Pasi Fränti Gavin Brown Marco Loog
Francisco Escolano Marcello Pelillo (Eds.)

LNCS 8621

Structural, Syntactic, and Statistical Pattern Recognition

Joint IAPR International Workshop, S+SSPR 2014
Joensuu, Finland, August 20–22, 2014
Proceedings

Springer

Lecture Notes in Computer Science 8621

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Pasi Fränti Gavin Brown Marco Loog
Francisco Escolano Marcello Pelillo (Eds.)

Structural, Syntactic, and Statistical Pattern Recognition

Joint IAPR International Workshop, S+SSPR 2014
Joensuu, Finland, August 20-22, 2014
Proceedings

Springer

Volume Editors

Pasi Fränti
University of Eastern Finland, Joensuu, Finland
E-mail: pasi.franti@uef.fi

Gavin Brown
The University of Manchester, UK
E-mail: gavin.brown@manchester.ac.uk

Marco Loog
Delft University of Technology, The Netherlands
E-mail: m.loog@tudelft.nl

Francisco Escolano
Universidad de Alicante, Spain
E-mail: sco@dccia.ua.es

Marcello Pelillo
Università Ca' Foscari Venezia, Venezia Mestre, Italy
E-mail: pelillo@dais.unive.it

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-662-44414-6 e-ISBN 978-3-662-44415-3
DOI 10.1007/978-3-662-44415-3
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014945232

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

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

IAPR Technical Committees TC1 and TC2 organized the Joint International Workshops on Statistical Techniques in Pattern Recognition (SPR), and Structural and Syntactic Pattern Recognition (SSPR) during August 20-22, 2014, in Joensuu, Finland. It is official biennial satellite event prior to the International Conference on Pattern Recognition (ICPR). The workshops (S+SSPR) were hosted by the School of Computing, University of Eastern Finland.

The workshops received 78 submissions, of which 53 were selected into a three-day program of 11 sessions. These include 47 original submissions (included in the proceedings), and six journal track presentations (only abstracts included). Original submissions were reviewed by three Program Committee members, on average. The presentations cover the core topics of pattern recognition methodology including clustering, graph kernels, graph edit distance, discriminant analysis, graph models and embedding, combining and selecting, metrics and dissimilarities, partial supervision and applications. Keynote talks were given by Prof. Ali Shokoufandeh from Drexel University (Philadelphia, USA) about approximation of hard combinatorial problems via embedding to hierarchically separated trees, and Prof. David Hand from Imperial College (London, UK) on evaluating supervised classification methods: error rate, ROC curves, and beyond.

This is the first time that S+SSPR was organized in Nordic countries. For many attendees, it was their first time to visit Finland, which has a unique mix of nature and modern technology. The country has 5.4 million inhabitants, 7 million mobile phones, 9 million Internet connections, 2 million saunas, and nearly 187,888 lakes. Joensuu is located by the beautiful lake Finland, and is the main campus of the University of Eastern Finland.

We thank all the Program Committee members and local organizers for their contributions toward making the workshops happen. Detailed information can be found on the workshop's website: http://cs.uef.fi/ssspr2014/.

August 2014

Pasi Fränti
Gavin Brown
Marco Loog
Francisco Escolano
Marcello Pelillo

Organization

General Chair

Pasi Fränti	University of Eastern Finland, Finland

SPR Chairs

Gavin Brown	University of Manchester, UK
Marco Loog	Delft University of Technology, The Netherlands

SSPR Chairs

Francisco Escolano	University of Alicante, Spain
Marcello Pelillo	University of Venice, Italy

Invited Speakers

Ali Shokoufandeh	Drexel University, USA
David Hand	Imperial College, UK

SPR Committee

Battista Biggio	University of Cagliari, Italy
Fransesc J. Ferri	University of Valencia, Spain
Ana Fred	Technical University of Lisbon, Portugal
Giogio Fumera	University of Cagliari, Italy
Michal Haindl	UTIA, Czech Republic
Laurent Heutte	University of Rouen, France
Tin Kam Ho	Bell Laboratories, USA
Tomi Kinnunen	University of Eastern Finland, Finland
Mineichi Kudo	Hokkaido University, Japan
Ludmila Kuncheva	Bangor University, UK
James Kwok	Hong Kong University, China
Heikki Kälviäinen	Lappeenranta University of Technology, Finland
Nikunj C. Oza	NASA, USA
Filiberto Pla	Jaume I University, Spain
Robi Polikar	Rowan University, USA
Fabio Roli	University of Cagliari, Italy
Clara I. Sánchez	Radboud University, The Netherlands

Carlo Sansone	University of Naples Federico II, Italy
David Tax	Delft University of Technology, The Netherlands
Francesco Tortorella	Cassino University, Italy
Seiichi Uchida	Kyushu University, Japan
Giorgio Valentini	University of Milan, Italy
Jinghao Xue	University College London, UK
Terry Windeatt	University of Surrey, UK
David Windridge	University of Surrey, UK

SSPR Committee

Terry Caelli	University of Melbourne, Australia
Tiberio Caetano	University of Melbourne, Australia
Mario Figueiredo	Technical University of Lisbon, Portugal
Marco Gori	University of Siena, Italy
Edwin Hancock	University of York, UK
Atsushi Imiya	IMIT Chiba University, Japan
Walter G. Kropatsch	Vienna University of Technology, Austria
Arjan Kuijper	Technical University of Darmstadt, Germany
Christoph Lampert	IST, Austria
Xuelong Li	Chinese Academy of Sciences, China
Frank Nielsen	Sony Corporation, Japan
Richard Nock	University of the French West Indies and Guiana, France
Tapio Pahikkala	University of Turku, Finland
Novi Quadrianto	University of Cambridge, UK
Antonio Robles-Kelly	University of Melbourne, Australia
Anand Rangarajan	University of Florida, USA
Samuel Rota Bulò	University of Venice, Italy
Salvatore Tabbone	Université de Nancy, France
Sinisa Todorovic	Oregon State University, USA
Andrea Torsello	University of Venice, Italy
Richard Wilson	University of York, UK

Local Organizers

Pasi Fränti	University of Eastern Finland, Finland
Tomi Kinnunen	University of Eastern Finland, Finland
Oili Kohonen	University of Eastern Finland, Finland
Rahim Saeidi	University of Eastern Finland, Finland
Ville Hautamäki	University of Eastern Finland, Finland
Rosa González Hautamäki	University of Eastern Finland, Finland
Radu Mariescu-Istodor	University of Eastern Finland, Finland
Sami Sieranoja	University of Eastern Finland, Finland
Md. Sahidullah	University of Eastern Finland, Finland

Sponsoring Institutions

International Association of Pattern Recognition
Technical Committee 1. Statistical Pattern Recognition Techniques
Technical Committee 2. Structural and Syntactical Pattern Recognition
Joensuun yliopiston tukisäätiö
Federation of Finnish Learned Societies
University of Eastern Finland

Journal Track Abstracts

Comparison between supervised and unsupervised learning of probabilistic linear discriminant analysis mixture models for speaker verification

Timur Pekhovsky, Aleksandr Sizov
Pattern Recognition Letters, 34 (11), 1307–1313, August 2013.

We present a comparison of speaker verification systems based on unsupervised and supervised mixtures of probabilistic linear discriminant analysis (PLDA) models. This paper explores current applicability of unsupervised mixtures of PLDA models with Gaussian priors in a total variability space for speaker verification. Moreover, we analyze the experimental conditions under which this application is advantageous, taking into account the existing limitations of training database sizes, provided by the National Institute of Standards and Technology (NIST). We also present a full derivation of the Maximum Likelihood learning procedure for PLDA mixture. Experimental results for a cross-channel NIST Speaker Recognition Evaluation (SRE) 2010 verification task show that unsupervised PLDA mixture is more effective than other state-of-the-art methods. We show that for this task a combination of a homogeneous i-vector extractor and a mixture of two Gaussian PLDA models is more effective than a cross-channel i-vector extractor with a single Gaussian PLDA.

Improving distance based image retrieval using non-dominated sorting genetic algorithm

Miguel Arevalillo-Herráez, Francesc J. Ferri, and Salvador Moreno-Picot
Pattern Recognition Letters, `http://dx.doi.org/10.1016/j.patrec.2014.05.008`

Relevance feedback has been adopted as a standard in Content Based Image Retrieval (CBIR). One major difficulty that algorithms have to face is to achieve and adequate balance between the exploitation of already known areas of interest and the exploration of the feature space to find other relevant areas. In this paper, we evaluate different ways to combine two existing relevance feedback methods that place unequal emphasis on exploration and exploitation, in the context of distance-based methods. The hybrid approach proposed has been evaluated by using three image databases of various sizes that use different descriptors. Results show that the hybrid technique performs better than any of the original methods, highlighting the benefits of combining exploitation and exploration in relevance feedback tasks.

Information-theoretic selection of high-dimensional spectral features for structural recognition

Boyan Bonev, Francisco Escolano, Daniela Giorgi, and Silvia Biasotti
Computer Vision and Image Understanding, 117 (3), 214–228, March 2013

Pattern recognition methods often deal with samples consisting of thousands of features. Therefore, the reduction of their dimensionality becomes crucial to make the data sets tractable. Feature selection techniques remove the irrelevant and noisy features and select a subset of features which describe better the samples and produce a better classification performance. In this paper, we propose a novel feature selection method for supervised classification within an information-theoretic framework. Mutual information is exploited for measuring the statistical relation between a subset of features and the class labels of the samples. Traditionally it has been measured for ranking single features; however, in most data sets the features are not independent and their combination provides much more information about the class than the sum of their individual prediction power. We analyze the use of different estimation methods which bypass the density estimation and estimate entropy and mutual information directly from the set of samples. These methods allow us to efficiently evaluate multivariate sets of thousands of features. Within this framework we experiment with spectral graph features extracted from 3D shapes. Most of the existing graph classification techniques rely on the graph attributes. We use unattributed graphs to show what is the contribution of each spectral feature to graph classification. Apart from succeeding to classify graphs from shapes relying only on their structure, we test to what extent the set of selected spectral features are robust to perturbations of the dataset.

The active geometric shape model: A new robust deformable shape model and its applications

Quan Wang, and Kim Boyer
Computer Vision and Image Understanding, 116 (12), 1178–1194, Dec 2012.

We present a novel approach for fitting a geometric shape in images. Similar to active shape models and active contours, a force field is used in our approach. But the object to be detected is described with a geometric shape, represented by parametric equations. Our model associates each parameter of this geometric shape with a combination of integrals (summations in the discrete case) of the force field along the contour. By iteratively updating the shape parameters according to these integrals, we are able to find the optimal fit of the shape in the image. In this paper, we first explore simple cases such as fitting a line, circle, ellipse or cubic spline contour using this approach. Then we employ this technique to detect the cross-sections of subarachnoid spaces containing cerebrospinal fluid (CSF) in phase-contrast magnetic resonance (PC-MR) images, where the object

of interest can be described by a distorted ellipse. The detection results can be further used by an s–t graph cut to generate a segmentation of the CSF structure. We demonstrate that, given a properly configured geometric shape model and force field, this approach is robust to noise and defects (disconnections and non-uniform contrast) in the image. By using a geometric shape model, this approach does not rely on large training datasets, and requires no manual labeling of the training images as is needed when using point distribution models.

Multi-label learning under feature extraction budgets

Pekka Naula, Antti Airola, Tapio Salakoski, and Tapio Pahikkala
Pattern Recognition Letters, 40 (15), 56–65, April 2014.

We consider the problem of learning sparse linear models for multi-label prediction tasks under a hard constraint on the number of features. Such budget constraints are important in domains where the acquisition of the feature values is costly. We propose a greedy multi-label regularized least-squares algorithm that solves this problem by combining greedy forward selection search with a cross-validation based selection criterion in order to choose, which features to include in the model. We present a highly efficient algorithm for implementing this procedure with linear time and space complexities. This is achieved through the use of matrix update formulas for speeding up feature addition and cross-validation computations. Experimentally, we demonstrate that the approach allows finding sparse accurate predictors on a wide range of benchmark problems, typically outperforming the multi-task lasso baseline method when the budget is small.

Semi-supervised nearest mean classification through a constrained log-likelihood

Marco Loog, and Are C. Jensen
IEEE Transactions on Neural Networks and Learning Systems
http://dx.doi.org/10.1109/TNNLS.2014.2329567

We cast a semi-supervised nearest mean classifier, previously introduced by the first author, in a more principled log-likelihood formulation that is subject to constraints. This, in turn, leads us to make the important suggestion to not only investigate error rates of semi-supervised learners but to also consider the risk they originally aim to optimize. We demonstrate empirically that in terms of classification error, mixed results are obtained when comparing supervised to semi-supervised nearest mean classification, while in terms of log-likelihood on the test set, the semi-supervised method consistently outperforms its supervised counterpart. Comparisons to self-learning, a standard approach in semi-supervised learning, are included to further clarify the way in which our constrained NMC improves over regular, supervised nearest mean classification.

