = \theta \end{cases} \tag{10}$$

i.e., first solve the system to find vector θ and then vector $\mathbf{w}$. In summary, SRKDA, $\mathbf{W}^{kda} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_{c-1}]$, only needs to solve a set of regularised regression problems and there is no eigenvector computation involved. This results in great improvement of computational cost and allows to handle large kernel matrices.

Complexity Analysis. The computation of SR-KDA involves two steps: (i) response generation which is the cost of the Gram-Schmidt method, and (ii) regularised regression which involves solving $(c-1)$ linear equations using the Cholesky decomposition where c is the number of classes. As in [20], we use the term flam, a compound operation consisting of one addition and one multiplication, to measure the operation counts. The cost of the Gram-Schmidt method requires $(mc^2 - \frac{1}{3}c^3)$ flams. The Cholesky decomposition requires $\frac{1}{6}m^3$ flams and the $c-1$ linear equations can be solved with m^2c flams. Thus, the computational cost of SRKDA excluding the cost of Kernel Matrix $\mathbf{K}$ is $\frac{1}{6}m^3 + m^2c + mc^2 - \frac{1}{3}c^3$ which can be approximated as $\frac{1}{6}m^3 + m^2c$. Comparing to the cost of ordinary KDA $(\frac{9}{2}m^3 + m^2c)$, SR-KDA significantly reduces the dominant part and achieves an order of magnitude (27 times) speed-up.

3 System Fusion

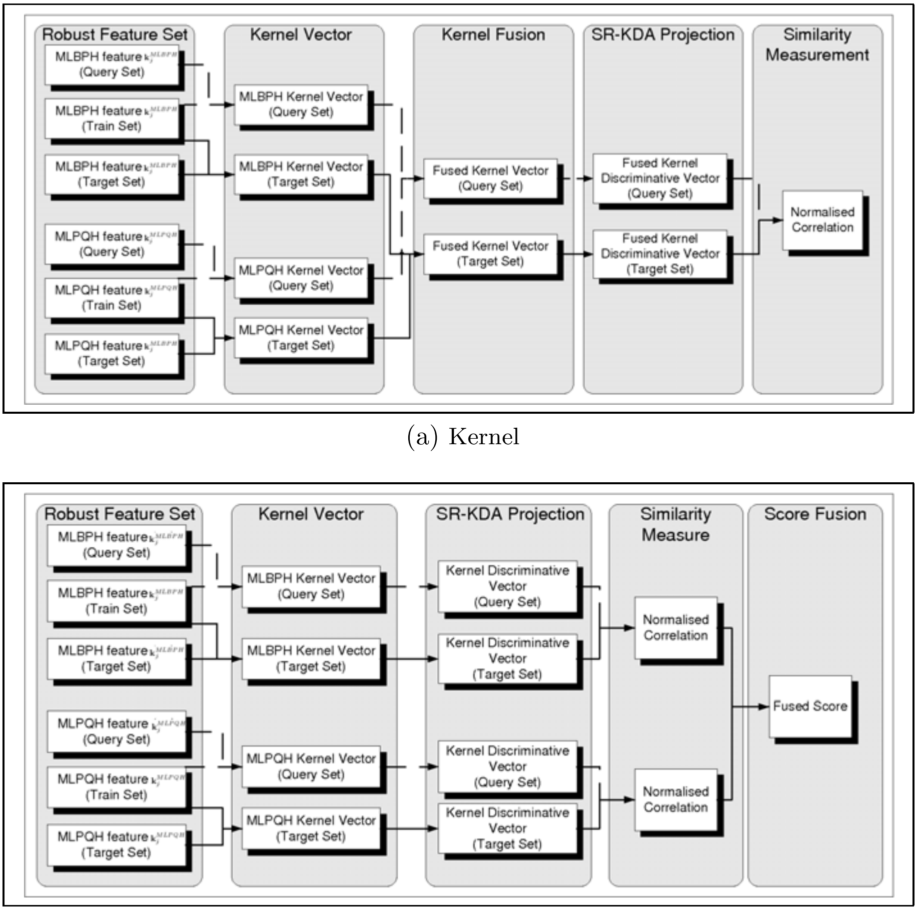
We investigate two frameworks for information fusion: Score level fusion and Kernel level fusion as shown in Figure 1.

Score level fusion: In the case of score-level fusion, for each representation the face recognition system is trained individually using SR-KDA. The output from each classifier is then combined using the sum rule.

Kernel level fusion: Given multiple features (MLPQ, MLBP), each kernel function produces a square matrix in which each entry encodes a particular notion of similarity of one face to another. This kernel formalism also allows these multiple features to be combined. Basic algebraic operations such as addition maintain the key property of positive semi-definiteness and thus allow a simple but powerful algebra of kernels. For example, it is possible to combine kernels computed from MLPQ and MLBP such that kernel $\mathbf{K} = \mathbf{K}_{MLPQH} + \mathbf{K}_{MLBPH}$. Once the kernels are combined, SR-KDA is then applied for feature extraction. It should be noted that this Kernel-level fusion has a speed advantage over the score-level fusion as only one classifier is required. In contrast, for score-level fusion, separate classifiers are required for the individual face representations.

4 Experimental Result

The Face Recognition Grand Challenge version 2 data set is used to evaluate the proposed framework. The faces of this database collected in controlled and uncontrolled environments are divided into training and test sets. The training set contains 12,775 images from 222 subjects, while the test set data contains 24,042 images from 466 in which 222 subjects are common to the training set but their image are not shared with the training set. We focus on experiments, EXP 1 and 4, in this work. EXP 1 is designed to measure the performance


Robust Feature Set
MLBPH feature (Query Set)
MLBPH feature (Train Set)
MLBPH feature (Target Set)
MLPQH feature (Query Set)
MLPQH feature (Train Set)
MLPQH feature (Target Set)
Kernel Vector
MLBPH Kernel Vector (Query Set)
MLBPH Kernel Vector (Target Set)
MLPQH Kernel Vector (Query Set)
MLPQH Kernel Vector (Target Set)
Kernel Fusion
Fused Kernel Vector (Query Set)
Fused Kernel Vector (Target Set)
SR-KDA Projection
Fused Kernel Discriminative Vector (Query Set)
Fused Kernel Discriminative Vector (Target Set)
Similarity Measurement
Normalised Correlation
Robust Feature Set
MLBPH feature (Query Set)
MLBPH feature (Train Set)
MLBPH feature (Target Set)
MLPQH feature (Query Set)
MLPQH feature (Train Set)
MLPQH feature (Target Set)
Kernel Vector
MLBPH Kernel Vector (Query Set)
MLBPH Kernel Vector (Target Set)
MLPQH Kernel Vector (Query Set)
MLPQH Kernel Vector (Target Set)
SR-KDA Projection
Kernel Discriminative Vector (Query Set)
Kernel Discriminative Vector (Target Set)
Kernel Discriminative Vector (Query Set)
Kernel Discriminative Vector (Target Set)
Similarity Measure
Normalised Correlation
Normalised Correlation
Score Fusion
Fused Score

(a) Kernel

(b) Score

Fig. 1. Block diagrams of Kernel fusion and score fusion methods

of face verification from frontal images taken under controlled illumination. In EXP 1, 16,028 images from 466 subjects are used to establish $16,028 \times 16,028$ similarity confusion matrix. EXP 4 is designed to measure the performance on controlled versus uncontrolled frontal face still images. The target set consists of 16,028 controlled images and the query set contains 8,014 uncontrolled still images. The ROC curve plotting the Face Verification rate (FVR) versus the False Acceptance Rate (FAR) is generated using the Biometric Experimentation Environment (BEE) evaluation tool. It produces three ROC curves (ROC I, II & III) corresponding to the images acquired within semesters, within a year, and between semesters, respectively.

Face images are extracted with the ground-truth annotated eye positions and scaled to a size of 142×120 ($rows \times columns$). The cropped faces are photometrically normalised by the Preprocessing sequence approach(PS) [10]. This

Table 2. The verification rate in % at 0.1% FAR for different methods on FRGC 2.0 Experiment 1 and 5

System	EXP 1			EXP 4		
	ROC I	ROC II	ROC III	ROC I	ROC II	ROC III
PS_MLBP+LDA	97.11	96.31	95.40	67.32	68.92	70.51
PS_MLBP+KDA(ED)	98.37	97.91	97.42	**77.85**	**79.68**	**81.53**
PS_MLBP+KDA(X2)	**98.48**	**98.03**	**97.55**	75.45	77.67	79.81
PS_MLPQ+LDA	97.30	96.51	95.67	67.32	68.86	70.50
PS_MLPQ+KDA(ED)	**98.76**	**98.39**	**97.98**	**81.05**	**82.44**	**83.80**
PS_MLPQ+KDA(X2)	**98.76**	98.37	97.94	78.27	79.98	81.54
PS_MLPQ+LDA+PS_MLBP+LDA	97.45	96.69	95.89	69.73	71.59	73.43
PS_MLPQ+KDA(ED)+PS_MLBP+KDA(ED)	98.70	98.31	97.88	**81.17**	**82.84**	**84.42**
PS_MLPQ+KDA(X2)+PS_MLBP+KDA(X2)	**98.77**	**98.38**	**97.95**	79.15	81.04	82.87
PS_MLPQ+PS_MLBP+KDA(ED)	98.84	98.50	98.13	**82.92**	**84.34**	**85.72**
PS_MLPQ+PS_MLBP+KDA(X2)	**98.88**	**98.54**	**98.16**	80.50	82.19	83.78
CVPR06'[16]	95.01	93.29	91.51	75.70	75.06	74.33
ICCV07'[17]			98.00			86.00
AMFG07'[22]						83.60
INNS09'[13]						83.4
TIP08' [14]				79.40	79.90	80.30
ICB09' [15]						92.40
CVPR05'LBP+KLDA(X2) [23]	97.40					
CVPR05'LBP [23]	79.90					
CVPR05'KLDA [23]	82.90					
PCA_Baseline			74.76	70.53	66.05	12.00

photometric normalisation method is designed to reduce the effects of illumination variation, local shadowing and highlights, while still keeping the essential visual appearance information for the use in recognition. Our objective is to evaluate the MLPQH and MLBPH descriptors and their combination. For MLBPH, ten LBP operators from $r = 1$ to 10 with $P = 8$ are employed to represent the face image, while eight LPQ operators from $z = 3$ to 17 for MLPQ. The coded images are then divided into 9 non-overlapping regions and the kernel vectors based on the local histograms are generated in the testing stage. In this work, we have used RBF kernel with Chi-squared (X2) and Euclidean distance (ED) metrics: $K(\boldsymbol{k}^{\boldsymbol{s1}}, \boldsymbol{k}^{\boldsymbol{s2}}) = e^{-\frac{1}{A}dist(\boldsymbol{k}^{\boldsymbol{s1}}, \boldsymbol{k}^{\boldsymbol{s2}})}$ where A is a scalar which normalises the distances. Following [21], A is set to the average Chi-squared or Euclidean distance between all elements of the kernel matrix. The default value of regularisation parameter $\delta = 0.01$ is used in all experiments.

In kernel fusion (MLBPH+MLPQH_KDA), the kernel vectors of MLBPH and MLPQH in each region are fused together and projected into SRKDA space to represent the regional discriminative facial descriptors. The final similarity score is obtained by summing the similarity, i.e. normalized correlation, of regional discriminative descriptors. On the other hand, in Score level fusion (MLBPH_KDA+MLPQH_KDA), SRKDA is applied to each of histogram descriptors and then the similarity score is fused by averaging the similarity scores. For the benchmark systems, the score level fusion of LDA version of MLBPH and

MLPQH (MLBPH_LDA+MLPQH_LDA), MLPQH_LDA and MLBPH_LDA are evaluated and the state of art methods are also reported in Table 2.