Table of Contents

Graph Kernels

Clustering

Graph Edit Distance

Graph Models and Embedding

Discriminant Analysis

Combining and Selecting

Joint Session

Metrics and Dissimilarities

Applications

Partial Supervision

Poster Session

A Graph Kernel from the Depth-Based Representation

Lu Bai[1], Peng Ren[2], Xiao Bai[3], and Edwin R. Hancock[1,⋆]

[1] Department of Computer Science, University of York, York, UK
[2] College of Information and Control Engineering, China University of Petroleum, China
[3] School of Computer Science and Engineering, Beihang University, Beijing, China

Abstract. In this paper we develop a novel graph kernel by matching the depth-based substructures in graphs. We commence by describing how to compute the Shannon entropy of a graph using random walks. We then develop an h-layer depth-based representations for a graph, which is effected by measuring the Shannon entropies of a family of K-layer expansion subgraphs derived from a vertex of the graph. The depth-based representations characterize graphs in terms of high dimensional depth-based complexity information. Based on the new representation, we establish a possible correspondence between vertices of two graphs that allows us to construct a matching-based graph kernel. Experiments on graphs from computer vision datasets demonstrate the effectiveness of our kernel.

Keywords: Depth-based representation, graph matching, graph kernels.

1 Introduction

Graph-based representations are widely used in computer vision and pattern recognition for characterizing shapes and structures [1, 2]. In this context, there has recently been an increasing interest in evolving graph kernels into kernel machines (e.g., a Support Vector Machine (SVM)) for graph classification [3–5]. A graph kernel is usually defined in terms of a (dis)similarity measure between graphs. Haussler [6] proposed a general graph kernel formulation referred to as R-convolution kernel, which is effected by decomposing graphs into substructures separately and then measuring the pairwise (dis)similarities between the resulting substructures. For a pair of sample graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$, suppose $\{\mathcal{S}_{p;1}, \ldots, \mathcal{S}_{p;x}, \ldots, \mathcal{S}_{p;N_p}\}$ and $\{\mathcal{S}_{q;1}, \ldots, \mathcal{S}_{p;y}, \ldots, \mathcal{S}_{q;N_q}\}$ are the sets of the substructures of $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ respectively. A R-convolution kernel k_R between $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ can be defined as

$$k_R(G_p, G_q) = \sum_{x=1}^{N_p} \sum_{y=1}^{N_q} s(\mathcal{S}_{p;x}, \mathcal{S}_{q;y}),$$

where $s(\mathcal{S}_{p;x}, \mathcal{S}_{q;y})$ is the (dis)similarity measure between the substructures $\mathcal{S}_{p;x}$ and $\mathcal{S}_{q;y}$, and k_R proves to be a positive definite kernel.

From the perspective of R-convolution, existing graph kernels can be generally categorized into three classes [3], i.e. graph kernels based on comparing all pairs of a)

⋆ Edwin R. Hancock is supported by a Royal Society Wolfson Research Merit Award.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 1–11, 2014.

walks, b) paths and c) restricted subgraph and subtree structures. One major limitation of these existing graph kernels is that in practical computation they do not easily scale up to structures of large sizes. To overcome this problem, most existing graph kernels compromise to use substructures of limited sizes, and examples include a) the shortest path graph kernel [7], b) the graphlet count graph kernel [3], c) the fast neighborhood subgraph pairwise distance kernel [9], and d) the backtrackless kernel [10] based on non-backtracking cycles that are identified by the Ihara zeta function [11]. Although this strategy curbs the notorious inefficiency of comparing large substructures, graph kernels with limited sized substructures simply can only reflect restricted topological characteristics of a graph.

In this work, we aim to overcome the topological restrictions by characterizing graph substructures in terms of depth-based representations [12], which provide richer topological features but also easily scale up to as large size as the original graph. To this end, we investigate how to incorporate the depth-based representations into graph matching and thus develop a novel graph kernel which not only reflects the rich depth-based structure of graphs but also enables a fast computation. We commence by computing the depth-based complexity traces [13] of a graph around each vertex. To avoid the burdensome subgraph enumeration in computing the intrinsic complexity [12], we compute the depth-based representation around a vertex by measuring a fast Shannon entropy of its expansion subgraph. The depth-based representation gauges the Shannon entropy flow via the expansion subgraphs, and thus reflects a high dimensional complexity characteristics of the graph around the vertex. Based on the obtained depth-based representations for two graphs we develop a matching strategy similar to that Scott et al. [16] previously used for point set matching. The purpose of this step is to match the vertices of the graphs by using the vertex information extracted from the depth-based representations. For a pair of graphs, we use the Euclidean distance between the depth-based representations to compute an affinity matrix. The correspondences between pairwise vertices are obtained from the affinity matrix. The affinity matrix characterizes local structural similarity between a pair of graphs and can be used for graphs of different sizes. Finally, we develop the novel depth-based graph matching kernel by counting the matched vertex pairs. We empirically demonstrate the effectiveness and efficiency of our new graph kernel on graphs from computer vision datasets.

The remainder of this paper is organized as follows. Section 2 presents the definition of depth-based representations for graphs. Section 3 presents the definition of the new graph matching kernel. Section 4 provides our experimental evaluations. Finally, Section 5 concludes our work.

2 Depth-Based Representations

We commence by introducing a fast Shannon entropy measure for a graph. Moreover, we show how to compute a depth-based representation around a vertex of a graph.

2.1 The Shannon Entropy of a Graph

We compute the Shannon entropy of a graph based on steady state random walks on the graph. Consider a graph $G(V, E)$ where V denotes the set of vertices and $E \subseteq V \times V$

denotes the set of undirected edges. The adjacency matrix A for $G(V,E)$ is a symmetric $|V| \times |V|$ matrix with the (v,u)th entry

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

The vertex degree matrix of $G(V,E)$ is a diagonal matrix D whose vth diagonal element is given by $D(v,v) = d(v) = \sum_{v,u \in V} A(v,u)$. As a result, the probability of a steady state random walk on $G(V,E)$ visiting vertex v is $P_G(v) = d(v)/\sum_{v,u \in V} d(u)$. The Shannon entropy of $G(V,E)$ associated with the steady state random walk is

$$H_S(G) = -\sum_{v \in V} P_G(v) \log P_G(v). \tag{2}$$

For a graph $G(V,E)$ ($|V| = n$), computing the Shannon entropy $H_S(G)$ requires time complexity $O(n^2)$. This is because $H_S(G)$ relies on the degree matrix D that is computed by visiting all the n^2 entries of the adjacency matrix A. This indicates that the Shannon entropy H_S defined in Eq.(2) can be efficiently computed. By contrast, the von Neumnann entropy defined in [14] and the Shannon entropy associated with an information functional defined in [15] both require time complexity $O(n^3)$.

2.2 The Depth-Based Representation for a Graph

For an undirected graph $G(V,E)$, the shortest path $S_G(v,u)$ between a pair of vertices v and u can be computed by using Dijkstra algorithm. The matrix S_G whose element $S_G(v,u)$ represents the shortest path length between v and u is referred to as the shortest path matrix for G. Let N_v^K be a subset of V satisfying $N_v^K = \{u \in V \mid S_G(v,u) \leq K\}$. For G, the K-layer expansion subgraph $\mathcal{G}_v^K(\mathcal{V}_v^K; \mathcal{E}_v^K)$ around vertex v is

$$\begin{cases} \mathcal{V}_v^K &= \{u \in N_v^K\}; \\ \mathcal{E}_v^K &= \{(u,v) \subset N_v^K \mid (u,v) \in E\}. \end{cases} \tag{3}$$

Assume L_{max} is the greatest length of the shortest paths from v to the remaining vertices of $G(V,E)$. If $L_v \geq L_{max}$, the L_v-layer expansion subgraph is $G(V,E)$ itself.

Definition 2.1 (h-layer depth-based representation). For a graph $G(V,E)$ and a vertex $v \in V$, the h-layer depth-based representation around v is a h dimensional vector

$$D_G^h(v) = [H_S(\mathcal{G}_v^1), \cdots, H_S(\mathcal{G}_v^K), \cdots, H_S(\mathcal{G}_v^h)]^T \tag{4}$$

where h ($h \leq L_v$) is the length of the shortest paths from v to other vertices in $G(V,E)$, $\mathcal{G}_v^K(\mathcal{V}_v^K; \mathcal{E}_v^K)$ ($K \leq h$) is the K-layer expansion subgraph of $G(V,E)$ around v, and $H_S(\mathcal{G}_v^K)$ is the Shannon entropy of $\mathcal{G}_v^K$ and is defined in Eq.(2). □

For a graph $G(V,E)$ ($|V| = n$) and a vertex $v \in V$, computing the h-layer depth-based representation $D_G^h(v)$ of $G(V,E)$ around v requires time complexity $O(hn^2)$. This follows the definitions in Eq.(3). For $G(V,E)$, the Dijkstra algorithm requires time complexity $O(n^2)$. Computing the Shannon entropies of the h K-layer expansion

subgraphs, which are derived from v, requires time complexity $O(hn^2)$. Hence, the whole time complexity is $O(hn^2)$. This indicates that the h-layer depth-based representation around a vertex of a graph can be efficiently computed. Key to this efficiency is that the Shannon entropy on an expansion subgraph only requires time complexity $O(n^2)$. By contrast, in [12] the intrinsic complexity measure of an expansion subgraph for measuring the depth-based complexity requires time complexity $O(n^5)$.

Moreover, the h-layer depth-based representation $D_G^h(v)$ characterizes the depth-based complexity of $G(V,E)$ with regard to the vertex v in a h dimensional feature space. It captures the rich depth-based complexity characteristics of substructures around the vertex v in terms of the entropies of the K-layer expansion subgraphs with K increasing from 1 to h. In contrast, the existing graph kernels in the literatures [4, 4, 5] tend to compute similarities on global subgraphs of limited sizes and can only capture restricted characteristics of graphs.

3 Depth-Based Graph Matching Kernel

We describe how the depth-based representations can be used for graph matching. Furthermore, we define a novel graph kernel based on the proposed matching method.

3.1 Depth-Based Graph Matching

We develop a matching method similar to that introduced in [16, 17] for point set matching, which computes an affinity matrix in terms of the distances between points. In our work, for a vertex v of $G(V,E)$, we treat the h-layer depth-based representations $D_G^h(v)$ as the point coordinate associated with v. We use the Euclidean distance between the depth-based representations $D_{G_p}^h(v_i)$ and $D_{G_q}^h(u_j)$ as the distance measure of the pairwise vertices v_i and u_j of graphs $G_p(V_p,E_p)$ and $G_q(V_q,E_q)$, respectively. The affinity matrix element $R(i,j)$ is defined as

$$R(i,j) = \sqrt{[D_{G_p}^h(v_i) - D_{G_q}^h(u_j)]^T[D_{G_p}^h(v_i) - D_{G_q}^h(u_j)]}. \tag{5}$$

where R is a $|V_p| \times |V_q|$ matrix. The element $R(i,j)$ represents the dissimilarity between the vertex v_i in $G_p(V_p,E_p)$ and the vertex u_j in $G_q(V_q,E_q)$. The rows of $R(i,j)$ index the vertices of $G_p(V_p,E_p)$, and the columns index the vertices of $G_q(V_q,E_q)$. If $R(i,j)$ is the smallest element both in row i and in column j, there should be a one-to-one correspondence between the vertex v_i of G_p and the vertex u_j of G_q. We record the state of correspondence using the correspondence matrix $C \in \{0,1\}^{|V_p||V_q|}$ satisfying

$$C(i,j) = \begin{cases} 1 \text{ if } R(i,j) \text{ is the smallest element} \\ \quad \text{both in row } i \text{ and in column } j; \\ 0 \text{ otherwise.} \end{cases} \tag{6}$$

Eq.(6) implies that if $C(i,j) = 1$, the vertices v_i and v_j are matched. Note that, in row i or column j there may be two or more than two elements satisfying Eq.(6). In other words, for a pair of graphs a vertex from a graph may have two or more than two

matched vertices from the other graph. To assign a vertex one matched vertex at most, we update the matrix C by employing the Hungarian method that is widely used for solving the assignment problem (e.g., the bipartite graph matching problem) in polynomial time [18]. Here the matrix $C \in \{0,1\}^{|V_p||V_q|}$ can be seen as the incidence matrix of a bipartite graph $G_{pq}(V_p, V_q, E_{pq})$, where V_p and V_q are the two sets of partition parts and E_{pq} is the edge set. By performing the Hungarian algorithm on the incidence matrix $C \in \{0,1\}^{|V_p||V_q|}$ (i.e., the correspondence matrix of G_p and G_q) of the bipartite graph G_{pq}, we assign each vertex from G_p or G_q at most one matched vertex from the other graph G_q or G_p. Note finally that, directly performing the Hungarian algorithm on the matrix R can also assign each vertex from G_p or G_q an unique matched vertex. However, it cannot guarantee that each identified element is the smallest both in the row and column in R. This is because some vertices will not have matched vertices.

For a pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ ($|V_p| = |V_q| = n$). Computing the correspondence matrix $C \in \{0,1\}^{|V_p||V_q|}$ (i.e. the final correspondence matrix updated by the Hungarian algorithm) requires time complexity $O(hn^3)$. This follows the definition in Section 3.1. For G_p, computing its n h-layer depth-based representations derived from each of its vertices requires time complexity $O(hn^3)$, and it is the same for G_q. Computing each element of the affinity matrix R requires time complexity $O(h)$, and hence computing the whole affinity matrix R requires time complexity $O(hn^2)$. The computation of the correspondence matrix C need to enumerate all the n^2 pairs of elements in R and thus requires time complexity $O(n^2)$. The Hungarian algorithm on the matrix C requires time complexity $O(n^3)$. As a result, the whole time complexity is $O(hn^3)$.

3.2 A Depth-Based Graph Kernel

Based on the graph matching strategy in Section 3.1, we define a new graph kernel.

Definition 3.1 (The depth-based graph kernel). Consider $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ as a pair of sample graphs. Based on the definitions in Eq.(4), Eq.(5) and Eq.(6), and the Hungarian algorithm, we compute the correspondence matrix C. The depth-based graph kernel $k_{DB}^{(h)}$ using the h-layer depth-based representations of the graphs is

$$k_{DB}^{(h)}(G_p, G_q) = \sum_{i=1}^{|V_p|} \sum_{j=1}^{|V_q|} C(i,j). \tag{7}$$

which counts the number of matched vertex pairs between $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$. Note that, the kernel $k_{DB}^{(h)}$ can also accommodate attributed graphs by computing the number of pairwise matched vertices that have the same vertex label. □

Lemma 3.1. *The depth-based graph kernel $k_{DB}^{(h)}$ is positive definite (**pd**).*

Proof. Intuitively, the proposed depth-based graph kernel is **pd** because it counts pairs of matched vertices (i.e., the smallest subgraphs). More formally, let the base kernel k be a function counting pairs of matched vertices in the pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$

$$k(G_p, G_q) = k_{DB}^{(h)}(G_p, G_q) = \sum_{v_i \in V_p} \sum_{u_j \in V_q} \delta(v_i, u_j). \tag{8}$$

where

$$\delta(v_i, u_j) = \begin{cases} 1 \text{ if } C(i,j) = 1; \\ 0 \text{ otherwise.} \end{cases} \tag{9}$$

where δ is the Dirac kernel, that is, it is 1 if the arguments are equal and 0 otherwise (i.e. it is 1 if a pair of vertices are matched and 0 otherwise). Hence the proposed kernel function $k_{DB}^{(h)}$ is the sum of several positive definite Dirac kernels, and is thus **pd**. ■

The depth-based graph kernel $k_{DB}^{(h)}$ on a pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ ($|V_p| = |V_q| = n$) requires time complexity $O(hn^3)$. For the pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$, computing their correspondence matrix C in terms of h-layer depth-based representations requires time complexity $O(hn^3)$, and counting the number of matched vertex pairs from the matrix C needs to enumerate all the n^2 pairs of elements in C. Hence, the whole time complexity of the depth-based graph kernel $k_{DB}^{(h)}$ is $O(hn^3)$. This indicates that our depth-based graph kernel $k_{DB}^{(h)}$ can be computed in polynomial time. Key to this efficiency is that the required h-layer depth-based representations and the corresponding matching can be efficiently computed.

Discussions. The depth-based graph kernel is related to the depth-based representation defined in [13]. However, there are two significant differences. First, the depth-based representation in [13] is computed by measuring the complexities of subgraphs from the centroid vertex, which is identified by evaluating the minimum shortest path length variance to the remaining vertices. By contrast, we compute the h-layer depth-based representation for each vertex as a point coordinate. Second,the depth-based representation from the centroid vertex can be seen as an embedding vector. Embedding a graph into a vector tends to approximate the structural correlations in a low dimensional space, and thus leads to information loss. By contrast, the depth-based graph kernel computed by matching the h-layer depth-based representation characterizes graphs in a high dimensional space and thus better preserves graph structure.

4 Experimental Results

4.1 Graph Datasets from the SHREC 3D Shape and COIL Image Databases

We demonstrate the performance of our kernel on several standard graph datasets from computer vision databases (i.e., the SHREC 3D Shape and COIL image databases).

a) BAR31, b) BSPHERE31 and c) GEOD31: The SHREC 3D Shape database consists of 15 classes and 20 individuals per class, that is 300 shapes [20]. This is a usual benchmark in 3D shape recognition. From the SHREC 3D Shape database, we establish three graph datasets named BAR31, BSPHERE31 and GEOD31 datasets through three mapping functions. These functions are a) ERG barycenter: distance from the center of

mass/barycenter, b) ERG bsphere: distance from the center of the sphere that circumscribes the object, and c) ERG integral geodesic: the average of the geodesic distances to the all other points. The number of maximum, minimum and average vertices for the three datasets are a) 220, 41 and 95.42 (for BAR31), b) 227, 43 and 99.83 (for BSPHERE31), and c) 380, 29 and 57.42 (for GEOD31), respectively.

d) COIL5: We establish a COIL5 dataset from the COIL database. The COIL image database consists of images of 100 3D objects. We use the images for the first five objects. For each object we employ 72 images captured from different viewpoints. For each image we first extract corner points using the Harris detector, and then establish Delaunay graphs based on the corner points as vertices. As a result, in the dataset there are 5 classes of graphs, and each class has 72 testing graphs. The number of maximum, minimum and average vertices for the dataset are 241, 72 and 144.90 respectively.

Table 1. Classification Accuracy (In $\%$ $\pm$ Standard Error) Using C-SVM and Runtime

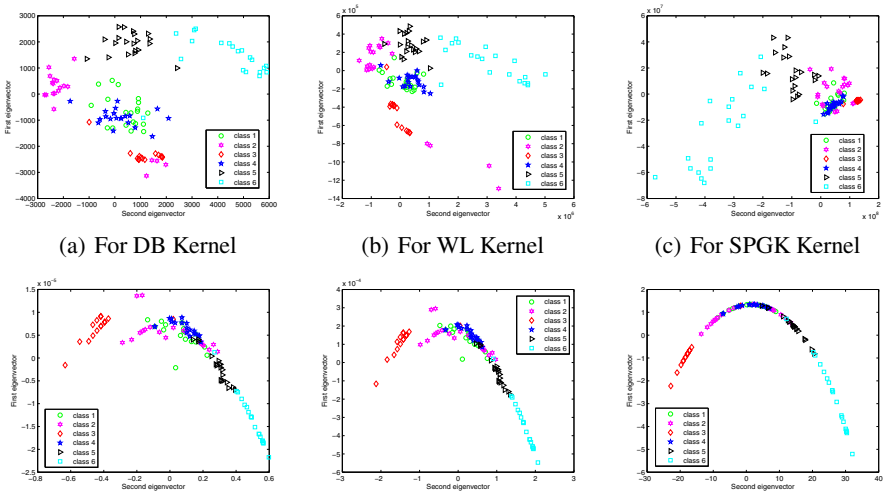
Datasets	DB	WL	SPGK	GCGK	GCGK4	JSK
BAR31	**69.40** $\pm$ .56	58.53 $\pm$ .53	55.73 $\pm$ .44	22.96 $\pm$ .65	23.40 $\pm$ .60	24.10 $\pm$ .86
BSPHERE31	**56.43** $\pm$ .69	42.10 $\pm$ .68	48.20 $\pm$ .76	17.10 $\pm$ .60	18.80 $\pm$ .50	21.76 $\pm$ .53
GEOD31	**42.83** $\pm$ .50	38.20 $\pm$ .68	38.40 $\pm$ .65	15.30 $\pm$ .68	22.36 $\pm$ .55	18.93 $\pm$ .50
COIL5	**74.22** $\pm$ .41	33.16 $\pm$ 1.01	69.97 $\pm$ .92	67.00 $\pm$ .55	68.77 $\pm$ .56	57.25 $\pm$ .46

Experimental Setup: a) First, we evaluate the performance of our depth-based graph kernel (DB) on graph classification problems. We also compare our kernel with several alternative state of the art graph kernels. These graph kernels include 1) the Weisfeiler-Lehman subtree kernel (WL) [3], 2) the shortest path graph kernel (SPGK) [7], 3) the graphlet count graph kernel with graphlets of size 3 (GCGK) and size 4 (GCGK4) [8], and 4) the Jensen-Shannon graph kernel (JSK) with the von Neumann entropy (i.e., the approximated von Neumann entropy [22] computed through the vertex degree) [21]. For our DB kernel, we set h as 10. For the WL kernel, we set the highest dimension (i.e. the highest height of subtrees) of the Weisfeiler-Lehman isomorphism as 10. For each kernel, we compute the kernel matrix on each graph dataset. We perform 10-fold cross-validation using the C-Support Vector Machine (C-SVM) Classification to compute the classification accuracies, using LIBSVM [23]. We use nine samples for training and one for testing. All the C-SVMs were performed along with their parameters optimized on each dataset. We repeat the experiment 10 times. We report the average classification accuracies and standard errors for each kernel in Table.1. **b)** Second, we evaluate the performance of different kernels on graph clustering problems. We commence by performing the kernel Principle Component Analysis (kPCA) on the kernel matrix to embed graphs into a 2-dimensional principal space. We visualize the embedding results of each kernel using the first two principal components. The embedding results on the BAR31 and COIL5 datasets are shown in Fig.1 and Fig.2 respectively. Note that, for the BAR31 dataset we only visualize the embedding points of the first six classes of graphs. For each kernel, the embedding results on the BSPHERE31 and GEOD31 datasets are similar to that on the BAR31 dataset. The space in the paper is also not sufficient to include all of the results obtained. Thus, we only show the results on the BAR31 dataset. Finally, to place our analysis of graph clustering on a more quantitative footing, for

each kernel we apply the K-means method to all the kernel embeddings. We calculate the Rand Index for the resulting clusters. The Rand indicating each kernel is listed in Table 2.

4.2 Experiments on Graph Datasets

Experimental Results and Discussions: a) In terms of the classification accuracies, we observe that the accuracies of our DB kernel are the greatest for any dataset. The performance of our kernel exceeds that of all other kernels. The reason for its effectiveness is that the required depth-based representations of graphs used in our framework capture a high dimensional depth-based complexity information of graphs. In contrast, the alternative graph kernels with limited sized substructures (including the vertex degree required for the JSK) only capture local topological information and reflects restricted characteristics of graphs. Moreover, we also observe that the classification performance of our kernel is more stable than that of the alternative kernels. This verifies again that our kernel defined by depth-based matching reflects precise similarities of graphs. **b)** In terms of the embedding results, it is clear that our DB kernel produces the best clusters. The different classes are separated better than other kernels on any dataset. Note that, for the COIL5 dataset the 72 images for each object are taken from different viewing directions spaced at intervals of 5° around the object. Hence, the embedded graphs for each class are expected to form a circular trajectory rather than a cluster in the feature space. In the light of this observation, our method shows a greater representational power in terms of giving a more trajectory-like embedding than the alternative methods. Moreover, Table 2 indicates that our DB kernel outperforms all the alternative kernels for all the object classes studied on any dataset. These observations verify that our proposed kernel has good ability to distinguish different classes of graphs.



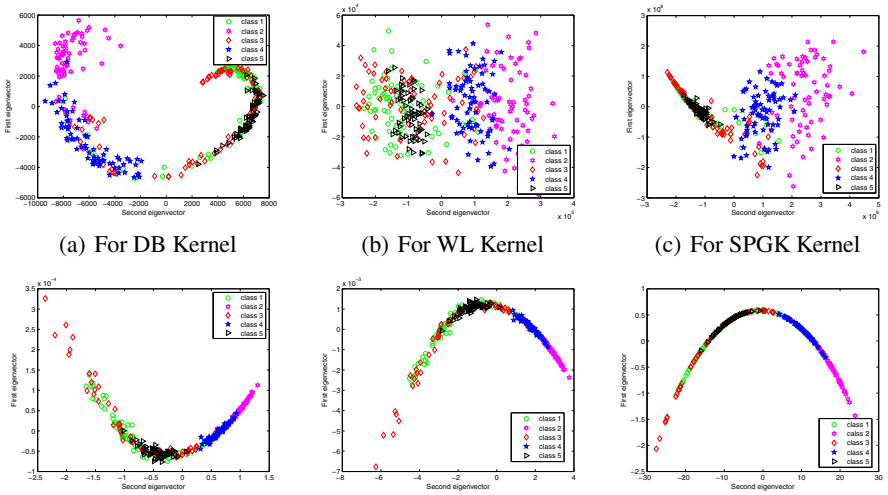
(a) For DB Kernel (b) For WL Kernel (c) For SPGK Kernel

(d) For GCGK Kernel (e) For GCGK4 Kernel (f) For JSK Kernel

Fig. 1. Clusters of Graphs from the BAR31 Dataset

Table 2. Rand Index for K-means Method

Datasets	DB	WL	SPGK	GCGK	GCGK4	JSK
BAR31	**0.2319**	0.2047	0.1734	0.1638	0.1641	0.1697
BSPHERE31	**0.1615**	0.1304	0.1582	0.1210	0.1238	0.1202
GEOD31	**0.1502**	0.1136	0.1142	0.1002	0.1207	0.1025
COIL5	**0.4436**	0.3503	0.4124	0.4119	0.4272	0.3295



(a) For DB Kernel (b) For WL Kernel (c) For SPGK Kernel

(d) For GCGK Kernel (e) For GCGK4 Kernel (f) For JSK Kernel

Fig. 2. Clusters of Graphs from the COIL5 Dataset

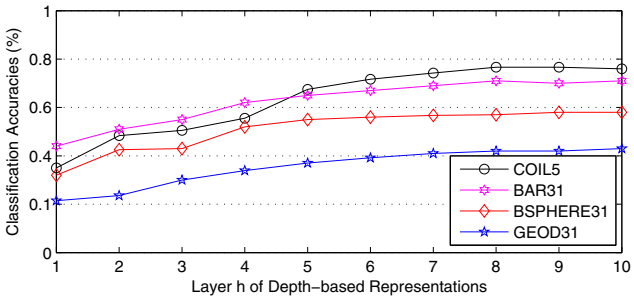


Fig. 3. The Accuracy with Different h Layer

Comparisons with Increasing h: To take our study one step further, we evaluate the performance of our DB graph kernel on graph datasets with increasing h. Here, we evaluate how the classification accuracies vary with increasing h (i.e. $h = 1, 2, \ldots, 10$). We report the results in Fig.3, in which the x-axis gives the varying of h, and the y-axis gives the classification accuracies of our DB kernel. The lines of different colours represent the results on different datasets. The classification accuracies tend to become greater with increasing h. This is because the greater the h, the higher dimensional depth-based complexity information of our kernel can be captured.