Compared to Linear Discriminant analysis-based systems, the Kernel Discriminant analysis performs significantly better in EXP 1. However, there is no significant difference between the performance of the RBF kernel with Chi-squared (X2) and Euclidean distance (ED) metrics. As expected, the performance obtained when combining two different face representations is better than the performance of the individual representation, except for PS_MLPQ+KDA(ED) + PS_MLBP+KDA(ED) in EXP 1. Kernel fusion always outperforms score level fusion. Our proposed frameworks using kernel fusion to combine two different face representations achieves slightly better performance than the system combining the scores from global and intrinsic face images[17]. However, the result of our proposed method is not better than the method in ICB2009 [15] where this complicated method integrating different face representations, such as LBP, Gabor and Fourier features in colour domain achieves better performance. Nevertheless, we argue that any system using colour may not be robust in the real environment and also has a heavy computational cost.

5 Conclusions

We have presented a kernel fusion method for integrating two new robust descriptors for face recognition under uncontrolled lighting conditions and blur. Tested on the challenging FRGC 2.0 database, our proposed framework achieves better performance than the score level fusion. It also outperforms all state of the art method in comparable conditions. The proposed method provides an alternative solution for integrating the descriptors together to achieve robust performance.

Acknowledgements. This work is supported by the EU-funded Mobio project grant IST-214324,(www.mobioproject.org).

References

1. Wiskott, L., Fellous, J.M., Krüger, N., von der Malsburg, C.: Face recognition by elastic bunch graph matching. PAMI 19(7), 775–779 (1997)
2. Zhang, W., Shan, S., Gao, W., Chen, X., Zhang, H.: Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition. In: ICCV, pp. 786–791 (2005)
3. Turk, M.A., Pentland, A.: Face recognition using eigenfaces. In: CVPR, pp. 586–591 (1991)
4. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. fisherfaces: recognition using class specific linear projection. PAMI 19(7), 711–720 (1997)
5. Ahonen, T., Hadid, A., Pietikäinen, M.: Face recognition with local binary patterns. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 469–481. Springer, Heidelberg (2004)
6. Ahonen, T., Rahtu, E., Ojansivu, V., Heikkilä, J.: Recognition of blurred faces using local phase quantization. In: ICPR, pp. 1–4 (2008)

7. Zhang, B., Shan, S., Chen, X., Gao, W.: Histogram of gabor phase patterns (hgpp): A novel object representation approach for face recognition. IEEE Transactions on Image Processing 16(1), 57–68 (2007)
8. Chan, C., Kittler, J., Messer, K.: Multi-scale local binary pattern histograms for face recognition. In: Lee, S.-W., Li, S.Z. (eds.) ICB 2007. LNCS, vol. 4642, pp. 809–818. Springer, Heidelberg (2007)
9. Chan, C., Kittler, J., Poh, N., Ahonen, T., Pietikäinen, M. (multiscale) local phase quantization histogram discriminant analysis with score normalisation for robust face recognition. In: VOEC, pp. 633–640 (2009)
10. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. In: Zhou, S.K., Zhao, W., Tang, X., Gong, S. (eds.) AMFG 2007. LNCS, vol. 4778, pp. 168–182. Springer, Heidelberg (2007)
11. Singh, S., Gyaourova, A., Bebis, G., Pavlidis, I.: Infrared and visible image fusion for face recognition, vol. 5404, pp. 585–596. SPIE (2004)
12. de Diego, I.M., Serrano, Á., Conde, C., Cabello, E.: Face verification with a kernel fusion method. Pattern Recognition Letters (2010)
13. Liu, Z., Tao, Q.: Face recognition using new image representations. In: IEEE - INNS - ENNS International Joint Conference on Neural Networks, pp. 1871–1876 (2009)
14. Liu, Z., Liu, C.: A hybrid color and frequency features method for face recognition. IEEE Transactions on Image Processing 17(10), 1975–1980 (2008)
15. Liu, Z., Liu, C.: Robust face recognition using color information. In: Tistarelli, M., Nixon, M.S. (eds.) ICB 2009. LNCS, vol. 5558, pp. 122–131. Springer, Heidelberg (2009)
16. Hwang, W., Park, G., Lee, J., Kee, S.C.: Multiple face model of hybrid fourier feature for large face image set. In: CVPR, pp. 1574–1581 (2006)
17. Su, Y., Shan, S., Chen, X., Gao, W.: Hierarchical ensemble of global and local classifiers for face recognition. In: ICCV, October 2007, pp. 1–8 (2007)
18. Baudat, G., Anouar, F.: Generalized discriminant analysis using a kernel approach. Neural Computation 2(12), 2385–2404 (2000)
19. Cai, D., He, X., Han, J.: Efficient kernel discriminat analysis via spectral regression. In: Proceedings of the International Conference on Data Mining (2007)
20. Stewart, G.W.: Matrix Algorithms Volume I: Basic Decomposition. SIAM, Philadelphia (1998)
21. Zhang, J., Marszałek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: a comprehensive study. International Journal of Computer Vision 73(2), 213–238 (2007)
22. Tan, X., Triggs, B.: Fusing gabor and lbp feature sets for kernel-based face recognition. In: Zhou, S.K., Zhao, W., Tang, X., Gong, S. (eds.) AMFG 2007. LNCS, vol. 4778, pp. 235–249. Springer, Heidelberg (2007)
23. Zhao, J., Wang, H., Ren, H., Kee, S.C.: Lbp discriminant analysis for face verification. In: CVPR, vol. 3, p.167 (2005)

Efficient OCR Post-Processing Combining Language, Hypothesis and Error Models[*]

Rafael Llobet, J. Ramon Navarro-Cerdan,
Juan-Carlos Perez-Cortes, and Joaquim Arlandis

Instituto Tecnologico de Informatica
Universidad Politecnica de Valencia
Camino de Vera s/n, 46071 Valencia, Spain
{rllobet,jonacer,jcperez,arlandis}@iti.upv.es

Abstract. In this paper, an OCR post-processing method that combines a language model, OCR hypothesis information and an error model is proposed. The approach can be seen as a flexible and efficient way to perform Stochastic Error-Correcting Language Modeling. We use Weighted Finite-State Transducers (WFSTs) to represent the language model, the complete set of OCR hypotheses interpreted as a sequence of vectors of *a posteriori* class probabilities, and an error model with symbol substitutions, insertions and deletions. This approach combines the practical advantages of a de-coupled (OCR + post-processor) model with the error-recovery power of a integrated model.

1 Introduction

Any method of optical recognition of printed or handwritten text is subject to variable amounts of error and uncertainty in the output. The application of a correction algorithm is therefore very important. The excelent performance shown by humans when we read a handwritten text is mostly due to our extraordinary error-recovery ability, thanks to the lexical, syntactic, semantic, pragmatic and discursive language constraints we routinely apply.

The goal of an OCR post-processing method is to optimize the likelihood that the strings generated as OCR hypotheses are correct, in the sense that they are compatible with the constraints imposed by the task. These constraints conform the Language Model and can be as simple as a small set of valid words (e.g. the possible values of the "country" field in a form) or as complex as an unconstrained sentence in a natural language.

In practice, the simplest method to handle OCR output correction is to use a lexicon to validate the known words and ask the operator to verify or input manually the unknown words. Specific techniques can be used to carry out

[*] Work partially supported by the Spanish MICINN grants TIN2009-14205-C04-02 and Consolider Ingenio 2010: MIPRCV (CSD2007-00018) and by IMPIVA and the E.U. by means of the ERDF in the context of the R+D Program for Technological Institutes of IMPIVA network for 2010 (IMIDIC_2009/204).

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 728–737, 2010.

approximate search in the lexicon. In [10] an excellent survey of string search methods is presented.

Other methods are based on n-grams or on finite-state machines [9,12,3], where a candidate string is parsed and the set of transitions with the lowest cost (highest probability) defines the output string. The classical algorithm, widely used in different fields, to find the maximum likelihood path on a finite-state machine and to perform error-correcting parsing on a regular grammar is the Viterbi Algorithm [7,8].

All these approaches use a string provided by the OCR as input, apply a Language Model and often optimize a transformation cost using an Error Model, but, in general, they do not take into account another valuable knowledge source that we call the *Hypothesis Model*. Depending on the OCR classifier used, this can include the *a posteriori* class probabilities of the output hypothesis or another reliability index for the most likely classes. Another element to take into account is the classifier's confusion matrix., that should be efficiently and adequately included into the Error Model.

2 Weighted Finite-State Transducers

Weighted Finite-State Transducers (WFST) have been widely used in speech recognition, machine translation and pattern recognition, among other disciplines. In this paper we propose the use of WFSTs in Stochastic Error-Correcting Language Modeling for OCR post-processing.

A WFST can be seen as a generalization of a Finite-State Automata (FSA) [1,4]. An FSA can be seen as a finite directed graph with nodes representing states and arcs representing transitions. Each transition is labeled with a symbol from an alphabet Σ. Formally, an FSA is defined as a five-tuple (Q, q_0, F, Σ, δ) where Q is a finite set of states, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the subset of final states, Σ is a finite set of symbols and $\delta : Q \times \Sigma \rightarrow Q$ is the set of transitions between states. Each transition t is labeled with a symbol $s(t) \in \Sigma$. FSAs are used to *accept* or *reject* sets of strings over Σ: given a string $w \in \Sigma^*$, w is accepted if there is a path from the initial state to a final state of the graph whose transition labels form the string w when concatenated.

However, in Finite State Transducers (FSTs) each transition is labeled with an input symbol $\in \Sigma$ and an output symbol $\in \Delta$. Therefore, the function δ is defined as $\delta : Q \times \Sigma \rightarrow Q \times \Delta$. FSTs are used to *transduce* strings of an input language over Σ into strings of an output language over Δ. The weighted version of an FST (WFST) include a weight in their transitions, used to represent the cost of taking a particular path. Furthermore, each state q has an initial weight $w_i(q)$ and a final weight $w_f(q)$. A state q is *initial* if $w_i(q) \neq \bar{0}$ and *final* if $w_f(q) \neq \bar{0}$.

An FSA and its weighted counterpart WFSA can be seen as an FST or WFST respectively, with same input and output symbols in each transition. This is called the identity transducer.

The FSTs (and WFSTs) are considered specially flexible and powerful due to some fundamental properties. In particular, the approach presented in this paper relies on the *composition* operation [6]. Given two transducers T_1 and T_2, if T_1 transduces the string $x \in \Sigma$ to $y \in \Delta$ with weight w_1 and T_2 transduces $y \in \Delta$ to $z \in \Gamma$ with weight w_2, then their composition $T_3 = T_1 \odot T_2$ transduces x to z with weight $w_1 \otimes w_2$.

3 Description of the Method

The proposed approach entails building and composing WFSTs that encode different informations and represent distinct models, extending the idea of OCR language modeling through Stochastic Error Correcting Parsing proposed in [3].

We identify three sources of information in the OCR post-processing task: a) the OCR output (including all the hypotheses for each character and their class probabilities), b) a model of the expected errors and their probabilities, and c) the language the strings of the task belong to. Each of these sources of information can be represented by a Stochastic Finite-State Machine that we call the Hypothesis Model (HM), the Error Model (EM) and the Language Model (LM) respectively.

3.1 The Language Model (LM)

We propose the use of a grammatical inference algorithm to build a stochastic finite-state machine that accepts the smallest k-Testable Language in the Strict Sense (k-TS language) [5] consistent with a task-representative language sample. The set of strings accepted by such an automaton is equivalent to the language model obtained using n-grams, for $n = k$.

A major advantage of the chosen setting resides in its flexibility. The language sample can be a simple lexicon (with each word appearing only once), a list of words extracted from a real instance of the task (with each word appearing as many times as in the sample), a list of sentences with characters, words or word categories as the symbols of the grammar, etc. Only in the first case, when using a classical lexicon, the automaton is not required to be stochastic, since a lexicon is not a representative language sample. In the other cases, the model will take advantage of the probabilistic information present in the data.

The value of k can also be used to define the behavior of the model. In a lexical model, if k is made equal to the length of the longest word in the sample, a post-processing method is obtained where only words that exist in the sample are valid, but if k is set to a lower value, a classical n-gram model will result, where the corrected words may not be in the reference sample.

Figure 1 shows the probabilistic identity transducer associated with the sample S={aba, abb, ba, bac} and $k = 3$. In this description, for convenience, we have used a transducer with input and output symbols equal in each transition, i.e., the identity transducer, which can be seen as an acceptor of the language $L(S)$.

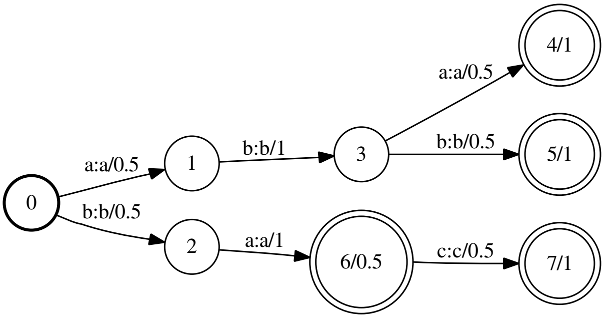


Fig. 1. Example of an identity transducer representing a language model

3.2 The Hypothesis Model (HM)

The output of a recognizer, in our case, an OCR classifier can be seen, in its most general form, as a sequence of n-dimensional vectors $\bar{v}_1 \ldots \bar{v_m}$, where n is the number of possible hypotheses for each character, m the length of the output string and $v_{i,j}$ the *a posteriori* probability of the j^{th} hypothesis of the i^{th} character. We propose to represent this sequence using a WFSA (or an identity WFST) with $m+1$ states and n transitions between each pair of states. Figure 2 shows an example of a WFST with alphabet $[a, b, c]$ that represents the OCR output $[0.8, 0.2, 0.0], [0.1, 0.7, 0.2], [0.0, 0.6, 0.4]$. This means that the first symbol of the OCR output is a with probability 0.8 or b with probability 0.2, the second symbol is a, b or c with probabilities 0.1, 0.7 and 0.2 respectively, and so on. Transitions with zero-probability are not shown in the graph.

Instead of working exclusively with the most probable output (*abb* in the example) this transducer models the uncertainty of the OCR classifier.

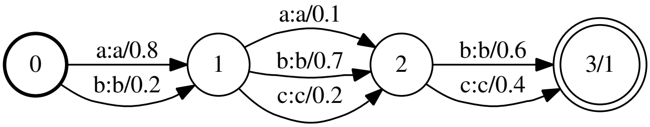


Fig. 2. Example of an identity transducer representing a hypothesis model. The *a posteriori* probabilities from the OCR classifier are shown as the arc weights.

3.3 The Error Model (EM)

In some cases, none of the character sequences included in the OCR hypothesis is compatible with the language model or a similar variant is more probable than any of the original alternatives. In a classical n-gram model, this effect is accounted for by a *smoothing* procedure. In our case, the possible variations allowed and their probabilities are represented by an *Error Model*.

Typically, the three usual edit operations will be defined: *substitutions* (including the substitution of a symbol by itself), *insertions* and *deletions*. Given two

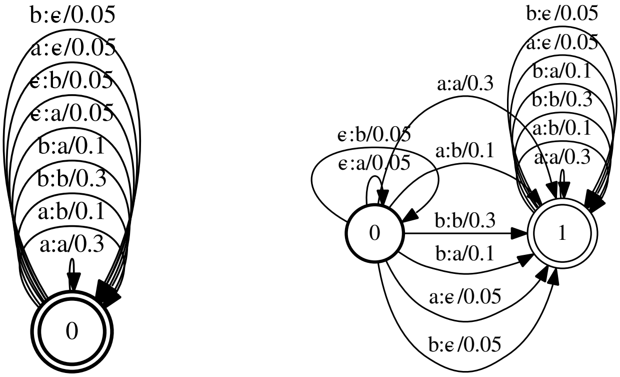


Fig. 3. Examples of two error model transducers, with all possible insertions, deletions and substitutions (left) and with insertions only at the begining of the string (right)

symbols s_1, s_2 and the empty symbol ϵ, substitutions, insertions and deletions are transductions of type s_1/s_2, ϵ/s_2 and s_1/ϵ respectively.

Each of these operations can be assigned a probability. The probability of substitutions is derived from the confusion matrix of the OCR classifier. This matrix is a table containing the confusion probability of each pair of symbols estimated using a representative corpus. It can be interpreted as a "static" model of the uncertainty of the OCR classifier, complementing the "dynamic" estimation provided by the *a posteriori* probabilities. The likelihoods of insertions and deletions are task-dependent and can be empirically estimated. Figure 3 shows an example of a WFST representing an error model with symbols in {a,b}.

3.4 Composing LM, EM and HM

The combination of the different models is performed through the composition operation between transducers:

Let L_1 be the set of strings that a given HM can produce, and L_2 the set of strings accepted by a given LM. Our goal is to find the most likely transduction of a string in L_1 into a string in L_2 by means of the intermediate transduction defined in an EM. This process is equivalent to finding the most probable path in the transducer HM $\odot$ EM $\odot$ LM.

The transducer $T_1 = \text{HM} \odot \text{EM}$ transduces any string from L_1 by applying the operations of the error model EM. Figure 4 shows the composition of the transducers HM and EM shown in Figures 2 and 3 respectively. This automaton transduces the strings accepted by HM to any string in Σ^*.

Therefore, the transducer $T_2 = T_1 \odot \text{LM}$ accepts only strings belonging to L_2, and the result of the transduction with the most probable path is the final corrected string. If several alternatives are needed, the n-best paths can also easily be obtained.

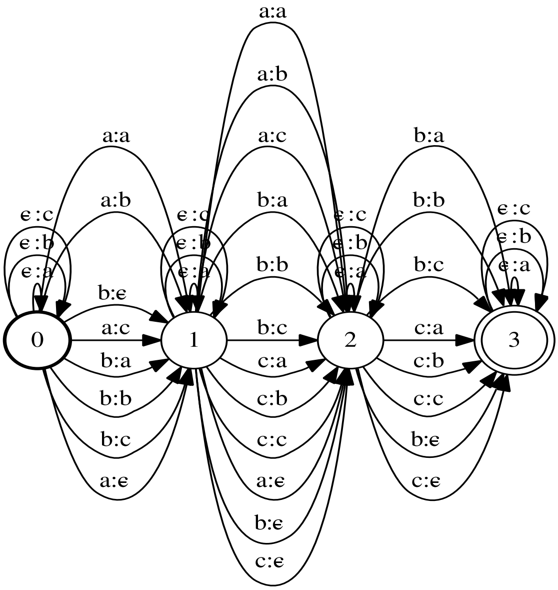


Fig. 4. Composition of the HM shown in Fig. 2 with an EM with all possible substitutions, insertions and deletions

3.5 Cost Definition and Parameter Optimization

The computation of the best path is obviously a key element in the process. A path is a sequence of transitions in the composed transducer and each transition t has an associated probability, which is computed as the product of the probabilities of the corresponding transitions in HM, LM and EM. Assuming independence and an equal influence from all models, we can define the probablity of a transition as:

$$P(t) = P(\mathrm{LM}, \mathrm{EM}, \mathrm{HM}|t) = P(\mathrm{LM}|t)P(\mathrm{EM}|t)P(\mathrm{HM}|t)$$

The probability of the output string can therefore be computed as the product of the probabilities of the transitions along the most probable path in the composed transducer. Given $x \in L_1$ and $y \in L_2$, the probability of the transduction x, y is $P(x, y) = \prod_{i=1}^{n} P(t_i)$, where $t_1 \ldots t_n$ is the sequence of transitions that transduces x into y.

To avoid underflow problems, instead of working with probabilities we have used tropical semiring WFSTs $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ where $\mathbb{K}$ are negative log probabilities, $\oplus$ is the `min` operation, $\otimes$ is $+$, $\bar{0}$ is $+\infty$ and $\bar{1}$ is 0. Therefore, the most probable path will be found using a lowest cost path search.

Since the optimum influence of each model is generally not known, two parameters λ_e and λ_h are defined to obtain a log-linear parametric combination of the models with different weights:

$$P(t) = P(\mathrm{LM}|t)P(\mathrm{EM}|t)^{\lambda_e}P(\mathrm{HM}|t)^{\lambda_h}$$

We consider a fixed weight 1 for the LM, therefore its influence is controlled by the absolute values of the other parameters. The values of λ_e and λ_h, along with the cost of insertions and deletions, mentioned in Section 3.3, can be empirically estimated using a supervised training set.

In a typical form-processing task in the data entry industry, it is very important to obtain a consistent confidence value (in our case, the probability associated to the shortest path in the combined transducer) allowing the user to define a threshold and a reliable reject strategy. Consequently, we have optimized the aforementioned parameters using a criterion function that maximizes the recognition rate, defined as the percentage (with respect to the total test set) of strings that were accepted and successfully corrected, for a given error rate (percentage, also in the total test set, of the strings that were accepted and generated wrong corrections). With this strategy, only rejected strings have to be reviewed by human operators, meaning that –for a commercially acceptable error rate– the economic savings yielded by the system are roughly equivalent to the number of accepted strings.

3.6 Pruning

WFST composition of very large transducers can incur in large computational costs. For a LM of 64000 states and 140000 transitions (like the one used in our experiments), a standard EM with all possible insertions, deletions and substitutions and an average-sized HM with 8 states and 5 transitions (hypotheses) per state, the resulting composed transducer can have up to 450000 states and more than two million transitions.

To avoid this problem, *lazy* composition together with a pruning scheme have been used. Lazy operations delay the computation of the result until it is required by another operation. This is useful when a large intermediate transducer must be constructed but only a small part of it needs to be visited [1]. In our approach, the composition is delayed until the search of the shortest path (the lowest cost path) in the resulting transducer is performed. In principle, it is necessary to completely compose the transducers to compute the shortest path, but we have used a simple pruning search optimization to provide an approximate solution that allows not to explore (and therefore compose) the whole transducer.

To deal with the shortest path search, a best-first algorithm which explores the automaton by expanding the lowest cost path at each state is used. A vector with the minimum cost found at each stage (path length) is maintained. During the search, a pruning is performed based on a parameter δ. If the cost of a partial solution of length n exceeds δ times the cost of the best path of length n found so far ($v[n]$), then the path of the partial solution is pruned.

Obviously, this heuristic leads to an approximate search, since the lowest cost path could be pruned. This can happen when δ is too low or when the best path contains high-cost transitions in its first stages. To avoid pruning a partial solution that could lead to the best path too early, a parameter ρ is used, so that the pruning scheme is not applied to partial solutions shorter than ρ states.

4 Experiments

The following experiments compare the system working with and without multiple hypotheses and *a posteriori* probabilities in HM. We used a sample of 14000 handwritten surnames from forms scanned in a real industrial task, with a reference language model of 4.5 million Spanish surnames (99157 of which were unique). A k equal to the largest surname was used in the LM, so only known surnames were accepted as corrected output. The OpenFST library was used for the experiments [1,2].

The corpus was divided into a training (15%) and a test (85%). The training set was used to estimate the parameters of the error model (insertion and deletion probabilities) and of the WFSTs composition (λ_h and λ_e) using the criterion function defined in Section 3.5. Since the influence of each individual model can vary depending on the selected approach –using multiple hypotheses and *a posteriori* probabilities (WFST-PP) or using only the most probable OCR output (WFST)– independent optimizations were performed for each approach.