5 Conclusion

In this paper, we have described how to construct a depth-based graph kernel in terms of matching graphs based on the depth-based representations. The depth-based representations for graphs capture a high dimensional depth-based complexity information of graphs. Furthermore, our matching strategy incorporates structural correspondence into the kernel. We have empirically demonstrated the effectiveness and efficiency of our new kernel on synthetic graphs and real-world graphs abstracted from computer vision datasets.

References

1. Harchaoui, Z., Bach, F.: Image classification with segmentation graph kernels. In: Proc. CVPR (2007)
2. Barra, V., Biasotti, S.: 3D shape retrieval using Kernels on Extended Reeb Graphs. Pattern Recognition 46, 2985–2999 (2013)
3. Shervashidze, N., Schweitzer, P., Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-Lehman graph kernels. Journal of Machine Learning Research 1, 1–48 (2010)
4. Gärtner, T., Flach, P.A., Wrobel, S.: On graph kernels: Hardness results and efficient alternatives. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT/Kernel 2003. LNCS (LNAI), vol. 2777, pp. 129–143. Springer, Heidelberg (2003)
5. Jebara, T., Kondor, R.I., Howard, A.: Probability product kernels. Journal of Machine Learning Research 5, 819–844 (2004)
6. Haussler, D.: Convolution kernels on discrete structures. In Technical Report UCS-CRL-99-10, UC Santa Cruz (1999)
7. Borgwardt, K.M., Kriegel, H.P.: Shortest-path kernels on graphs. In: Proc. ICDM, pp. 74–81 (2005)
8. Shervashidze, N., Vishwanathan, S.V.N., Petri, T., Mehlhorn, K., Borgwardt, K.M.: Efficient graphlet kernels for large graph comparison. Journal of Machine Learning Research 5, 488–495 (2009)
9. Costa, F., Grave, K.D.: Fast neighborhood subgraph pairwise distance kernel. In: Proc. ICML, pp. 255–262 (2010)
10. Aziz, F., Wilson, R.C., Hancock, E.R.: Backtrackless walks on a graph. IEEE Trans. Neural Netw. Learning Syst. 24(6), 977–989 (2013)
11. Ren, P., Wilson, R.C., Hancock, E.R.: Graph characterization via Ihara coefficients. IEEE Transactions on Neural Networks 22(2), 233–245 (2011)
12. Escolano, F., Hancock, E.R., Lozano, M.A.: Heat diffusion: Thermodynamic depth complexity of networks. Physical Review E 85, 036206 (2012)
13. Bai, L., Hancock, E.R.: Depth-based complexity traces of graphs. Pattern Recognition 47(3), 1172–1186 (2014)
14. Dehmer, M., Mowshowitz, A.: A history of graph entropy measures. Information Sciences 181, 57–78 (2011)
15. Anand, K., Bianconi, G., Severini, S.: Shannon and von Neumann entropy of random networks with heterogeneous expected degree. Physical Review E 83, 036109 (2011)
16. Scott, G.L., Longuett-Higgins, H.C.: An algorithm fro associating the features of two images. Proc. the Royal Society of London B 244, 313–320 (1991)
17. Xiao, B., Hancock, E.R., Wilson, R.C.: A generative model for graph matching and embedding. Computer Vision and Image Understanding 113(7), 777–789 (2009)

18. Munkres, J.: Algorithms for the assignment and transportation Problems. Journal of the Society for Industrial and Applied Mathematics 5(1), 32–38 (1957)
19. Borgwardts, K.M.: Graph Kernels. PhD thesis, Munchen (2007)
20. Biasotti, S., Marini, S., Mortara, M., Patané, G., Spagnuolo, M., Falcidieno, B.: 3D shape matching through topological structures. In: Nyström, I., Sanniti di Baja, G., Svensson, S. (eds.) DGCI 2003. LNCS, vol. 2886, pp. 194–203. Springer, Heidelberg (2003)
21. Bai, L., Hancock, E.R.: Graph kernels from the Jensen-Shannon divergence. Journal of Mathematical Imaging and Vision 47(1-2), 60–69 (2013)
22. Han, L., Escolano, F., Hancock, E.R., Wilson, R.C.: Graph characterizations from von Neumann entropy. Pattern Recognition Letters 33(15), 1958–1967 (2012)
23. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines (2011), Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

Incorporating Molecule's Stereisomerism within the Machine Learning Framework

Pierre-Anthony Grenier[1], Luc Brun[1], and Didier Villemin[2]

[1] GREYC UMR CNRS 6072,
Caen, France
[2] LCMT UMR CNRS 6507,
Caen, France
{pierre-anthony.grenier,luc.brun,didier.villemin}@ensicaen.fr

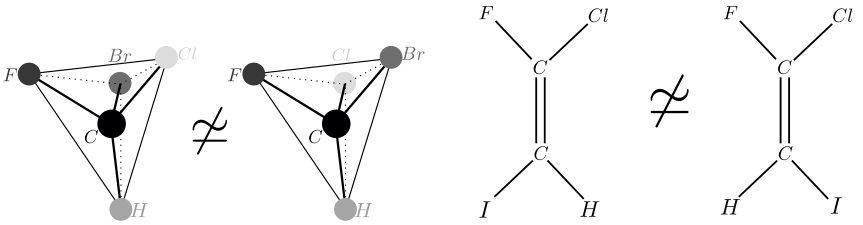
Abstract. An important field of chemoinformatics consists in the prediction of molecule's properties, and within this field, graph kernels constitute a powerful framework thanks to their ability to combine a natural encoding of molecules by graphs, with classical statistical tools. Unfortunately some molecules encoded by a same graph and differing only by the three dimensional orientation of their atoms in space have different properties. Such molecules are called stereoisomers. These latter properties can not be predicted by usual graph methods which do not encode stereoisomerism. In this paper we propose to encode the stereoisomerism property of each atom of a molecule by a local subgraph. A kernel between bags of such subgraphs provides a similarity measure incorporating stereoisomerism properties. We then propose two extensions of this kernel incorporating in each sub graph information about its surroundings.

1 Introduction

A molecular graph is a graph $G = (V, E, \mu, \nu)$, where each node $v \in V$ encodes an atom and each edge $e \in E$ a bond between two atoms. The labelling functions μ and ν associate to each vertex and each edge a label encoding respectively the nature of the atom (carbon, oxygen,...) and the type of the bond (single, double, triple or aromatic). However, those graphs have a limitation: they do not encode the spatial configuration of atoms. Some molecules, called stereoisomers, are associated to a same molecular graph but differ by the relative positioning of their atoms.

Most of stereoisomers are characterized by the three dimensional orientation of the direct neighbors of a single atom or two connected atoms. We can consider for example, a carbon atom, with four neighbors, each of them located on a summit of a tetrahedron. If we permute two of the atoms, we obtain a different spatial configuration and hence an alternative stereoisomer (Figure 1(a)). An atom is called a stereocenter if a permutation of two atoms belonging to its neighborhood produces a different stereoisomer. We should stress here that, to a large extend, stereoisomerism is independent of a particular embedding of a molecule. Indeed, in Figure 1(a), any particular embedding keeping the same relative positioning of atoms H, Cl, Br and F according to the central carbon atom C, would

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 12–21, 2014.



(a) Two different spatial configurations of the neighbors of a carbon

(b) Two different spatial configurations of two carbons linked by a double bond.

Fig. 1. Two types of stereocenters

correspond to a same stereoisomer. In the same way, two connected atoms form a stereocenter if a permutation of the positions of two atoms belonging to the union of their neighborhoods produces a different stereoisomer (Figure 1(b)). According to chemical experts [9], within molecules currently used in chemistry, 98% of stereocenters correspond either to carbons with four neighbors, called asymmetric carbon (Figure 1(a)) or to couples of two carbons adjacent through a double bond (Figure 1(b)). We thus restrict the present paper to such cases.

Graph kernels [10,6], provide a measure of similarity between graphs. Under the assumption that a kernel k is symmetric and definite positive, the value $k(G, G')$, where G and G' encode two graphs, corresponds to a scalar product between two vectors $\Psi(G)$ and $\Psi(G')$ in an Hilbert space. This latter property allows us to combine graph kernels with usual machine learning methods such as SVM or kernel ridge regression by using the well known kernel trick, which consists in replacing the scalar product between $\Psi(G)$ and $\Psi(G')$ by $k(G, G')$ in these algorithms.

Up to now, only few methods have attempted to incorporate stereoisomerism within the graph kernel framework. Brown et al. [2] have proposed to incorporate this information through an extension of the tree-pattern kernel [10]. One drawback of this method is that, patterns which encode stereo information, and patterns which do not, are combined without any weighting in the final kernel value. So for a property only related to stereoisomerism, patterns that do not encode stereo information may be assimilated to noise which deteriorates the prediction. Grenier et al. [8] have introduced the minimal subtree which characterizes a stereocenter within an acyclic molecule. They also proposed a kernel based on this minimal subtree, which takes into account stereoisomerism. This kernel is however restricted to acyclic graphs.

Based on [8], we present in Section 2 an encoding of molecules distinguishing stereoisomers. Section 3 present the construction of a subgraph, which allows to characterizes locally a stereocenter. Then in Section 4, we use this subgraph to propose new graph kernels valid for cyclic as well as acyclic molecules, thus overcoming the main limitation of [8]. We finally present in Section 5 results obtained using those kernels and compare these results with state of the art methods.

2 Ordered Graph and Stereo Vertices

The spatial configuration of the neighbors of each atom may be encoded through an ordering of its neighborhood. For example, considering the left part of Figure 1(a), and looking at the central carbon from the hydrogen atom (H), the sequence of remaining neighbors of the carbon: Cl, Br and F may be considered as lying on a plane and are encountered clockwise. Thus, this spatial configuration is encoded by the sequence H, Cl, Br, F and the sequence H, Br, Cl, F encodes the second configuration.

In order to encode this information in a graph, we introduce the notion of ordered graph. An ordered graph $G = (V, E, \mu, \nu, ord)$ is a molecular graph $G_m = (V, E, \mu, \nu)$ together with a function $ord : V \rightarrow V^*$ which maps each vertex to an ordered list of its neighbors. Two ordered graphs G and G' are isomorphic ($G \underset{o}{\simeq} G'$) if there exists an isomorphism f between their respective molecular graphs G_m and G'_m such that $ord'(f(v)) = (f(v_1) \dots f(v_n))$ with $ord(v) = (v_1 \dots v_n)$ (where $N(v) = \{v_1, \dots, v_n\}$ denotes the neighborhood of v). In this case f is called an ordered isomorphism between G and G'.

However, different ordered graphs may encode a same molecule. In the example of the left part of Figure 1(a), if we look to the central carbon from a different neighbor, we can obtain a different sequence, for example F, Br, Cl, H, that represents the same configuration but now considered from the atom F. We thus have to define an equivalence relationship between ordered graphs, such that two ordered graphs are equivalent if they represent a same configuration.

To do so, we introduce the notion of re-ordering function σ, which associates to each vertex $v \in V$ of degree n a permutation $\sigma(v)$ on $\{1, \dots, n\}$, which allows to re-order its neighborhood. The graph with re-ordered neighborhoods $\sigma(G)$ is obtained by mapping for each vertex v its order $ord(v) = v_1 \dots v_n$ onto the sequence $v_{\sigma(v)(1)} \dots v_{\sigma(v)(n)}$ where $\sigma(v)$ is the permutation applied on v.

In order to define a permutation $\sigma(v)$ for each vertex of a graph, we first introduce the notion of potential asymmetric carbon which corresponds to a carbon with four neighbors. Such a vertex corresponds to a stereocenter if one permutation of two of its neighbors provides a different stereoisomer (Section 1). Permutations associated to a potential asymmetric carbon correspond to all even permutations of its four neighbors [11]. For a double bond between two carbons, permutations associated to each carbon of the double bound must have a same parity. Finally, for any vertex which does not correspond to a potential asymmetric carbon nor to a carbon of a double bond, we do not search to characterize its spatial configuration. So these vertices are associated to all possible permutations of their neighbors.