Table 1 shows the best parameters found for WFST and WFST-PP. It can be noted that the optimal working point in the WFST approach is achieved when all the models have similar weights (note that LM has a fixed weight of 1), whereas the WFST-PP approach achieves better performance when the HM has a higher weight than the other two models. Also the insertion and deletion probabilities are lower in the WFST-PP approach, since more strings can be corrected with a lower cost by choosing one of the symbols proposed by the HM rather than by deletion and insertion operations.

Table 1. Optimal parameters found with and without *a posteriori* probabilities

	λ_e	λ_h	p_i	p_d
WFST-PP	1.17	2.38	0.005	0.004
WFST	1.05	1.04	0.007	0.008

Figure 5 shows the recognition and error rates of the proposed method using *a)* multiple hypotheses and *a posteriori* probabilities in HM (WFST-PP), *b)* the same approach using only the OCR strings (WFST), and *c)* the original, uncorrected OCR output.

The computational cost is another important issue in this task, where the size of the models can be very large in practice, and the typical operations involve large batchs of documents to recognize. A set of experiments were carried out to test the influence of the pruning method presented in Section 3.6. Figure 6 shows the average correction time (ms.) obtained in an Intel Xeon 2.5 GHz with 2 GB of memory, Linux OS and gcc 4.4, and the accuracy (percentage of well corrected words) achieved for different values of δ and ρ. These results were obtained for a

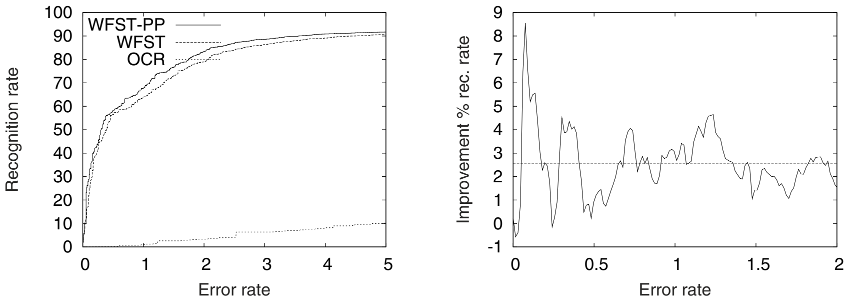


Fig. 5. Recognition rate against error rate comparison for different approaches in the range of 0 to 5% error rate and detail of the improvement obtained using an Hypothesis Model with posterior probabilities, in the range of 0 to 2% error rate

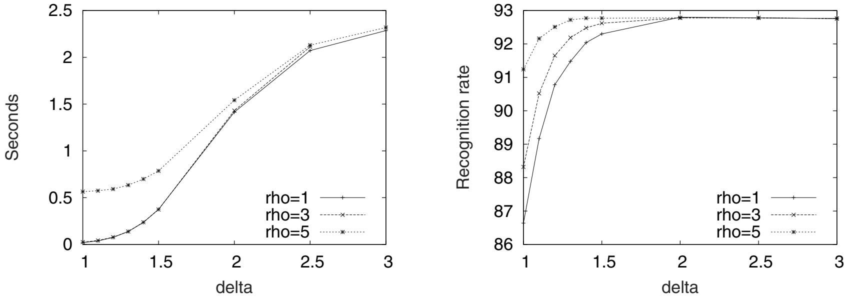


Fig. 6. Average correction time (left) and percentage of well corrected words at zero rejection rate (right) for different values of δ and ρ

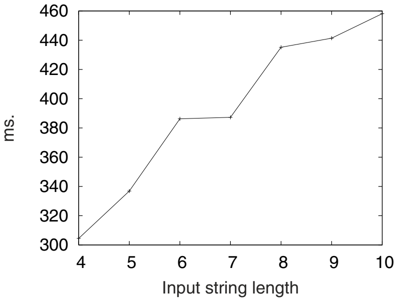


Fig. 7. Average correction time (ms.) against input string length for $\delta = 1.5$ and $\rho = 3$

language model built from 99157 unique words. For larger language models, the computational cost grows sub-linearly. Figure 7 plots the average computational cost for $\delta = 1.5$ and $\rho = 3$, against the length of the input OCR hypothesis.

5 Conclusions

A post-processing method for OCR using WFSTs to encode the set of classifier hypotheses, an error model and a language model implementing a k-Testable Language has been proposed. The lowest cost path in the composed transducer gives the most probable string compatible with the language, the hypothesis and the error models. According to the tests conducted with handwritten data, significant improvements over previous approaches can be obtained efficiently.

Finally, in our view, using independent error, language and OCR models that can be modified without affecting the other parts of the system offers important practical advantages over other more closely coupled paradigms.

References

1. Mohri, M., Pereira, F., Riley, M.: The design principles of a weighted finite-state transducer library. Theoretical Computer Science 231, 17–32 (2000)
2. Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., Mohri, M.: OpenFst: A General and Efficient Weighted Finite-State Transducer LIbrary. In: Holub, J., Žd̆árek, J. (eds.) CIAA 2007. LNCS, vol. 4783, pp. 11–23. Springer, Heidelberg (2007)
3. Perez-Cortes, J.C., Amengual, J.C., Arlandis, J., Llobet, R.: Stochastic Error Correcting Parsing for OCR Post-processing. In: Proceedings of the ICPR, vol. 4, pp. 405–408 (2000)
4. Vidal, E., Thollard, F., de la Higuera, C., Casacuberta, F., Carrasco, R.: Probabilistic Finite-State Machines - Parts I and II. IEEE Trans. on Pattern Analysis and Machine Intelligence 27, 1013–1039 (2005)
5. Garcia, P., Vidal, E.: Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. IEEE Trans. on PAMI 12, 920–925 (1990)
6. Riley, M., Pereira, F., Mohri, M.: Transducer composition for context-dependent network expansion. In: Proc. of Eurospeech 1997 (1997)
7. Amengual, J., Vidal, E.: Efficient error-correcting viterbi parsing. IEEE Trans. on PAMI 20, 1109–1116 (1998)
8. Neuhoff, D.: The viterbi algorithm as an aid in text recognition. IEEE Trns. on Inf. Theory 21, 222–226 (1975)
9. Berghel, H.L.: A logical framework for the correction of spelling errors in electronic documents. Information Processing and Management 23, 477–494 (1987)
10. Hall, P., Dowling, G.: Approximate string matching. ACM Surveys 12, 381–402 (1980)
11. Beaufort, R., Mancas-Thillou, C.: A Weighted Finite-State Framework for Correcting Errors in Natural Scene OCR. In: Proceedings of the Ninth International Conference on Document Analysis and Recognition, vol. 2, pp. 889–893 (2007)
12. Farooq, F., Jose, D., Govindaraju, V.: Phrase-based correction model for improving handwriting recognition accuracies. Pattern Recognition 42, 3271–3277 (2009)

Rejection Threshold Estimation for an Unknown Language Model in an OCR Task*

Joaquim Arlandis, Juan-Carlos Perez-Cortes,
J. Ramon Navarro-Cerdan, and Rafael Llobet

Instituto Tecnológico de Informática
Universitat Politècnica de València
Camí de Vera s/n, 46071 València, Spain
{arlandis,jcperez,jonacer,rllobet}@iti.upv.es

Abstract. In an OCR post-processing task, a language model is used to find the best transformation of the OCR hypothesis into a string compatible with the language. The cost of this transformation is used as a confidence value to reject the strings that are less likely to be correct, and the error rate of the accepted strings should be strictly controlled by the user. In this work, the expected error rate distribution of an unknown language model is estimated from a training set composed of known language models. This means that after building a new language model, the user should be able to automatically "fix" the expected error rate at an acceptable level instead of having to deal with an arbitrary threshold.

Keywords: Error rate, rejection threshold, language model, error-correcting parsing, OCR post-processing, regression model.

1 Introduction

Optical recognition of printed or handwritten text is often followed by a post-processing phase that can significantly improve the final performance if some constraints are imposed on the contents of the text. The set of constraints can be formally represented as a *language model* (be it a natural language or a subset of a natural language, a closed list of words or expressions, a code following some pattern, etc.). Forms with fields that are filled-in by hand are typical documents where different models can be defined for each field. Frequent field types are "Name", "Age", "Date", "Country", "Street", "Symptoms", "Incident description", "Id. Number", "Phone Number", etc. The language models associated with each of these fields are widely different in many regards (alphabet, size, complexity, perplexity...) and, unlike the OCR classifier for example, that is often kept unchanged for a reasonable amount of time, new language models appear routinely in the normal form-processing large-scale industrial activity.

* Work partially supported by the Spanish MICINN grants TIN2009-14205-C04-02 and Consolider Ingenio 2010: MIPRCV (CSD2007-00018) and by IMPIVA and the E.U. by means of the ERDF in the context of the R+D Program for Technological Institutes of IMPIVA network for 2010 (IMIDIC-2009/204).

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 738–747, 2010.

Very different techniques have been employed to post-process the OCR hypotheses according to a required model (see section 2) and most of them provide or can be easily modified to provide a reliability index (directly related to the *correction confidence* and inversely related to the *transformation cost*).

Applying a threshold to these costs or confidence values allows the system to reject those strings that are less likely to be correct (those involving a high cost or "effort" to convert the OCR hypothesis to a correct output). Usually, the rejected sequences are submitted to a manual data-entry process and therefore the threshold selection has a high impact in the practical performance and economic benefit of the system. The maximum acceptable error rate in the accepted strings (which could be regarded as *false positives*) depends on the particular task at hand, and the number of rejections must be minimized due to the cost, in terms of time and money, of the human data-entry process.

In this paper, a technique to estimate the expected error rate distribution of a new, unknown, language model, is proposed. That distribution is used to estimate the rejection threshold of a test sample in order to obtain a given expected error rate. Experiments are presented comparing the accuracy of the estimations in different conditions.

The rest of the paper is organized as follows: section 2 contains an overview of the related work. Section 3 describes how the error rate distribution as a function of the transformation costs can be learned, predicted, and used to compute the rejection threshold. In section 4, experiments and results on error rate estimation for different languages are reported, and, finally, the conclusions are presented in section 5.

2 Related Work

Many works on language modeling have been carried out in the field of continuous speech recognition [10]. Although the requirements are very different, many basic techniques used in that discipline can be applied to OCR tasks with little modification. Word and sentence level models typically apply dictionary search methods, n-grams, Hidden Markov Models, Edit Distance-based techniques, and other character or word category transition models. In [6], an excellent survey of approximate string search methods is presented. There are several works of using language modeling techniques for error correcting applied to OCR and text recognition tasks, either on constrained or unconstrained environments. Some examples can be found in [9], [18], [15], [12].

In this work, the error-correcting parsing (ECP) technique has been used to post-process the OCR hypotheses is as described in [15]. It consists of building a finite-state machine from a formal grammar, that accepts (or generates with a certain probability) the strings in the lexicon or language sample. When the model is applied to a candidate word the smallest set of transitions that could not be traversed shows which is the most similar string in the model, and the minimal cost of the selected path is provided as a transformation cost of the input. The classical algorithm, widely used, to find the maximum likelihood path on a Markov model, and to perform ECP, on a regular grammar, is the Viterbi Algorithm, based on the Dynamic Programming paradigm. The extension of the Viterbi algorithm used in this work is described in [1].

The construction of the finite-state machine has been performed using a grammatical inference algorithm that accepts the smallest k-Testable Language in the Strict Sense (k-TS language) [19] consistent with a task-representative language sample. The set of strings accepted by such an automaton is equivalent to the language model obtained using n-grams, for n=k. The stochastic extension of the basic k-TS language is performed through a maximum likelihood estimation of the probabilities associated to the grammar rules, evaluated according to their frequency of utilization by the input strings. This computation is carried out incrementally and simultaneously with the inference process.

Given the impact of the quality of the confidence estimation on the practical use of an OCR system, many recent works exist that deal with this problem. The work of Landgrebe [13] proposes a modified version of the ROC curve, where a factor to tune the number of expected false positives is introduced in order to tackle with imprecise environments. Other works directly related to post-processing in OCR and text recognition tasks, propose rejection strategies oriented to yield reliable confidence measures [3], [5], [16]. The use of confidence measures has also been specially and traditionally studied in the Speech Recognition and Natural Language Processing areas.

The particular problem of automatic rejection threshold estimation has also applications in economics, medicine, network management, signal processing, and others. A statistical approach often used in many different areas is based on the conventional Monte Carlo techniques, where the thresholds are set according to the distribution percentiles of the measures (or cost functions). These approaches demand very large number of samples to be useful.

Also, statistical methods have been developed, like in [7], where threshold estimation is studied in the context of regression. In sensor systems, where large amounts of data are usually available, the target detection is seriously affected by false positives, and a special effort has been made to improve their behavior. Thus, Ozturk *et al.* [14] used the generalized Pareto distribution to approximate the extreme tail of the distributions of radar measures, and propose the ordered sample least squares method for estimating the parameters of the distributions. Recently, Broadwater and Chellappa [4] proposed an algorithm using extreme value theory through the use of the generalized Pareto distribution, too, and a Kolmogorov-Smirnov statistical test, and propose a way to adaptively maintain low false positive rates and to overcome differences between the model assumptions and the real data.

In other Pattern Recognition tasks, the problem of rejection threshold estimation has also been studied. For instance, in [2], several methods for estimating speaker-independent and speaker-dependent decision thresholds for automatic speaker verification were compared using only relevant parameters estimated from training data.

In handwritten numeral recognition, He *et al* [8] used Linear Discriminant Analysis to determine the rejection threshold by taking into account the confidence values of the classifier outputs and the relations between them. In text correction, Kae and Huang [11] used a technique for identifying a set of correct words by bounding the probability that any given word from an OCR output is incorrect using an approximate worst case analysis.

In the context of many real tasks, specifically estimating an automatic rejection threshold from an user-defined expected error rate would alleviate the problem of dealing with arbitrary (in practice) confidence measures. In this sense, a closer goal to the one presented in this work has been proposed by Serrano *et al* in the context of error supervision in interactive-predictive handwriting recognition [17]. The objective was to assist the user in locating possible transcription errors: the user decides on a maximum tolerance threshold for the recognition error (after supervision), and the system adjusts the required supervision effort on the basis of an estimate for this error.

3 Approach

If we take a representative sample of strings consisting of OCR hypotheses, and compute the transformation costs using a post-processing algorithm (in our

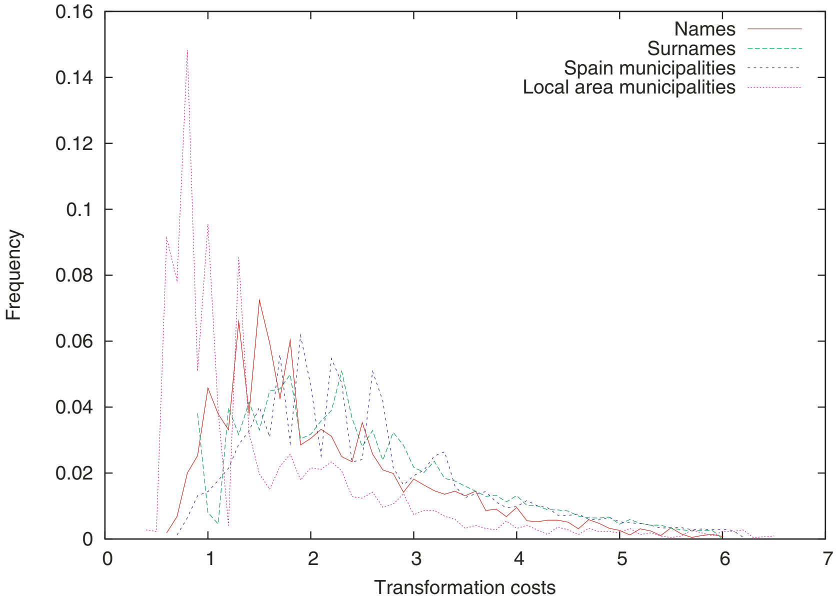


Fig. 1. Histogram of the correction costs of OCR hypotheses strings from four different language models (Spanish names, Spanish surnames, all Spanish towns, and towns from a local region: Comarca de “La Ribera Alta”)

case, ECP on a k-TS language [15]), the distribution obtained varies widely for different language models, as can be seen in Figure 1. This means that choosing a consistent rejection threshold is nothing but trivial, since the number of accepted and rejected strings for a given threshold will be very different depending on the characteristics of the language model. Also, moving the threshold value slightly can lead to unpredictable changes on the ratio of accepted/rejected strings.

Therefore, a more predictable confidence index is needed. A technique to estimate the error rate distribution of a test sample as a function of the transformation costs is proposed, consisting on the following steps:

- Given a set of transformation costs obtained from a representative sample of manually labeled OCR hypotheses strings from a language, the error rates associated to each cost *(error rate distribution)* are learned, and then used to find the rejection threshold for new samples of the same language. As described in the next section, the error rate distribution can be used to estimate the rejection threshold for a given expected error rate.
- When a new language model is defined in the system, an automatic way to estimate its error rate distribution that uses exclusively characteristics measured directly on the language model by means of regression techniques is proposed. This way, the time-consuming process of acquisition, OCR and manual validation of a significant amount of strings is avoided. This is specially important if new language models are needed frequently, even if they are subsets or special variants of known models. In section 3.2, the details of the approach are explained.

3.1 Modeling the Error Rate Distribution of a Language Model

Given a language model and a set of transformation costs obtained using a post-processing algorithm with a representative sample of OCR hypothesis strings (for which the ground-truth transcriptions have been manually obtained), a smoothed histogram $H_E(c)$ of error rates for different costs c can be computed using the expression,

$$H_E(c, w) = \frac{|S^-_{c,w}|}{|S_{c,w}|} \tag{1}$$

where w is a smoothing window size parameter, $|S^-_c|$ is the number of strings "erroneously corrected" into an incorrect string having a cost between $c-w$ and $c+w$, and $|S_c|$ is the total number of strings having a cost also in that interval. The window size can also be defined dynamically to enclose a given number of costs around c instead of a fixed cost interval. In Figure 2, a histogram H_E obtained using the post-processing algorithm of [15] on different language models is shown.

We can easily find the rejection threshold $\mathcal{T}_c$ required to obtain a given error rate ϵ on a test sample S' by accumulating averaged values of H_E according to increasing values of c,

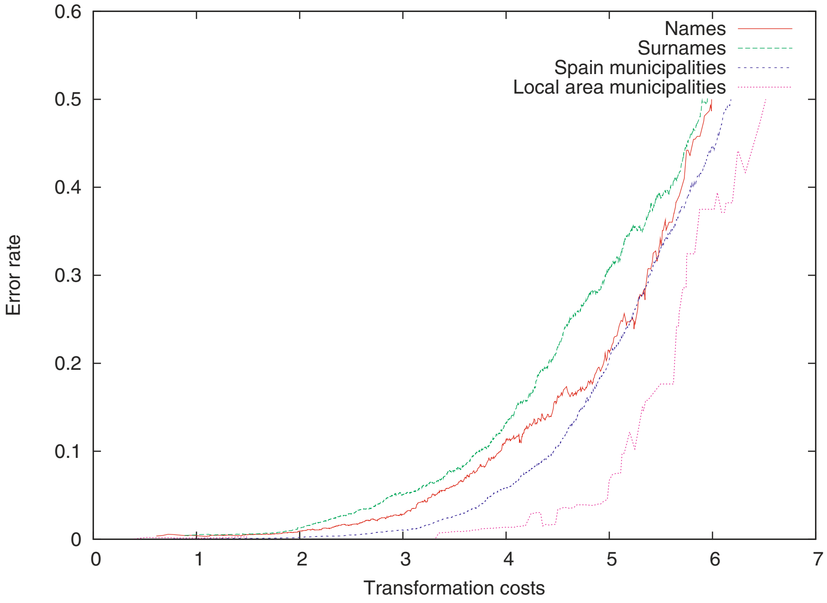


Fig. 2. Error rate histogram H_E, for the sample of language models plotted in Figure 1 using $w = 0.5$ (Equation 1)

$$E(i) = \sum_{c=c_1}^{c_i} \frac{H_e(c,w)}{i} \ , \ c \in S'$$

where the value of $E(i)$ at each point is the average error rate of the strings with costs smaller or equal than c_i. Then, the $\mathcal{T}_c$ value we seek is the largest one where the curve reaches ϵ (since the curve can decrease at some points, we should choose the last value of c to maximize the number of accepted strings for a given ϵ).

E can be seen as a cumulative averaged version of H_E for a given test sample and it can be used to approximate the appropriate cost threshold to use when we want to fix the expected error rate. In practice, different test samples will require different rejection thresholds for a given user-defined error rate.

3.2 Estimating the Error Rate Distribution of New Language Models

Let H_C be the histogram of the transformation costs of the strings that belong to a language (positive sample). H_C can be easily obtained from the list of positive strings because it does not depend on the OCR process. Figure 3 shows the histogram H_C of the same four language models shown in Figure 1.

Both figures 1 and 3 suggest that there is a correlation between the distributions of the costs of OCR hypothesis strings (many of them having errors), and

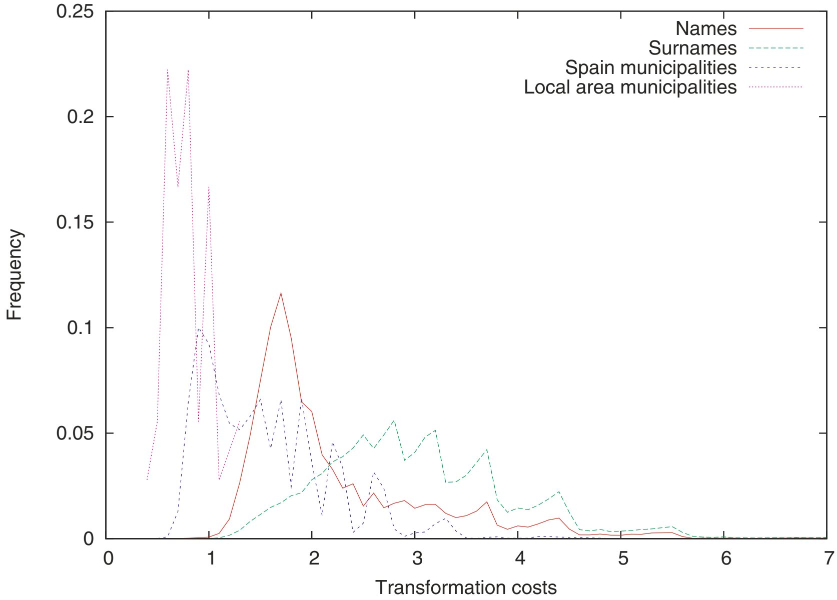


Fig. 3. Histogram H_C of the correction costs of strings belonging to four different language models (Spanish names, Spanish surnames, all Spanish towns, and towns from a smaller local region: Comarca de "La Ribera Alta")

positive samples of the same language model (without errors). And, as already mentioned, the cost distributions of different languages clearly differ.

Assuming the above statement, we propose that a training set composed by features extracted from the histograms H_C and H_E of a set of known languages models is used to build a regression model able to predict the expected error rate distribution $\widehat{H_E}$ (target output) of a new language based on features extracted from its H_C histogram (inputs).

Several regression methods have been tested. The results obtained and the details on these methods and their parametrization are described in the next section.