The set of re-ordering functions, transforming an ordered graph into another one representing the same configuration is called a valid family of re-ordering functions Σ [7]. We say that it exists an equivalent ordered isomorphism f between G and G' according to Σ if it exists $\sigma \in \Sigma$ such that f is an ordered isomorphism between $\sigma(G)$ and G' ($\sigma(G) \underset{o}{\simeq} G'$). The equivalent order relationship defines an equivalence relationship [7] and two different stereoisomers are

encoded by non equivalent ordered graphs. We denote by $\text{IsomEqOrd}(G, G')$ the set of equivalent ordered isomorphism between G and G'.

Potential asymmetric carbons, and double bonds between carbons, are not necessarily stereocenters. For example if the label of vertex Br of Figure 1(a) is replaced by Cl, both left and right molecules of Figure 1(a) would be identical. In the same way, if the label of the vertex F in Figure 1(b) is replaced by Cl, the left and right molecules of this figure also become identical. For those cases, any permutation in the ordered list of the carbons would lead to an equivalent ordered graph. We thus define a stereo vertex as a vertex for which any permutation of two of its neighbors produces a non-equivalent ordered graph:

Definition 1 (Stereo vertex). *Let $G = (V, E, \mu, \nu, ord)$ be an ordered graph. A vertex $v \in V$ is called a stereo vertex iff:*

$$\forall (i,j) \ \in \{1, \ldots, |N(v)|\}^2, i \neq j, \nexists f \in \mathit{IsomEqOrd}(G, \tau^v_{i,j}(G)) \text{ with } f(v) = v. \tag{1}$$

where $\tau^v_{i,j}(G)$ corresponds to an ordered graph deduced from G by permuting nodes of index i and j in $ord(v)$.

3 Minimal Stereo SubGraph

Definition 1 is based on the whole graph G to test if a vertex v is a stereo vertex. However, given a stereo vertex s, one can observe that on some configurations, the removal of some vertices far from s should not change its stereo property. In order to obtain a more local characterization of a stereo vertex, we should thus determine a vertex induced subgraph H of G, including s, large enough to characterize the stereo property of s (i.e. $\forall (i,j) \in \{1, \ldots, |N(s)|\}^2, i \neq j, \nexists f \in \text{IsomEqOrd}(H, \tau^s_{i,j}(H))$ with $f(s) = s$), but sufficiently small to encode only the relevant information characterizing the stereo vertex s. Such a subgraph is called a minimal stereo subgraph of s.

We now present an heuristic, used to compute a minimal stereo subgraph of a stereo vertex. We focus our attention on asymmetric carbons. Let H be a subgraph of G containing a stereo vertex s corresponding to an asymmetric carbon. We say that the stereo property of s is not captured by H if (Definition 1):

$$\exists (i,j) \in \{1, \ldots, |N(s)|\}^2, i \neq j, \exists f \in \text{IsomEqOrd}(H, \tau^s_{i,j}(H)) \text{ with } f(s) = s \tag{2}$$

To define a minimal stereo subgraph of s, we consider a finite sequence $(H^k_s)_{k=1}^n$ of vertex induced subgraphs of G. The first element of this sequence H^1_s is the smaller vertex induced subgraph for which we can test (2) :

$$V(H^1_s) = \{s\} \cup N(s)$$

where $V(H^1_s)$ and $N(s)$ denote respectively the set of vertices of H^1_s and the set of neighbors of s in G.

If the current vertex induced subgraph H^k_s does not capture the stereo property of s, we know by (2), that it exists some isomorphisms f of equivalent ordered graphs between H^k_s and $\tau^s_{i,j}(H^k_s)$ with $i \neq j$ and $f(s) = s$. Let us consider

such an isomorphism f. By definition of equivalent ordered isomorphism, it exists $\sigma \in \Sigma$ such that f is an ordered isomorphism between H_s^k and $\sigma\left(\tau_{i,j}^s(H_s^k)\right)$. By definition of ordered isomorphisms, and since $f(s) = s$, we have:

$$\forall l \in \{1, \ldots, |N(s)|\}, f(v_l) = v_{\sigma(s) \circ \tau_{i,j}^s(l)}.$$

with $ord(s) = v_1, \ldots, v_n$.

As $\sigma(s)$ is an even permutation, $\sigma(s) \circ \tau_{i,j}^s$ is an odd one. Hence it exists l in $\{1, \ldots, |N(s)|\}$ such that $l \neq \sigma(s) \circ \tau_{i,j}^s(l)$ and we have $f(v_l) \neq v_l$ and $f^{(2)}(v_l) \neq f(v_l)$. In other words, any equivalent ordered isomorphism corresponding to equation (2) maps at least two vertices in the neighborhood of s in H_s^k onto a different vertex in the same neighborhood. Let us denote by $\mathcal{E}_f^k$ the set of vertices of H_s^k connected to s by a path whose all vertices are mapped onto other vertices by f:

$$\mathcal{E}_f^k = \{v \in V(H_s^k) \mid \exists c = (v_0, \ldots, v_q) \in H_s^k \text{ with } v_0 = s \text{ and } v_q = v \text{ } s.t. \\ \forall r \in \{1, \ldots, q\}, f(v_r) \neq v_r\} \quad (3)$$

For any equivalent ordered isomorphism f satisfying (2), the set $\mathcal{E}_f^k$ is not empty since it contains at least 2 vertices. A vertex v belongs to $\mathcal{E}_f^k$ if neither its label nor its neighborhood in H_s^k allows to differentiate it from $f(v)$. The basic idea of our algorithm consists in enforcing constraints on each $v \in \mathcal{E}_f^k$ at iteration $k+1$ by adding to H_s^k the neighborhood in G of all vertices belonging to $\mathcal{E}_f^k$. This last set is denoted by $N(\mathcal{E}_f^k)$. The set of vertices of the vertex induced subgraph H_s^{k+1} is thus defined by:

$$V(H_s^{k+1}) = V(H_s^k) \cup \bigcup_{f \in \mathcal{F}_s^k} N(\mathcal{E}_f^k) \quad (4)$$

where $\mathcal{F}_s^k$ denotes all equivalent ordered isomorphisms satisfying (2).

Since $f \in \mathcal{F}_s^k$ implies that $\mathcal{E}_f^k$ is not empty, adding iteratively constraints on the existence of vertices in $\mathcal{E}_f^k$ removes f from $\mathcal{F}_s^k$. The algorithm stops when the set $\mathcal{F}_s^k$ becomes empty. Note that such a condition must be satisfied since s is a stereocenter and hence the whole molecule does not satisfy (2).

The intermediate vertex induced subgraphs found by our algorithm are illustrated in Fig. 2. Note that at iteration 2, it exists an equivalent ordered isomorphism $f \in \mathcal{F}_C^2$ mapping the path CCO (bottom right of the figure) onto the same path located on the top right part of Fig 2. In this case $\mathcal{E}_f^2$ contains the three carbons of these two paths and both oxygen atoms. The oxygen atoms belong to $\mathcal{E}_f^2$ since their neighborhoods in H_C^2 does not allow to differentiate them (Fig. 2). At iteration 3, the neighborhood in G of these oxygen atoms are added to H_C^3, hence adding N and Br which allow to differentiate both paths and thus removes the equivalent ordered isomorphism f from $\mathcal{F}_C^3$.

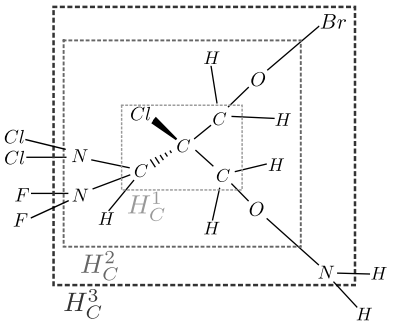


Fig. 2. An asymmetric carbon and its associated sequence $(H_C^k)_{k=1}^3$

4 Stereo Kernel and Extensions

4.1 Stereo Kernel

Given an ordered graph G, we can associate a minimal stereo subgraph to each of its stereocenter. A same stereo subgraph may be present more than once in a given molecule, we thus need to associate a unique code to each such subgraph in order to enumerate efficiently the eventual multiple occurences of a stereo subgraph within a molecule. To do so, we use [13], which associates to each molecule a unique code which allows to test the existence of an equivalent ordered isomorphism between two stereo subgraphs, unlike [1] which allows to find efficiently all isomorphisms between two graphs. We can thus compute the set of minimal stereo subgraphs $\mathcal{H}(G)$ together with the spectrum $spec(G) = (f_H(G))_{H \in \mathcal{H}(G)}$ which encodes the frequency $f_H(G)$ of each $H \in \mathcal{H}(G)$. The set $\mathcal{H}(G)$ and the spectrum *spec* provide a characterisation of each stereo center of G and hence describe the stereoisomerism of G.

The comparison of the spectrum of two ordered graphs, is then used to define a kernel between two molecules taking into account the stereoisomerism:

$$k(G, G') = \sum_{H \in \mathcal{H}(G) \cap \mathcal{H}(G')} K(f_H(G), f_H(G')). \tag{5}$$

where K denotes a kernel between real values (e.g. Gaussian, intersection or polynomial). The choice of a particular kernel, together with its parameters is performed through cross-validation.

4.2 Augmented Labels

Cycles are important sub-parts of molecules, and thus two atoms with identical label could have different influence if one of them is included in a cycle. Thus it can be useful, during the computation of a minimal stereo subgraph, to consider two atoms with a same label, but not included in a same number of cycles, as different.

To do so, we first compute the set of relevant cycles of each ordered graph. Relevant cycles are defined as cycles of a graph which can not be deduced from shorter cycles [12]. We can associate to each vertex v, the number n_v of relevant cycles to which it belongs. Then, for an ordered graph G, this information is added to the label of each of its vertex v ($\mu_A(v) = \mu(v).n_v$) to obtain a new ordered graph G_A. The method described in Section 3 is then applied on the ordered graph G_A. We thus obtain a different set of minimal stereo subgraphs $\mathcal{H}(G_A)$, composed of smaller stereo subgraphs where nodes encode a more global information. As in Section 4.1 we define from this set of subgraphs a spectrum encoding the frequency of each subgraph, and define a kernel between graphs by comparing those spectrum:

$$k(G, G') = \sum_{H \in \mathcal{H}(G_A) \cap \mathcal{H}(G'_A)} K(f_H(G), f_H(G')). \tag{6}$$

4.3 Expanded Subgraphs

Equations 5 and 6 allow to compare two molecules through the distribution of their stereo subgraphs. However those kernels are based on a binary similarity measure between configurations: either two stereo subgraphs are different, and thus the configurations encoded by those subgraphs are dissimilar, or the subgraphs are identical and thus the configurations are similar. By adding information about the adjacency relationships between these stereo subgraphs and the remaining part of the molecule, we can obtain a finer measure of similarity between configurations around stereocenters.

To take into account the adjacency relationships between a stereo subgraph H_s and its surrounding, we consider larger vertex induced subgraphs than H_s. Let H be a subgraphs of G, the neighborhood $N(H)$ of H is the set of vertices of $G - H$ which are neighbors of a vertex of H:

$$N(H) = \{v \in V(G) - V(H) \mid \exists (u, v) \in E \ s.t \ u \in V(H)\}$$

The set of vertex induced subgraphs obtained by adding k of its neighbors to H_s can be used to construct a kernel between graphs. We can also consider subgraphs where vertex located farther from the stereo subgraph than its direct neighborhood are added. However we have to limit the number of vertices we add, in order to keep a local information. Moreover the number of subgraphs increases quickly with the number of added vertices. Indeed, C_N^k subgraphs can be constructed by adding k vertices of $N(H_s)$ to H_s, with $N = |N(H_s)|$. We thus, have to determine a number of vertex to add, which is large enough to characterize the adjacency relationships between a stereo subgraph and the remaining part of a molecule, but also sufficiently small to keep a local information. In our experiment, we have considered subgraphs obtained by adding up to three different neighbors of H_s and those obtained by adding one neighbor v of H_s, and one neighbor of v not included in the neighborhood of H_s.

For each minimal stereo subgraph, we have a set of subgraphs which encodes its adjacency relationships with other parts of the molecule. As in section 4.1, we

associate to those subgraphs a unique code. By adding those subgraphs to the set of minimal stereo subgraphs, we obtain a new set of subgraphs $\mathcal{H}^{\mathcal{E}}(G)$, for which we can associate a spectrum which encodes the frequency of each subgraphs $H \in \mathcal{H}^{\mathcal{E}}(G)$. We thus define a kernel between ordered graphs, which takes into account stereoisomerism, and the adjacency relationships of each stereo subgraphs with its surrounding:

$$k(G, G') = \sum_{H \in \mathcal{H}^{\mathcal{E}}(G) \cap \mathcal{H}^{\mathcal{E}}(G')} K(f_H(G), f_H(G')). \tag{7}$$

5 Experiments

Our first experiment is based on a dataset composed of all the stereoisomers of the perindoprilate [3]. As this molecule has 5 stereocenters, the dataset is composed of $2^5 = 32$ molecules. In this dataset, we try to predict if a molecule inhibit the angiotensin-converting enzyme (ACE). The dataset is split into a training set of 23 molecules, and a test set of 9 molecules, as in [3].