4 Experiments

The goal of the experiments has been to measure the capability of the regression techniques to learn a function that approximates the error rate distribution $\widehat{H_E}$ of new language models from a model built using features extracted from known language models as described in the former section.

The four different language models shown in the figures of the previous sections have been used to perform a leaving-one-out estimation. They are the names and surnames in the last census of Spain: 66363 names and 97157 surnames with probabilities derived from their frequencies in the census, all Spanish municipalities (8201 towns without frequencies, and 35 municipalities, without frequencies,

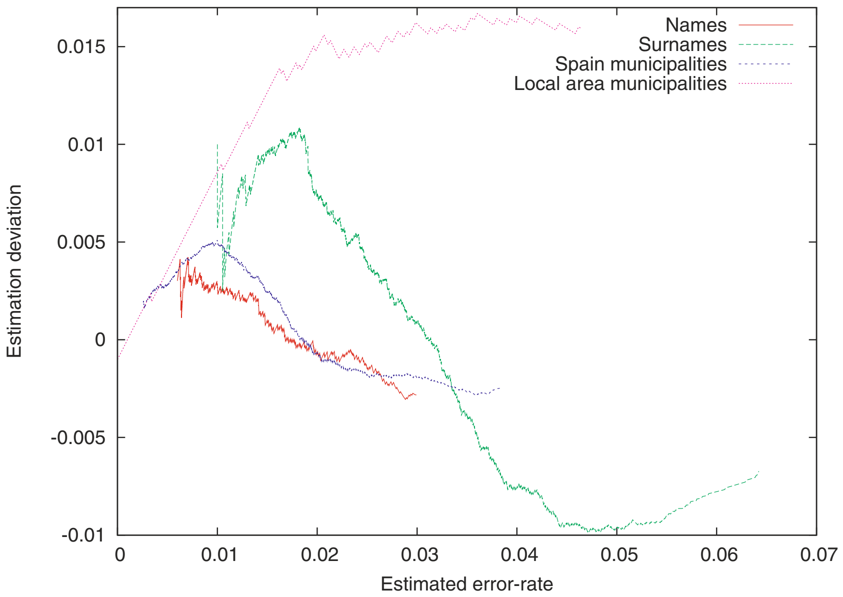


Fig. 4. Differences between the estimated error rate and the real error rates for the four language models in a leaving-one-out experiment

from a local region: Comarca de “La Ribera Alta”). These languages have been chosen since they are representative of real tasks and span a wide range of sizes.

For each experiment, a single regression model has been built using 2000 OCR hypothesis strings chosen randomly from each language model. To train the model, a number of features of each language including transformation cost and error rate to describe the distributions of H_E, and statistics like mean, median variance, percentiles, coefficient of variation and frequencies of the bins describing the distribution of H_C have been combined. The target output variable for the regression is the error rate, measured applying the language model, for each cost.

Several regression models have been tested (Support Vector Machines for Regression, Radial Basis Functions and a Multilayer Perceptron, with similar results). The results are provided in terms of estimation deviation, i.e., the difference between the estimated error –computed as explained in section 3.1, but on the error rate distribution $\widehat{H_E}$ estimated by the regression model– and the real error measured in the test set.

In Figure 4, the estimation deviation is plotted against the estimated error, for the four language models. For the test of each language model, the regression model has been built using the other three language models.

We can see that the estimation can be useful in all cases, but it is more accurate in the case of Names and Spain Municipalities. In practice, the typically acceptable error rates are in the range of 1% or 2% (between 0.01 and 0.02 in

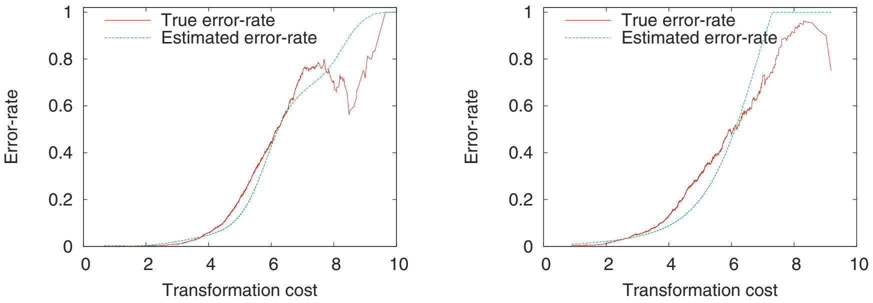


Fig. 5. Error rate histograms, H_E, and estimated error rate histograms $\widehat{H_E}$, for the language models of Spain municipalities (left) and Spanish surnames (right)

the figures). In that useful range, the error deviations are small enough to be directly usable in the two best languages, and a good starting point for a slight empirical adjustment in the case of the two worst languages. With a larger set of language models to train the regression model, we think these results can be significantly improved.

In Figure 5, the error rate histogram of the test sample, H_E, along with the estimated error rate histogram, $\widehat{H_E}$, are plotted for two of the language models studied.

5 Conclusions

We have presented a method for the estimation of the expected error rate distribution of an unknown language model, so that a user can establish the error rate at an acceptable level and the system estimates the rejection threshold automatically.

Experiments where a regression model is built using OCR hypotheses from a set of known languages have been performed, and the model is tested against a new language. The results show a useful behavior, with reasonably accurate estimations of the rejection threshold in the typically practical range of error rates. As a future work, we plan to train the regression model with a larger set of language models.

References

1. Amengual, J., Vidal, E.: Efficient error-correcting viterbi parsing. IEEE Trans. on Pattern Analysis and Machine Intelligence 20(10), 1109 (1998)
2. Lindberg, J., Koolwaaij, J., Hutter, H., Genoud, D., Pierrot, J., Blomberg, M., Bimbot, F.: Techniques for a priori decision threshold estimation in speaker verification. In: Proceedings RLA2C, pp. 89–92 (1998)
3. Bertolami, R., Zimmermann, M., Bunke, H.: Rejection strategies for offline handwritten text line recognition. Pattern Recognition Letters 27(16), 2005–2012 (2006)

4. Broadwater, J., Chellappa, R.: Adaptive threshold estimation via extreme value theory. IEEE Transactions on Signal Processing 58, 490–500 (2010)
5. Gandrabur, S., Foster, G.F., Lapalme, G.: Confidence estimation for nlp applications. TSLP 3(3), 1–29 (2006)
6. Hall, P., Dowling, G.: Approximate string matching. ACM Surveys 12(4), 381–402 (1980)
7. Hansen, B.E.: Sample splitting and threshold estimation. Econometrica 68(3), 575–604 (2000)
8. He, C.L., Lam, L., Suen, C.Y.: A novel rejection measurement in handwritten numeral recognition based on linear discriminant analysis. In: 10th Intl. Conf. on Document Analysis and Recognition, pp. 451–455. IEEE Computer Society, Los Alamitos (2009)
9. Hull, J., Srihari, S.: Experiments in text recognition with binary n-gram and viterbi algorithms. IEEE Trans. on Pattern Analysis and Machine Intelligence 4(5), 520–530 (1982)
10. Jelinek, F.: Up from trigrams, the strugle for improved language models. In: European Conf. on Speech Communication and Technology, Berlin, pp. 1037–1040 (1993)
11. Kae, A., Huang, G.B., Learned-Miller, E.G.: Bounding the probability of error for high precision recognition. CoRR, abs/0907.0418 (2009)
12. Kolak, O., Resnik, P.: Ocr post-processing for low density languages. In: Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT), pp. 867–874. Association for Computational Linguistics (2005)
13. Landgrebe, T., Paclík, P., Duin, R.P.W.: Precision-recall operating characteristic (p-roc) curves in imprecise environments. In: International Conference on Pattern Recognition ICPR (4), pp. 123–127 (2006)
14. Ozturk, A., Chakravarthi, P.R., Weiner, D.D.: On determining the radar threshold for non-gaussian processes from experimental data. IEEE Transactions on Information Theory 42(4), 1310–1316 (1996)
15. Perez-Cortes, J., Amengual, J., Arlandis, J., Llobet, R.: Stochastic error correcting parsing for ocr post-processing. In: International Conference on Pattern Recognition ICPR-2000, Barcelona, Spain, vol. 4, pp. 405–408 (2000)
16. Pitrelli, J.F., Subrahmonia, J., Perrone, M.P.: Confidence modeling for handwriting recognition: algorithms and applications. International Journal of Document Analysis 8(1), 35–46 (2006)
17. Serrano, N., Sanchis, A., Juan, A.: Balancing error and supervision effort in interactive-predictive handwriting recognition. In: International Conference on Intelligent User Interfaces (ICIUI), Hong-Kong, China (2010)
18. Tong, X., Evans, D.A.: A statistical approach to automatic ocr error correction in context. In: Fourth Workshop on Very Large Corpora, pp. 88–100 (1996)
19. Garcia, P., Vidal, E.: Inference of K-Testable Languages in the Strict Sense and Application to Syntactic Pattern Recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence 12(9), 920–925 (1990)

A New Editing Scheme Based on a Fast Two-String Median Computation Applied to OCR

José Ignacio Abreu Salas[1] and Juan Ramón Rico-Juan[2]

[1] Universidad de Matanzas, Cuba
jose.abreu@umcc.cu
[2] Dpto Lenguajes y Sistemas Informáticos, Universidad de Alicante, Spain
juanra@dlsi.ua.es

Abstract. This paper presents a new fast algorithm to compute an approximation to the median between two strings of characters representing a 2D shape and its application to a new classification scheme to decrease its error rate. The median string results from the application of certain edit operations from the minimum cost edit sequence to one of the original strings. The new dataset editing scheme relaxes the criterion to delete instances proposed by the Wilson Editing Procedure. In practice, not all instances misclassified by its near neighbors are pruned. Instead, an artificial instance is added to the dataset expecting to successfully classify the instance on the future. The new artificial instance is the median from the misclassified sample and its same-class nearest neighbor. The experiments over two widely used datasets of handwritten characters show this preprocessing scheme can reduce the classification error in about 78% of trials.

1 Introduction

Dataset editing has received considerable attention from the seminal works of Wilson [17] about the *edited near neighbor rule* (ENN) because this technique can be useful to improve *nearest neighbor classifiers* response. Mainly, these algorithms focus on deleting wrong tagged instances from a set, which will be used as training set for a given classifier. Several modifications have been proposed such as [2][4][13] and [15][18] more recently, facing with some problems of basic Wilson procedure as statistically dependence of estimations over each instance [10]. Another group of algorithms also changes some instances tag while editing, such [6] and [14].

In many problems, patterns do not have a vectorial representation, instead another syntactic coding such strings and trees are commonly used. Methods cited before mainly concern with these vectorial representations and distances. Therefore, when dealing with strings a suitable distance has be selected. In our case, the widely used Levenshtein edit distance [7] is choosen. In addition, some approaches find representatives instances as centroids [12] or prototypes, so this concepts need to be extended to the new coding schemes.

Several authors [1] [3] [5] have described algorithms to get a prototype representing a set of strings, as the centroid, the median string or an approximation. Most of this works build the desired string by successive refinements of a initial string or by some *ad-hoc* procedure.

E.R. Hancock et al. (Eds.): SSPR & SPR 2010, LNCS 6218, pp. 748–756, 2010.

This work proposes new Wilson based approach to edit a dataset of instances that have been encoded by some string representation. The inclusion of a prototype representing both, an instance and its same-class nearest neighbor inside a k-neighborhood if exist, when the instance is misclassified, is the main difference from others described in the literature. Besides, this paper presents a fast algorithm to compute the prototype representing two strings suitable for requirements of described editing procedure. Section 2 provides a detailed explanation of the algorithm to build the prototype and some useful concepts. Section 3 describes the proposed editing procedure and several considerations related to the computational complexity of proposed algorithms. Finally, section 4 illustrates with different experiments the behavior of proposed methods.

2 Prototype Construction

To compute the prototype representing two strings, this case defined as the median string, the proposed approach focus on information gathered from calculation of the distance between those strings. This section contains a glance of the selected distance measure, the Levenshtein [7] edit distance. Latter a procedure to compute the median string is covered.

2.1 Edit Distance

Let Σ be an alphabet and $S_1 = \{S_{11}, S_{12}..S_{1m}\}$, $S_2 = \{S_{21}, S_{22}..S_{2n}\}$ two strings over Σ where $m, n \geq 0$, the edit distance between S_1 and S_2, $D(S_1, S_2)$, is defined in terms of elementary edit operations which are required to transform S_1 into S_2. Usually three edit operations are considered:

- *substitution* of a symbol $a \in S_1$ by a symbol $b \in S_2$, denoted as $w(a, b)$
- *insertion* of a symbol $b \in S_2$ in S_1, denoted as $w(\varepsilon, b)$
- *deletion* of a symbol $a \in S_1$, denoted as $w(a, \varepsilon)$.

where ε denotes an empty string. Let $Q_{Si}^{Sj} = \{q_1, q_2, ..., q_k\}$ be a sequence of edit operations transforming S_i into S_j, if each operation has cost $e(q_i)$ the cost of Q_{S2}^{S1} is $E_{Q_{S2}^{S1}} = \sum_{i=1}^{k} e(q_i)$ and the edit distance $D(S_1, S_2)$ is defined as:

$$D(S_1, S_2) = argmin\{\ E_{Q_{S2}^{S1}}\}\ .$$

Strings involved in this work are Freeman Chaincodes, for that reason substitution costs are computed as follows:

$$e(w(a, b)) = argmin\{|a - b|, 8 - |a - b|\}$$

In the case of the insertions and deletions the cost 2 was chosen which is half of the maximum cost of the substitution operation; this same fixed number is used in [11]. The dynamic programming algorithm exposed by Wagner [16] allows to compute $D(S_1, S_2)$ in $\mathcal{O}(L_{S1} \times L_{S2})$ time, where L_S denotes the length of string S.

2.2 Fast Median String Computation

The median of a set T of strings can be briefly defined as the string R which realizes:

$$argmin_R\{\sum D(R,S_i)|S_i \in T\} \tag{1}$$

As explained above, the proposed approach computes the median R of two strings S_1 and S_2 by applying a subset of edit operations from the minimum cost edit sequence Q_{S2}^{S1} to S_1. To choose those editions that will be applied, each element at Q_{S2}^{S1} is tested to estimate how it will affect $D(S_1,R)$ and $D(S_2,R)$, since the algorithm seeks for a string R satisfying (1) and (2). This additional requirement means that an R near of the halfway between S_1 and S_2 will be preferred.

$$argmin_R\{|D(R,S_1)-D(R,S_2)|\} \tag{2}$$

The idea behind the algorithm is that each operation q_i in Q_{S2}^{S1} affects the future $D(R,S_1)$ and $D(R,S_2)$ since can be guessed that a rejected operation keeps R similar to S_1 while accepted editions makes R resembles to S_2. A close examination of each possible operation help to explain this conjecture.

For insertions, let b_{S2}^k be the k-esime symbol from S_2. An operation $q_i = w(\varepsilon, b_{S2}^k)$ from $Q_{S_1}^{S2}$ indicates the insertion of this symbol into S_1 to obtain R. Suppose q_i is accepted, thus can be expected that Q_R^{S2} does not involves an insertion of a symbol in R to be matched with b_{S2}^k in S_2 since it was done before. A similar reasoning led to guess this symbol is market to deletion from R when $D(R,S_1)$ is computed. In turn, a symbol $b_{S_1}^k$ deleted from S_1 to get R will be pointed to be inserted again while computing distance from R to S_1.

Substitutions $w(b_{S_1}, b_{S_2})$ will always applied, but whenever possible a symbol m will be placed in R instead b_{S_1} or b_{S_2}. The choice of m tries to make R similar to both S_1 and S_2, thus must satisfies:

$$e(w(b_{S_1},b_{S_2})) = e(w(b_{S_1},m)) + e(w(m,b_{S_2})). \tag{3}$$

$$argmin_m\{|e(w(m,b_{S_1})) - e(w(m,b_{S_2}))|\}. \tag{4}$$

Previous assumptions allow estimating how applying or not an operation q_i will affect distances from R to S_1 and S_2. Chosen insertions of b_{S2} into S_1 contributes with $e(w(b_{S2},\varepsilon))$ to $D(R,S_1)$ since the inserted symbol need to be deleted, now q_i has not an effect on $D(R,S_2)$. If the insertion is rejected implies $D(R,S_1)$ does not change, but $D(R,S_2)$ will grow by $e(w(\varepsilon,b_{S2}))$. Unlike the insertion occurs in the case of the deletion operation. If a deletion is discarded $D(R,S_2)$ increases by $e(w(b_{S1},\varepsilon))$ or $D(R,S_1)$ by $e(w(\varepsilon,b_{S1})$, if operation is accepted. Substitutions make both $D(R,S_1)$ and $D(R,S_2)$ grow by $e(w(b_{S1},m))$ and $e(w(m,b_{S2}))$ respectively.

For example, let $S_1 = \{a,b\}$, $S_2 = \{d,e\}$ and $e(w(\cdot,\varepsilon)) = e(w(\varepsilon,\cdot)) = 1$. Table 1 shows the substitution cost between symbols, thus $Q_{S1}^{S2} = \{w(a,\varepsilon), w(b,d), w(\varepsilon,e)\}$. Possible options to select or not an operation yields the tree at Figure 1 where each leaf node shows a candidate R. Inside brackets, an estimation of the cumulative contribution of each operation up from the node to $D(R,S1)$ and $D(R,S2)$ respectively. Procedures

FMSC and *FindOp* outlined below allow searching through the tree for those edit operations which yields an R satisfying established requirements.

```
Let:
$Q_{S1}^{S2}$:minimum cost edit sequence to transform $S_1$ into $S_2$.
$d$: difference between cumulative $S1$ and $S2$.
$r$: the better consecutive symbols corresponding to $d$ difference.
function FindOp($op_i$,$a_{S1}$,$a_{S2}$) : ($d$, $r$)
/* $op_i$: index of the operation $op \in Q_{S1}^{S2}$ to analyze if is applied or not */
/* $a_{S1} = 0$: cumulative distance of applied editions over $D(R, S1)$ */
/* $a_{S2} = 0$: cumulative distance of applied editions over $D(R, S2)$ */
/* $better = (\infty, \emptyset)$: local better result */
if ($op_i == 0$) then
    $better = (a_{S1} - a_{S2}, \emptyset)$
else
    case $Q_{S1}^{S2}[op_i]$ :
      - $w(b_{S1}, \varepsilon, )$: /* Deletion */
            /* Rejected */
            $(d_{no}, r_{no})$ = FindOp($op_i - 1$,$a_{S1}$,$a_{S2} + e(w(b_{S1}, \varepsilon))$)
            /* Accepted */
            $(d_{yes}, r_{yes})$ = FindOp($op_i - 1$,$a_{S1} + e(w(\varepsilon, b_{S1}))$,$a_{S2}$)
            if ($|d_{yes}| < |d_{no}|$) then
                $better = (d_{yes}, r_{yes} \cup \{b_{S1}\})$
            else
                $better = (d_{no}, r_{no})$
            end if
      - $w(\varepsilon, b_{S2})$: /* Insertion */
            /* Rejected */
            $(d_{no}, r_{no})$ = FindOp($op_i - 1$,$a_{S1}$,$a_{S2} + e(w(\varepsilon, b_{S2}))$)
            /* Accepted */
            $(d_{yes}, r_{yes})$ = FindOp($op_i - 1$,$a_{S1} + e(w(b_{S2}, \varepsilon))$,$a_{S2}$)
            if ($|d_{yes}| < |d_{no}|$) then
                $better = (d_{yes}, r_{yes} \cup \{b_{S2}\})$
            else
                $better = (d_{no}, r_{no})$
            end if
      - $w(b_{S1}, b_{S2})$: /* Substitution */
            foreach symbol $m$ satisfying (3) and (4)
                $(d, r)$ = FindOp($op_i - 1$,$a_{S1} + e(w(m, b_{S1}))$,$a_{S2} + e(w(m, b_{S2}))$)
                if ($|d| < |better|$) then
                   $better = (d, r \cup \{m\})$
                end if
            end foreach
    end case
end if
return $better$
end function

procedure FMSC($S_1$,$S_2$)
/* $S_1$ and $S_2$: strings to compute its median $R$
  - compute $D(S_1, S_2)$ to get $Q^{S2}_{S1}$
  - $(d, r)$ = FindOp($L_{Q^{S2}_{S1}}$,0,0)
  - return $r$
end procedure
```

Table 1. Substitution cost between two symbols. $e(w(\cdot, \cdot))$

	a	b	c	d	e
a	0	1	2	3	4
b	1	0	1	2	3
c	2	1	0	1	2
d	3	2	1	0	1
e	4	3	2	1	0

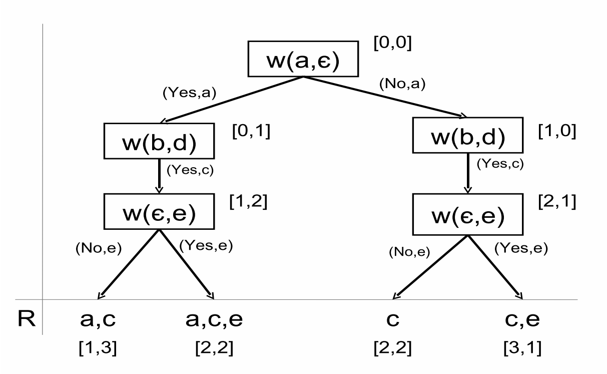


Fig. 1. Each branch represents a possible set of operations to get R from S_1

3 Editing Algorithm

Let T a set of instances. Wilson [17] based editing procedures such [4][13] remove all misclassified instances, t_i, by its k-nearest neighbors (K-NN). This kind of editing cleans interclass overlapping regions while the boundaries between classes are smoothed. A K-NN classifier that uses the edited set as training set could be improved its classification results respect the classification through the original dataset.

As point Wilson [18], in some cases editing has be done carefully because the algorithm may remove a lot of instances, while spoiling the generalization capabilities.

When $k \geq 1$, a wrong classification of t_i does not mean that no one k-nearest neighbors belongs to the same class of t_i. Thus, it is reasonable to think that t_i does not need to be an outlier, but can be a boundary instance useful for next classifications.

The proposed approach aims to face successfully this problem by adding to T an artificial instance R computed from t_i and its same-class nearest neighbor, t_j, if it belongs to k-nearest neighbors. If R, tagged as t_i, satisfies $D(R, t_i) \leq D(t_i, t_j)$ and (2). Its inclusion boosts the chance t_i will be correctly reclassified since by definition R lies in the t_i k-neighborhood. Moreover, this can be viewed as some space regions poorly covered from the original instances become better represented. From these assumptions can be guessed this editing scheme leads to lower classification errors versus the original dataset. Artificial instance R will be computed by the procedure **F**ast **M**edian **S**tring **C**omputation, FMSC for short, described at section 2.2, as $R = FMSC(t_i, t_j)$. The algorithm sketched below allows compute the edited set.

```
Let:
$T$:instance set to edit.
$K$:number of near neighbors.
foreach instance $t_i$ in $T$

 - classify $t_i$ by its k-near neighbors in $T - t_i$.
   if wrong classified then
    - find $t_j$, the k-near same-class neighbor of $t_i$.
      if exist $t_j$ then
       - build $R = FMSC(t_i, t_j)$.
       - make $T = T \cup R$.
      else
       - mark $t_i$ to deletion.
      end if
   end if

end foreach
- delete from $T$ all market instances.
```

3.1 Computational Cost Analysis

Computing the median R from strings S_1 and S_2 involves the calculation of $D(S_1, S_2)$, which can be accomplished in $\mathcal{O}(L_{S1} \times L_{S2})$ as was pointed at subsection 2.1. From definition of this distance $L_{Q_{S1}^{S2}} = L_{S1} + L_{S2}$ for the worst case, i.e when there are no substitutions.