Table 1 shows the results obtained by our kernels and the adaptation of the tree pattern kernel to stereoisomerism [2]. All these kernels are combined with the standard SVM method [4] to classify molecules. As all molecules in the dataset are stereoisomers of each other, methods which do not include stereoisomerism information [10,6] are unable to differentiate any molecule of this dataset and are consequently unable to predict the considered property. Moreover, information not related to stereoisomerism included in kernel [2] consists of the same patterns for all molecules. This leads to add a constant shift to all values of the kernel and hence does not deteriorate the prediction for this dataset. Two stereocenters of the molecules of the dataset have a same minimal stereo subgraph, however one of them contains vertices belonging to a cycle. The stereo kernel (Section 4.1) is not able to distinguish these stereocenters, and hence misclassified molecules containing these stereocenters. By using augmented labels (Section 4.2), these two stereocenters are distinguished, and this distinction allows us to reach a prediction accuracy of 100%. The expanded subgraph (Section 4.3) may also help to differentiate the two stereocenters, but for this dataset, one molecule of the testset remains misclassified due to the same stereocenters which are not sufficiently discriminated by this kernel.

Table 1. Classification of the ACE inhibitory activity of perindopirilates stereoisomers

Method	Accuracy Trainset %	Accuracy Testset %
Stereo Kernel (Section 4.1)	91.3	88.9
Stereo Kernel + Augmented Labels (Section 4.2)	100	100
Stereo Kernel + Expanded subgraph (Section 4.3)	100	88.9
Tree patterns Kernel with stereo information [2]	100	100

The second dataset is a dataset of synthetic vitamin D derivatives, used in [2]. This dataset is composed of 69 molecules containing cycles, with an average of 9 stereocenters per molecule. This dataset is associated to a regression problem, which consists in predicting the biological activities of each molecules. Each kernel is test by using it with the standard SVM regression method [5].

After normalizing the values of the dataset, the standard deviation of the biological activities is equal to 0.258. To choose the different parameters and estimate the performance of each kernel on this dataset we use a nested cross-validation. The outer cross-validation is a leave-one-out procedure, used to compute an error for each molecule of the dataset. For each fold, we use another leave-one-out procedure on the remaining molecules, to compute a validation error. Parameters which provide the lowest root mean squared error on the validation are selected. We obtain for each molecule an error, and report in Table 2, the mean of this distribution of errors together with the confidence interval at 95% of this distribution.

Greatest errors in Table 2 are obtained by methods [10,6] which do not include stereo information. The adaptation of the tree pattern kernel to stereoisomerism [2] improves the results over the two previous methods hence showing the insight of adding stereoisomerism information. Our kernel with no extensions obtain results not as good as [2]. For this experiment the modification of label to incorporate information about cycles, decrease our results. However, the addition of information about relationships between minimal stereo subgraphs, and remaining part of the molecules, allow us to obtain better results than [2]. In this case the best results are obtained by considering only subgraphs including one neighbor of H_s. Our methods use a subgraph isomorphism algorithm, but the minimal stereo subgraph are small and thus we have small execution times as we can see in Table 2.

Table 2. Prediction of the biological activity of synthetic vitamin D derivatives

Method	Mean Error	Confidence interval	Gram's matrices computation (s)
Tree patterns kernel [10]	0.193	± 0.060	230
Treelet kernel [6]	0.207	± 0.064	7
Tree patterns kernel with stereo information [2]	0.138	± 0.043	230
Stereo kernel	0.141	± 0.047	1
Stereo kernel + Augmented Labels (Sec. 4.2)	0.192	± 0.061	3
Stereo kernel + Expanded subgraph (Sec. 4.3)	**0.122**	± 0.041	8

6 Conclusion

The study of stereoisomers constitutes an important subfield of chemistry and thus a major challenge in chemoinformatics. We have proposed in this paper, a graph kernel based on an explicit enumeration of all the stereo subgraphs of a molecule. Each stereo subgraph is associated to a stereo vertex and encodes the

part of the graph which provides the stereo property to this vertex. Based on the notion of stereo subgraphs we propose to describe a molecule by its bag of stereo subgraphs. The similarity between two molecules is then encoded through a graph kernel based on the similarity of both bags. Moreover we propose two extensions of this kernel. One extension consists in adding to the labels of the graphs information about cycles of molecules. The second one consists in considering larger subgraphs encoding relationships between each stereo subgraph and the remaining part of the molecule. Experiments related to stereoisomerism properties demonstrate the relevance of our approach and of its extensions.

Acknowledgements. The authors wish to thank the association CRIHAN for their computing resources.

References

1. Bonnici, V., Giugno, R., Pulvirenti, A., Shasha, D., Ferro, A.: A subgraph isomorphism algorithm and its application to biochemical data. BMC Bioinformatics 14(suppl. 7), S13 (2013)
2. Brown, J.B., Urata, T., Tamura, T., Arai, M.A., Kawabata, T., Akutsu, T.: Compound analysis via graph kernels incorporating chirality. Journal of Bioinformatics and Computational Biology 8(1), 63–81 (2010)
3. Castillo-Garit, J.A., Marrero-Ponce, Y., Torrens, F., Rotondo, R.: Atom-based stochastic and non-stochastic 3d-chiral bilinear indices and their applications to central chirality codification. Journal of Molecular Graphics and Modelling 26(1), 32–47 (2007)
4. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
5. Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A.J., Vapnik, V.: Support vector regression machines. In: NIPS, pp. 155–161 (1996)
6. Gaüzère, B., Brun, L., Villemin, D.: Two New Graphs Kernels in Chemoinformatics. Pattern Recognition Letters 33(15), 2038–2047 (2012)
7. Grenier, P.-A., Brun, L., Villemin, D.: Incorporating stereo information within the graph kernel framework. Technical report, CNRS UMR 6072 GREYC (2013), `http://hal.archives-ouvertes.fr/hal-00809066/`
8. Grenier, P.-A., Brun, L., Villemin, D.: Treelet kernel incorporating chiral information. In: Kropatsch, W.G., Artner, N.M., Haxhimusa, Y., Jiang, X. (eds.) GbRPR 2013. LNCS, vol. 7877, pp. 132–141. Springer, Heidelberg (2013)
9. Jacques, J., Collet, A., Wilen, S.: Enantiomers, racemates, and resolutions. Krieger Pub. Co. (1991)
10. Mahé, P., Vert, J.-P.: Graph kernels based on tree patterns for molecules. Machine Learning 75(1), 3–35 (2008)
11. Petitjean, M.: Chirality in metric spaces. Symmetry, Culture and Science 21, 27–36 (2010)
12. Vismara, P.: Union of all the minimum cycle bases of a graph. The Electronic Journal of Combinatorics 4(1), 73–87 (1997)
13. Wipke, W.T., Dyott, T.M.: Stereochemically unique naming algorithm. Journal of the American Chemical Society 96(15), 4834–4842 (1974)

Transitive State Alignment for the Quantum Jensen-Shannon Kernel

Andrea Torsello[1], Andrea Gasparetto[1], Luca Rossi[2], Lu Bai[3], and Edwin R. Hancock[3]

[1] Università Ca' Foscari Venezia, Italy
[2] University of Birmingham, UK
[3] University of York, UK
torsello@dais.unive.it, andrea.gasparetto@unive.it, l.rossi@cs.bham.ac.uk, {lu,erh}@cs.york.ac.uk

Abstract. Kernel methods provide a convenient way to apply a wide range of learning techniques to complex and structured data by shifting the representational problem from one of finding an embedding of the data to that of defining a positive semidefinite kernel. One problem with the most widely used kernels is that they neglect the locational information within the structures, resulting in less discrimination. Correspondence-based kernels, on the other hand, are in general more discriminating, at the cost of sacrificing positive-definiteness due to their inability to guarantee transitivity of the correspondences between multiple graphs. In this paper we generalize a recent structural kernel based on the Jensen-Shannon divergence between quantum walks over the structures by introducing a novel alignment step which rather than permuting the nodes of the structures, aligns the quantum states of their walks. This results in a novel kernel that maintains localization within the structures, but still guarantees positive definiteness. Experimental evaluation validates the effectiveness of the kernel for several structural classification tasks.

1 Introduction

Structural representations have become increasingly popular due to their representational power. However, the descriptiveness comes at the cost of an increased difficulty in applying standard machine learning and pattern recognition techniques to them, as these usually require data that reside in a vector space. The famous kernel trick allows the focus to be shifted from the vectorial representation of data, which now becomes implicit, to a similarity representation. This allows standard learning techniques to be applied to structural data for which no obvious vectorial representation exists.

One of the most influential works on structural kernels was the generic R-convolution kernel proposed by Haussler [6]. Here graph kernels are computed by comparing the similarity of the basic elements for a given decomposition of the two graphs. Depending on the decomposition chosen, we obtain different

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 22–31, 2014.

kernels. Most R-convolution kernels simply count the number of isomorphic substructures in the two graphs. For example, Kashima et al. [8] compute the kernel by decomposing the graph into random walks, while Borgwardt et al. [3] have proposed a kernel based on shortest paths. Here, the similarity is determined by counting the numbers of pairs of shortest paths of the same length in a pair of graphs. Shervashidze et al. [16] have developed a subtree kernel on subtrees of limited size, where the number of subtrees common between two graphs is computed efficiently using the Weisfeiler-Lehman graph invariant.

One drawback of these kernels is that they neglect the locational information for the substructures in a graph. In other words, the similarity does not depend on the relationships between substructures. As a consequence, these kernels cannot establish reliable structural correspondences between the substructures. This limits the precision of the resulting similarity measure. To overcome this problem, Fröhlich et al. [5] introduced alternative optimal assignment kernels. Here each pair of structures is aligned before comparison. However, the introduction of the alignment step results in a kernel that is not positive definite in general [19]. The problem results from the fact that alignments are not in general transitive. In other words, if σ is the vertex-alignment between graph A and graph B, and π is the alignment between graph B and graph C, in general we cannot guarantee that the alignment between graph A and graph C is $\pi \circ \sigma$. On the other hand, when the alignments are transitive, there is a common simultaneous alignment of all the graphs. Under this alignment, the optimal assignment kernel is simply the sum over all the vertex/edge kernels, which is positive definite since it is the sum of separate positive definite kernels. While lacking positive definiteness the optimal assignment kernels cannot be guaranteed to represent an implicit embedding into a Hilbert space, they have nonetheless been proven to be very effective in classifying structures.

There has recently been an increasing interest in quantum computing because of the potential speed-ups over classical algorithms. Recently Bai et al. [1] introduced a graph kernel based on a Quantum analogue of the Jensen-Shannon divergence between average states of continuous-time quantum walks over the structures to be analyzed. Being based on the divergence which is conjectured to be negative definite [4], the kernel is thought to be positive definite. However it lacks permutational invariance, thus different permutations of the same graphs result in different values of the kernel. This fact, while mitigated by the long range interactions reinforced by the interference patterns in quantum walks, is a rather undesirable property for a structural kernel. For this reason in this paper we modify the kernel by adding a novel alignment step that rather than permuting the nodes of the structures, aligns the quantum states of the walks. This results in a novel kernel that is permutationally invariant and maintains similar localization property of the alignment kernels [5]. Further, we prove that the alignment transformations between multiple structures are transitive and that, for this particular alignment, the kernel is always positive definite.

2 Quantum Mechanical Background

Quantum walks are the quantum analogue of classical random walks. Given a graph $G = (V, E)$, the state space of the continuous-time quantum walk defined on G is the set of the vertices V of the graph. Unlike the classical case, where the evolution of the walk is governed by a stochastic matrix, in the quantum case the dynamics of the walker is governed by a complex unitary matrix i.e., a matrix that multiplied by its conjugate transpose yields the identity matrix. As a consequence, the evolution of the quantum walk is reversible, which implies that quantum walks are non-ergodic and do not possess a limiting distribution. See [9] for an overview of the properties of quantum walks. Using Dirac notation, we denote the basis state corresponding to the walk being at vertex $u \in V$ as $|u\rangle$. Here a ket $|u\rangle$ is simply representing a unit vector associated with state u, for example, if we use the vertices as the basis for the space, $|u\rangle = \mathbf{e}_u$, i.e., the u-th vector in the canonical basis. Conversly, a bra $\langle u|$ is the co-vector obtained taking the conjugate-transpose of $|u\rangle$. A general state of the walk is a complex linear combination of the basis states, such that the state of the walk at time t is defined as $|\psi_t\rangle = \sum_{u\in V} \alpha_u(t)\,|u\rangle$ where the amplitude $\alpha_u(t) \in \mathbb{C}$ and $|\psi_t\rangle \in \mathbb{C}^{|V|}$ are both complex.

At each point in time the probability of the walker being at a particular vertex of the graph is given by the square of the norm of the amplitude of the relative state. More formally, let X^t be a random variable giving the location of the walker at time t. Then the probability of the walker being at the vertex u at time t is given by $\Pr(X^t = u) = \alpha_u(t)\alpha_u^*(t)$ where $\alpha_u^*(t)$ is the complex conjugate of $\alpha_u(t)$. Moreover, in a closed system $\sum_{u\in V} \alpha_u(t)\alpha_u^*(t) = 1$.

The evolution of the walk over graph $G = (V, E)$ is governed by Schrödinger equation, where we take the Hamiltonian of the system to be the graph Laplacian L, which, eliminating scaling constants, yields

$$\frac{d}{dt}\,|\psi_t\rangle = -iL\,|\psi_t\rangle \tag{1}$$

Given an initial state $|\psi_0\rangle$, we can solve Equation 1 to determine the state vector at time t $|\psi_t\rangle = e^{-iLt}\,|\psi_0\rangle = \Phi e^{-i\Lambda t}\Phi^\top\,|\psi_0\rangle$, where $L = \Phi\Lambda\Phi^\dagger$ is the spectral decomposition of the Laplacian matrix.

While a pure state can be naturally described using a single ket vector, in general a quantum system can be in a *mixed state*, i.e., a statistical ensemble of pure quantum states $|\psi_i\rangle$, each with probability p_i. The *density operator* (or *density matrix*) of such a system is defined as

$$\rho = \sum_i p_i\,|\psi_i\rangle\,\langle\psi_i|\,. \tag{2}$$

Density operators are positive unit-trace matrices directly linked with the observables of the (mixed) quantum system. Let O be an observable, i.e., an Hermitian operator acting on the quantum states and providing a measurement. The expected value of the measurement O over a mixed state can be calculated from the density matrix ρ: $\langle O\rangle = \mathrm{tr}\,(\rho O)$, where tr is the trace operator.

The Von Neumann entropy of a density operator ρ is

$$H_N(\rho) = -Tr(\rho \log \rho) = -\sum_j \lambda_j \log \lambda_j \,, \tag{3}$$

where the λ_js are the eigenvalues of ρ. With the Von Neumann entropy to hand, we can define the quantum Jensen-Shannon divergence between two density operators ρ and σ as

$$D_{\rm JS}(\rho, \sigma) = H_N\Big(\frac{\rho+\sigma}{2}\Big) - \frac{1}{2}H_N(\rho) - \frac{1}{2}H_N(\sigma) \tag{4}$$

This quantity is symmetric, bounded between 0 and 1, and negative definite for pure states and is conjectured with ample experimental evidence to be negative definite for all states [4].

Finally, for a graph $G(V, E)$, let $|\psi_t\rangle$ denote the state corresponding to a continuous-time quantum walk that has evolved from time $t = 0$ to time $t = T$. We define the time-averaged density matrix $\rho_G^\dagger$ for $G(V, E)$

$$\rho_G^\dagger = \frac{1}{T}\int_0^T |\psi_t\rangle \langle\psi_t| \, {\rm d}t \,. \tag{5}$$

Let ϕ_{xy} denote the (xy)th element of the matrix of eigenvectors Φ of the Laplacian. Following [14], we compute the (r, c)th element of ρ_T as follows:

$$\rho_G^\dagger(r, c) = \sum_{k=1}^{n}\sum_{y=1}^{n} \phi_{rk}\phi_{cy}\bar{\psi}_k\bar{\psi}_y \frac{1}{T}\int_0^\dagger e^{i(\lambda_y-\lambda_k)t} \, {\rm d}t. \tag{6}$$

If we let $T \to \infty$, Eq.(6) further simplifies to

$$\rho_G^\infty = \sum_{\lambda\in\tilde{\Lambda}} P_\lambda \rho_0 P_\lambda^\top \tag{7}$$

where $\tilde{\Lambda}$ is the set of distinct eigenvalues of the Laplacian matrix L and P_λ is the orthogonal projector onto the eigenspace associated with λ.

3 State-Aligned QJSD Kernel

In [1] the Bai et al. defined a kernel based on the Quantum Jensen Shannon divergence between two continuous-time quantum walks between the graphs. The QJSD kernel was defined as

$$K_{\rm QJSD}(G_1, G_2) = \exp\left(-\beta D_{\rm JS}(\rho_1, \rho_2)\right) \tag{8}$$

where ρ_1 and ρ_2 are the time-averaged density matrices associated with the quantum walks over G_1 and G_2 respectively, and β is a decay parameter of the kernel. The walks are initialized in the starting state

$$|\Psi_0\rangle = \sum_{u\in V} \sqrt{\frac{d_u}{\sum_{v\in V} d_v}} \, |u\rangle \,. \tag{9}$$

The kernel is positive definite under the conjecture that the quantum Jensen-Shannon divergence is negative definite for all states, and exhibited good performance on several graph classification tasks, but its value is dependent on the order under which the nodes are presented due to the mixing term $\frac{\rho+\sigma}{2}$ in the definition of the divergence.

In this paper we solve the permutational invariance problem of the QJSD kernel by adding an alignment step to the computation of the kernel. In contrast to alternative alignment kernels such as [5], the alignment is not performed over the node permutations Σ_n of the graphs. Rather it is performed over the quantum basis under which the walker can be observed. In classical random walks the nodes of the graph provide a preferred basis for observation as the walker cannot be simultaneously on multiple nodes, thus the only available degree of freedom is in the choice of an order within the basis vectors, i.e., the observation basis is fully defined up to a permutation $\pi \in \Sigma_n$. This is in stark contrast with quantum mechanics where, due to quantum superposition, prior to observation a quantum walker can be simultaneously at multiple nodes, and the observation itself can be performed under any quantum superposition of states. This means that any orthogonal basis is valid for observation and, thus, the basis is defined up to a unitary transformation $O \in U(n)$, where $U(n)$ is the Unitary group over $\mathbb{C}^n$.

Following this property, we define a *State-aligned* QJSD kernel as

$$\begin{aligned} K_{\mathrm{SAQJSD}}(G_1, G_2) &= \max_{O \in U(n)} \exp\left(-\beta D_{\mathrm{JS}}(\rho_1, O\rho_2 O^\dagger)\right) \\ &= \exp\left(-\beta \min_{O \in U(n)} D_{\mathrm{JS}}(\rho_1, O\rho_2 O^\dagger)\right) \end{aligned} \tag{10}$$

In the following we will prove some important properties of the state-aligned kernel. Namely we will give a closed form solution to the alignment, prove that the optimal transformation are transitive, and prove that the resulting kernel is positive definite without making use of the negative-definiteness conjecture for the quantum Jensen-Shannon divergence.

3.1 Properties of the State-Aligned QJSD Kernel

We start by enunciating a theorem relating the optimal state-alignment to the eigenvectors of the density matrices. For a proof of this result see [18].

Theorem 1. *Let* $\rho 1 = \Phi_1 \Lambda_1 \Phi_1^\dagger$ *and* $\rho 2 = \Phi_2 \Lambda_2 \Phi_2^\dagger$ *be the singular value decompositions of* ρ_1 *and* ρ_2 *respectively, with the eigenvalues in descending order in both* Λ_1 *and* Λ_2*, then the global minimum of* $\bar{H}_N(O)$ *is attained by* $O^* = \Phi_1 \Phi_2^\dagger$.

This theorem tells us how to efficiently compute the state alignment. Further, this transformation aligns the eigenvectors resulting in a mixed density matrix $\frac{1}{2}(\rho_1 + O^* \rho_2 O^{*\dagger})$ with eigenvalues $\frac{1}{2}(\lambda_1 + \mu_i)$ where $\lambda_1, \ldots, \lambda_n$ and $\mu_1, \ldots, \mu_n$ are the eigenvalues of ρ_1 and ρ_2 respectively taken in descending order with their

multiplicity. This means that the aligned Jensen Shannon divergence only needs the eigenvalues of ρ_1 and ρ_2 to be computed, in fact:

$$\min_{O\in U_n} D_{\rm JS}(\rho_1, O\rho_2 O^\dagger) = \sum_{j=1}^{n} -\frac{\lambda_j+\mu_j}{2}\log\left(\frac{\lambda_j+\mu_j}{2}\right) + \frac{\lambda_j\log(\lambda_j)+\mu_j\log(\mu_j)}{2}. \tag{11}$$

This reduces the computational complexity of computing the kernel for all times at which the mixed density matrix is computed, as we do not need to perform the eigendecomposition of the mixed matrix $\frac{1}{2}(\rho_1+\rho_2)$ for each pair of graphs in the kernel. Rather, we only need to compute the eigenvalues (not the eigenvectors) of all the density matrices beforehand. The resulting complexity for the whole kernel computation is $O(Nn^3+N^2n)$ where N is the number of graphs and n their (maximum) size. In contrast, the QJSD kernel has complexity $O(Nn^3+N^2n^2)$ due to the eigenvalue computation for each pair of graphs.

Further, in the case of the infinite-time mixing matrix, we can significantly reduce the computational burden of computing the eigenvalues of the density matrix, by using a result presented in [14]. There it was proven that the infinite-time mixing matrix commuted with the graph Laplacian. As a consequence, ρ^∞ expressed in the eigenbasis of the Laplacian, is a block diagonal matrix where blocks correspond to eigenspaces associated with a single eigenvalue. Let $L=\Phi\Lambda\Phi^\dagger$, be the spectral decomposition of the graph Laplacian, we denote with Φ_j the matrix formed with the columns of Φ corresponding to the eigenvectors associated with the j-th distinct eigenvalue. The j-th diagonal block of ρ^∞ expressed in the eigenbasis Φ is $\Phi_j^\dagger\rho^\infty\Phi_j$. using Eq. (7) and recalling that $P_j=\Phi_j^\dagger\Phi_j$, we have

$$\Phi_j^\dagger\rho^\infty\Phi_j = \Phi_j^\dagger\rho^\infty\Phi_j = \Phi_j^\dagger\rho^0\Phi_j = \Phi_j^\dagger\left|\psi_0\right\rangle\left\langle\psi_0\right|\Phi_j = \left|\Phi_j^\dagger\psi_0\right\rangle\left\langle\Phi_j^\dagger\psi_0\right| \tag{12}$$

which is a rank 1 matrix with a single non-zero eigenvalue $\lambda_j=\|\Phi_j^\dagger\psi_0\|^2$. Hence, once the singular value decomposition of the graph's Laplacian is to hand, we can compute the eigenvalues of the infinite-time mixing matrix directly, without the need for an additional decomposition. This makes the infinite-time kernel particularly efficient to compute.

It is worth noting that as the graph Laplacian has eigenvalues with higher multiplicity the infinite-time mixing matrix has more zero eigenvalues resulting in a lower Von Neumann entropy. This is particularly interesting since higher multiplicities of the eigenvalues is associated with the presence of symmetries in the graph [12] which, in turn, have been used to characterize the entropy of the structure [11].

We can now prove the following properties for the state-aligned kernel.

Theorem 2. *The Unitary transformations minimizing the quantum Jensen Shannon divergence between pairs of density matrices in a set are transitive, i.e. let*

$$O_{i,j} = \operatorname*{argmin}_{O\in U(n)} D_{\rm JS}(\rho_i, O\rho_j O^\dagger)$$

with $i, j \in \{1, 2, 3\}$*, then*

$$D_{\mathrm{JS}}\left(\rho_1, O_{1,2}O_{2,3}\rho_3 O_{2,3}^{\dagger}O_{1,2}^{\dagger}\right) = D_{\mathrm{JS}}\left(\rho_1, O_{1,3}\rho_3 O_{1,3}^{\dagger}\right)$$

Proof. The optimal transformation between two density matrices is completely determined by the relation $O_{1,2}^{*} = \Phi_1\Phi_2^{\dagger}$ up to a change of sign of the eigenvalue and a change of base for each eignespace associated with an eigenvalue with multiplicity greater than one. In any case these changes do not affect the value of the divergence. However,

$$O_{1,2}^{*}O_{2,3}^{*} = \Phi_1\Phi_2^{\dagger}\Phi_2\Phi_3^{\dagger} = \Phi_1\Phi_3^{\dagger} = O_{1,3}^{*} \tag{13}$$

QED.

Theorem 3. *The quantum aligned QJSD kernel is positive definite.*

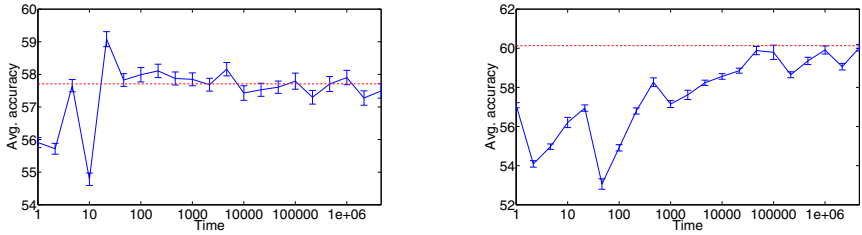
Proof. As a consequence of the previous theorems, the value of the quantum Jensen Shannon divergence of the optimally aligned density matrices is equal to the normal Jensen Shannon divergence of the sorted eigenvalues of the density matrices (taken as probability distributions). Since the Jensen Shannon divergence is proven to be negative definite [4] the state-aligned QJSD kernel, being an exponentiation of a negative definite kernel is positive definite [10]. QED.

4 Experimental Results

We now evaluate the performance of the State-Aligned (SA) QJSD kernel and we compare it with a number of well-known alternative graph kernels. More specifically, we compare our kernel with the unaligned QJSD kernel [1], the Weisfeiler-Lehman kernel [16], the graphlet kernel [17], the shortest-path kernel [3], and the random walk kernel [8]. Note that for the Weisfeiler-Lehman we set the number of iterations $h = 3$ and we attribute each node with its degree.