Searching through the tree with *FindOp* can be viewed as evaluating all possibilities to assigning $\{accepted/rejected\}$ to every q_i in Q_{S1}^{S2}, so there are $2^{L_{Q_{S1}^{S2}}}$ chances, this is, the number of branches on the tree. Denoting a rejected operation by $\sim q_i$, let $Op = \{q_1, \sim q_2, \sim q_3, ..., q_n\}$ be a possible assignation and q_i, $i < n$, an arbitrary operation. Clearly, the estimate a_{S1} to $D(R, S_1)$ associated with Op can be decomposed as

$a_{S_1} = a_{S_1}^{0..i} + a_{S_1}^{i+1..n}$ where $a_{S_1}^{k..h}$ denotes the cumulative contribution of those operations $q_k, .., q_h$ to a_{S1}, this holds also for a_{S2}.

Be Op' a new assignment derived from Op by changing the $\{accepted/rejected\}$ tag to some operations $\{q_0, .., q_i\}$ while $\{q_{i+1}, .., q_n\}$ gets unchanged. Consequently the value a'_{S1} related to Op' is $a'_{S_1} = a'^{0..i}_{S_1} + a_{S_1}^{i+1..n}$, an expression which is partially calculated before, similarly can be computed a'_{S2}. Moreover, if those assignations to $\{q_{i+1}, .., q_n\}$ minimizes $|a_{S_1}^{i+1..n} - a_{S_2}^{i+1..n}|$ thus, the optimal sequence of assignations is one which have $\{q_{i+1}, .., q_n\}$ as subsequence.

Considerations above allows to speed up computation of *FindOp* procedure by applying a dynamic programming approach leading a $\mathcal{O}(max\{L_{S1} \times L_{S2}, L_{Q_{S1}^{S2}} \times D_s\})$, where $D_s = D(S_1, S_2)$.

The editing procedure needs to classify every instance at T, which requires an $\mathcal{O}(|T|^2)$ time. A second steps involves computing *FMSC* for every wrong classified instance having a same-class k-near neighbor. The worst case, all $|T|$ instances will need to be processed, so this step entails $\mathcal{O}(|T| \times \textit{FindOp})$ time.

4 Experimental Results

The behavior of the proposed algorithms was analyzed using two sets of strings (Freeman Chaincodes). Digits and character contours from the NIST 3 DATABASE with 26 and 10 classes respectively.

To evaluate the editing algorithm a sample of 80 instances per class was drawn and each set splits in 4-fold to use a crossvalidation technique. At a first stage, for a fixed value of K, all training sets were edited by the Wilson procedure and each test set classified by the K-NN rule using the respective edited set. Latter, each original training set was edited but this time by our proposed approach classifying again the test sets. As

Table 2. Average error rate (4-folds) as percent for classification with different edited sets. (Characters set).

		K on Edition															
		K=3		K=5		K=7		K=9		K=11		K=13		K=15		K=17	
K on Classif.	Not Edited	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson
1	13.7	16.5	14.6	15.8	13.8	16.3	13.8	16.8	13.8	17.1	**13.4**	17.1	**13.4**	17.5	**13.2**	17.8	**13.1**
3	14.7	17.6	15.3	17.6	14.6	17.5	**14.4**	17.8	**14.2**	18.8	**14.0**	18.9	**13.9**	19.6	**13.8**	19.9	**13.3**
5	15.4	17.6	**15.2**	17.9	**14.4**	17.9	**14.1**	18.5	**14.1**	18.9	**14.2**	19.5	**14.1**	19.8	**13.8**	20.2	**13.7**
7	16.0	19.4	16.2	19.6	**15.1**	19.9	**15.2**	19.8	**14.7**	20.2	**14.2**	20.7	**14.0**	20.8	**14.0**	21.3	**14.1**
9	17.1	19.5	**16.3**	20.1	**15.5**	20.3	**15.3**	20.5	**14.8**	21.0	**14.8**	21.6	**14.5**	21.8	**14.6**	22.2	**14.6**
11	17.7	20.0	17.7	20.8	**16.5**	20.8	**16.1**	21.3	**15.4**	21.6	**15.0**	22.3	**15.0**	22.3	**15.2**	23.1	**15.0**
13	18.3	21.0	**18.2**	21.2	**17.1**	21.7	**16.4**	22.1	**16.5**	22.5	**15.8**	22.8	**15.5**	23.3	**15.5**	23.7	**15.7**
15	18.6	22.0	18.9	21.6	**18.0**	22.3	**17.1**	22.6	**16.7**	23.3	**16.2**	23.5	**16.3**	23.7	**16.0**	24.2	**16.0**
17	19.6	22.1	**18.8**	22.7	**18.0**	23.0	**17.5**	23.8	**17.4**	24.0	**16.9**	24.0	**16.5**	24.6	**16.5**	24.8	**16.4**

Table 3. Average error rate (4-folds) as percent for classification with different edited sets. (Digits set).

		K on Edition															
		K=3		K=5		K=7		K=9		K=11		K=13		K=15		K=17	
K on Classif.	Not Edited	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson	Wilson	JJWilson
1	1.8	2.8	1.9	2.6	1.8	2.5	1.8	2.9	**1.6**	2.8	**1.6**	2.8	**1.6**	2.8	**1.5**	2.8	**1.5**
3	2.0	3.1	2.3	2.9	2.3	3.0	2.0	3.3	**1.9**	3.4	2.0	3.4	2.0	3.8	**1.9**	3.9	**1.8**
5	3.0	3.6	**2.9**	3.6	**2.8**	3.8	**2.8**	4.3	**2.6**	4.3	**2.6**	4.1	**2.5**	4.3	**2.3**	4.4	**2.5**
7	3.5	4.3	3.5	4.3	**2.9**	4.3	**2.9**	4.5	**2.6**	4.6	**2.6**	4.8	**2.6**	5.0	**2.5**	5.1	**2.4**
9	3.6	4.3	**3.5**	4.1	**3.3**	4.3	**3.0**	4.6	**2.9**	4.9	**2.8**	4.9	**2.8**	5.1	**2.8**	5.4	**2.8**
11	4.1	4.5	**2.9**	4.5	**2.9**	4.6	**2.9**	4.6	**3.0**	4.6	**3.6**	4.8	**3.5**	4.9	**3.5**	5.3	**3.4**
13	4.4	4.8	**3.1**	4.8	**3.3**	5.0	**3.1**	5.0	**3.3**	5.3	**4.0**	5.6	**3.5**	5.9	**3.5**	5.9	**3.4**
15	4.8	5.1	**3.8**	5.1	**3.8**	5.4	**3.6**	5.5	**3.8**	5.5	**4.5**	6.1	**4.3**	6.3	**4.1**	6.3	**3.9**
17	4.9	5.1	**4.1**	5.1	**4.0**	5.4	**3.8**	5.5	**3.9**	5.5	**4.6**	6.1	**4.2**	6.0	**4.3**	6.1	**4.3**

a baseline, the original training sets classficantion is used. At each fold, editing was repeated for odds values of K from 3 to 17, while in the classification stage, the range was from 1 to 17. Remaining a total of 288 trials on each dataset. As distance, the Levenshtein distance was chosen which is described in the section 2.1.

Tables 2 and 3 show some results for the 4-fold experiments when the test set uses: the original training set, different edited sets for the characters and the digits datasets, respectively. Through these experiments, Wilson procedure never reduces the baseline error rate (classification with original tranining sets), while our proposed approach, labeled as JJWilson, is to able to improve by 79.1% of the trials in the case of characters dataset and by 77.7% in the case of digits dataset. These improvements are highlighted in bold type in the tables of results. So, the experiments in both datasets show that the proposed algorithm for editing outperforms the Wilson approach, with respect to the error rate reduction.

5 Conclusions and Future Work

A novelty method was presented to edit a dataset of contours encoded by Freeman Chaincodes. In addition, a new fast procedure to compute the median between two strings based on a string edit distance is explained. Experiments show that the edit scheme behaves well on the studied datasets. Further investigations can be addressed to revise the method to identify the misclassified instance, and consider other near neigbor belonging to the same class instead the nearest one to build the new prototype. Also, others datasets could be studied and compared with additional edit methods. Moreover, our fast median string algorithm between two examples could be extended to compute the average of N examples.

Acknowledgements

This work is partially supported by the Spanish CICYT under project DPI2006-15542-C04-01, the Spanish MICINN through project TIN2009-14205-CO4-01 and by the Spanish research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

References

1. Cárdenas, R.: A Learning Model for Multiple-Prototype Classification of Strings. In: 17th International Conference on Pattern Recognition, vol. 4, pp. 420–442 (2004)
2. Devijver, I., Kittler, J.: On the edited nearest neighbour rule. In: 5th Int. Conf. on Pattern Recognition, pp. 72–80 (1980)
3. Duta, N., Jain, A., Dubuisson-Jolly, M.: Automatic Construction of 2D Shape Models. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 433–446 (2001)
4. Ferri, F., Vidal, E.: Comparison of several editing and condensing techniques for colour image segmentation and object location. Pattern Recognition and Image Analysis (1992)
5. Jiang, X., Schiffmann, L., Bunke, H.: Computation of median shapes. In: 4th Asian Conference on Computer Vision (2000)
6. Koplowitz, J., Brown, T.: On the relation of performance to editing in nearest neighbour rules. Pattern Recognition 13, 251–255 (1981)
7. Levenshtein, V.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics 10, 707–710 (1966)
8. Martínez, C., Juan, A., Casacubierta, F.: Median strings for k-nearest neighbour classification*1. Pattern Recognition Letters 24, 173–181 (2003)
9. Olvera, J., Martínez, F.: Edition schemes based on BSE. In: 10th Iberoamerican Congress on Pattern Recognition, pp. 360–368 (2005)
10. Penrod, C., Wagner, T.: Another look at the edited neares neighbour rule. IEEE Trans. on Systems, Man and Cybernetics 7, 92–94 (1977)
11. Rico-Juan, J.R., Micó, L.: Comparison of AESA and LAESA search algorithms using string and tree-edit-distances. Pattern Recognition Letters 24, 1417–1426 (2003)
12. Sánchez, J., Pla, F., Ferri, F.: Using the nearest centroid neighbourhood concept for editing purposes. In: 7th Symposium National de Reconocimiento de Formas y Análisis de Imágen, vol. 1, pp. 175–180 (1997)
13. Tomek, I.: An experiment with the edit nearest neighbour. IEEE Trans. on Systems, Man and Cybernetics 6, 448–452 (1976)
14. Tomek, I.: A generalization of the k-NN rule. IEEE Trans. on Systems, Man and Cybernetics 6, 121–126 (1976)
15. Vázquez, F., Sánchez, J., Pla, F.: A stochastic approach to Wilson's editing algorithm. In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds.) IbPRIA 2005. LNCS, vol. 3523, pp. 35–42. Springer, Heidelberg (2005)
16. Wagner, R., Fischer, M.: The String-to-String Correction Problem. Journal of the ACM 21, 168–173 (1974)
17. Wilson, D.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. on Systems, Man. and Cybernetics 2, 408–421 (1972)
18. Wilson, D., Martínez, T.: Reduction techniques for instance based learning algorithms. Machine Learning 38, 257–286 (2000)

Author Index