We run our experiments on the following datasets: 1) The **PPI** dataset, which consists of protein-protein interaction (PPIs) networks related to histidine kinase [7] (40 PPIs from *Acidovorax avenae* and 46 PPIs from *Acidobacteria*). 2) The **PTC** (The Predictive Toxicology Challenge) dataset, which records the carcinogenicity of several hundred chemical compounds for male rats (MR), female rats (FR), male mice (MM) and female mice (FM) [15] (here we use the 344 graphs in the MR class). 3) The **COIL** dataset, which consists of 5 objects from [13], each with 72 views obtained from equally spaced viewing directions, where for each view a graph was built by triangulating the extracted Harris corner points. 4) The **Reeb** dataset, which consists of a set of adjacency matrices associated to the computation of reeb graphs of 3D shapes [2].

We use a binary C-SVM to test the efficacy of the kernels. We perform 10-fold cross validation, where for each sample we independently tune the value of C, the SVM regularizer constant, by considering the training data from that sample. The process is averaged over 100 random partitions of the data, and the results are reported in terms of average accuracy ± standard error.



(a) Sub-optimal β (b) Optimal β

Fig. 1. The average classification accuracy as the time parameter of the continuous-time quantum walk varies, for an optimal (left) and sub-optimal value of the decay factor β.

Fig. 1 shows the value of the average classification accuracy ($\pm$ standard error) on the PTC dataset as we let the time parameter of the continuous-time quantum walk vary. Here the red horizontal line denotes the average accuracy for $T \to \infty$. Note that in Fig. 1(a) we set the decay parameter β of the kernel to a sub-optimal value, while in Fig. 1(b) we set it to its optimal value, i.e., the value that results in the best classification accuracy. The plot shows that when β is sub-optimal the choice of the time parameter has a clear influence on the performance of our kernel. In fact, we see that the average accuracy reaches a maximum before stabilizing around the asymptotic value. On the other hand, when β is optimized the best classification performance is achieved when $T \to \infty$. Moreover, in the latter case the average classification accuracy is higher than that recorded for smaller values of T and a sub-optimal β.

Table 1 shows the average classification accuracy ($\pm$ standard error) of the different kernels on the selected datasets. As expected, we see that the state alignment almost invariably yields an increase of the performance with respect to the standard QJSD kernel. Indeed, the localization property of the kernel that results from the quantum state alignment leads to a better discrimination, and

Table 1. Classification accuracy ($\pm$ standard error) on unattributed graph datasets. SA QJSD and QJSD denote the proposed kernel and its original unaligned version, respectively, WL is the Weisfeiler-Lehman kernel [16], GR denotes the graphlet kernel computed using all graphlets of size 3 [17], SP is the shortest-path kernel [3], and RW is the random walk kernel [8]. For each kernel and dataset, the best performing kernel is highlighted in bold.

Kernel	PPI	PTC	COIL	Reeb
SA QJSD	75.69 ± 0.85	$\mathbf{60.13 \pm 0.51}$	67.84 ± 0.15	38.50 ± 0.26
QJSD	69.12 ± 1.01	56.06 ± 0.45	$\mathbf{69.90 \pm 0.22}$	35.78 ± 0.42
WL	$\mathbf{79.40 \pm 0.96}$	56.95 ± 0.31	29.00 ± 0.57	50.53 ± 0.41
GR	51.94 ± 0.97	55.22 ± 0.19	66.46 ± 0.44	22.80 ± 0.36
SP	63.31 ± 0.80	56.51 ± 0.36	69.68 ± 0.36	$\mathbf{55.93 \pm 0.36}$
RW	50.37 ± 0.78	55.68 ± 0.14	12.18 ± 0.21	16.47 ± 0.43

Table 2. Runtime comparison on the four graph datasets

Kernel	PPI	PTC	COIL	Reeb
SA QJSD	3.68”	13.30”	33.66”	15.35”
QJSD	126.09”	35.28”	2371.17”	544”
WL	4.10”	3.79”	22.52”	11.86”
GR	2.51”	0.73”	9.25”	1.98”
SP	3.85”	0.74”	19.13”	6.15”
RW	11.58”	231”	294.24”	757.67”

thus a higher classification accuracy. Moreover, while the QJSD kernel has not been proven to be positive definite, as the quantum Jensen-Shannon divergence has only been experimentally shown to be negative definite for mixed states, our kernel is indeed positive definite, as proved in the previous Section.

Finally, Table 2 shows the runtimes of the different kernels on the four graph datasets. Note that in terms of runtime the SA QJSD kernel easily outperforms the other spectral methods, i.e., the QJSD kernel and the random walk kernel, and it is still competitive when compared with the remaining kernels.

With respect to the other kernels, the SA QJSD kernel achieves the best accuracy on the PTC dataset, and it remains competitive with the best performing ones on the PPI and COIL dataset. On the Reeb dataset, on the other hand, the shortest-path kernel and the Weisfeiler-Lehman kernel outperform our kernel and the remaining ones. Note also that the Weisfeiler-Lehman mitigates the localization problem by making use of the node labels and thus improving node localization in the evaluation of the kernel. On the other hand, our kernel does not take node attributes into account.

5 Conclusions

In this paper we have generalized a recent structural kernel based on the Jensen-Shannon divergence between quantum walks over the graph by introducing a novel alignment step which, rather than permuting the nodes of the structures, aligns the quantum states of their walks. We proved that the resulting kernel mantains the localization within the structures, but still guarantees positive definiteness. We tested our kernel against a number of alternative graph kernels and we showed its effectiveness in a number of structural classification tasks.

Acknowledgments. Edwin Hancock was supported by a Royal Society Wolfson Research Merit Award.

References

1. Bai, L., Hancock, E.R., Torsello, A., Rossi, L.: A Quantum Jensen-Shannon Graph Kernel Using the Continuous-Time Quantum Walk. In: Kropatsch, W.G., Artner, N.M., Haxhimusa, Y., Jiang, X. (eds.) GbRPR 2013. LNCS, vol. 7877, pp. 121–131. Springer, Heidelberg (2013)

2. Biasotti, S., Marini, S., Mortara, M., Patané, G., Spagnuolo, M., Falcidieno, B.: 3D shape matching through topological structures. In: Nyström, I., Sanniti di Baja, G., Svensson, S. (eds.) DGCI 2003. LNCS, vol. 2886, pp. 194–203. Springer, Heidelberg (2003)
3. Borgwardt, K.M., Kriegel, H.P.: Shortest-path kernels on graphs. In: Proc. IEEE Int. Conf. Data Mining (ICDM), pp. 74–81 (2005)
4. Briët, J., Harremoës, P.: Properties of classical and quantum jensen-shannon divergence. Physical Review A 79, 052311 (2009)
5. Fröhlich, H., Wegner, J.K., Sieker, F., Zell, A.: Optimal assignment kernels for attributed molecular graphs. In: International Conference on Machine Learning, pp. 225–232 (2005)
6. Haussler, D.: Convolution kernels on discrete structures. Technical Report UCS-CRL-99-10, Santa Cruz, CA, USA (1999)
7. Jensen, L.J., Kuhn, M., Stark, M., Chaffron, S., Creevey, C., Muller, J., Doerks, T., Julien, P., Roth, A., Simonovic, M., et al.: String 8-a global view on proteins and their functional interactions in 630 organisms. Nucleic Acids Research 37(suppl. 1), 412–416 (2009)
8. Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized kernels between labeled graphs. In: Proc. Int. Conf. Machine Learning (ICML), pp. 321–328 (2003)
9. Kempe, J.: Quantum random walks: an introductory overview. Contemporary Physics 44, 307–327 (2003)
10. Konder, R., Lafferty, J.: Diffusion kernels on graphs and other discrete input spaces. In: Proc. Int. Conf. Machine Learning (ICML), pp. 315–322 (2002)
11. Mowshowitz, A.: Entropy and the complexity of graphs: I. An index of the relative complexity of a graph. The Bulletin of Mathematical Biophysics 30(1), 175–204 (1968)
12. Mowshowitz, A.: Graphs, groups and matrices. In: Proc. 25th Summer Meeting Canad. Math. Congress, Congr. Numer., vol. 4, pp. 509–522 (1971)
13. S. A. Nene, S. K. Nayar, H. Murase, Columbia object image library (coil-20), Dept. Comput. Sci., Columbia Univ., New York, `http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php`
14. Rossi, L., Torsello, A., Hancock, E.R., Wilson, R.: Characterising graph symmetries through quantum Jensen-Shannon divergence. Physical Review E 88(3), 032806 (2013)
15. Li, G., Semerci, M., Yener, B., Zaki, M.J.: Effective graph classification based on topological and label attributes. Statistical Analysis and Data Mining 5, 265–283 (2012)
16. Shervashidze, N., Schweitzer, P., van Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-lehman graph kernels. Journal of Machine Learning Research 1, 1–48 (2010)
17. Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K., Borgwardt, K.: Efficient graphlet kernels for large graph comparison. In: Proceedings of the International Workshop on Artificial Intelligence and Statistics (2009)
18. Torsello, A.: On the state alignment of the Quantum Jensen Shannon Divergence. Technical report DAIS-2014-1, DAIS, Università Ca' Foscari Venezia (2014), `http://www.dsi.unive.it/~atorsell/DAIS-2014-1.pdf`
19. Vert, J.-P.: The optimal assignment kernel is not positive definite, arXiv:0801.4061 (2008)

Balanced K-Means for Clustering

Mikko I. Malinen and Pasi Fränti

School of Computing, University of Eastern Finland,
Box 111, FIN-80101 Joensuu, Finland
{mmali,franti}@cs.uef.fi
http://cs.uef.fi/~mmali, http://cs.uef.fi/pages/franti

Abstract. We present a k-means-based clustering algorithm, which optimizes mean square error, for given cluster sizes. A straightforward application is balanced clustering, where the sizes of each cluster are equal. In k-means assignment phase, the algorithm solves the assignment problem by Hungarian algorithm. This is a novel approach, and makes the assignment phase time complexity $O(n^3)$, which is faster than the previous $O(k^{3.5}n^{3.5})$ time linear programming used in constrained k-means. This enables clustering of bigger datasets of size over 5000 points.

Keywords: clustering, balanced clustering, assignment problem, Hungarian algorithm.

1 Introduction

Euclidean sum-of-squares clustering is an NP-hard problem [1], which groups n data points into k clusters so that intra-cluster distances are low and inter-cluster distances are high. Each group is represented by a center point (centroid). The most common criterion to optimize is the mean square error (MSE):

$$\text{MSE} = \sum_{j=1}^{k} \sum_{X_i \in C_j} \frac{|| X_i - C_j ||^2}{n}, \tag{1}$$

where X_i denotes data point locations and C_j denotes centroid locations. K-means [19] is the most commonly used clustering algorithm, which provides a local minimum of MSE given the number of clusters as input. K-means algorithm consists of two repeatedly executed steps:

Assignment Step: Assign the data points to clusters specified by the nearest centroid:

$$P_j^{(t)} = \{X_i : \|X_i - C_j^{(t)}\| \leq \|X_i - C_{j^*}^{(t)}\| \quad \forall \quad j^* = 1, ..., k\}$$

Update Step: Calculate the mean of each cluster:

$$C_j^{(t+1)} = \frac{1}{|P_j^{(t)}|} \sum_{X_i \in P_j^{(t)}} X_i$$

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 32–41, 2014.

These steps are repeated until centroid locations do not change anymore. K-means assignment step and update step are optimal with respect to MSE: The partitioning step minimizes MSE for a given set of centroids; the update step minimizes MSE for a given partitioning. The solution therefore converges to a local optimum but without guarantee of global optimality. To get better results than in k-means, slower agglomerative algorithms [10,13,12] or more complex k-means variants [3,11,21,18] are sometimes used.

In *balanced clustering* there are an equal number of points in each cluster. Balanced clustering is desirable for example in divide-and-conquer methods where the divide step is done by clustering. Examples can be found in circuit design [14] and in photo query systems [2], where the photos are clustered according to their content. Applications can also be used in workload balancing algorithms. For example, in [20] multiple traveling salesman problem clusters the cities, so that each salesman operates in one cluster. It is desirable that each salesman has equal workload. Networking utilizes balanced clustering to obtain some desirable goals [17,23].

We next review existing balanced clustering algorithms. In *frequency sensitive competitive learning* (FSCL) the centroids compete of points [5]. It multiplicatively increases the distance of the centroids to the data point by the times the centroid has already won points. Bigger clusters are therefore less likely to win more points. The method in [2] uses FSCL, but with additive bias instead of multiplicative bias. The method in [4] uses a fast ($O(kNlogN)$) algorithm for balanced clustering based on three steps: sample the given data, cluster the sampled data and populate the clusters with the data points that were not sampled. The article [6] and book chapter [9] present a *constrained k-means algorithm*, which is like k-means, but the assignment step is implemented as a linear program, in which the minimum number of points τ_h of clusters can be set as parameters. The constrained k-means clustering algorithm works as follows:

Given m points in $\mathbb{R}^n$, minimum cluster membership values $\tau_h \geq 0, h = 1, ..., k$ and cluster centers $C_1^{(t)}, C_2^{(t)}, ..., C_k^{(t)}$ at iteration t, compute $C_1^{(t+1)}, C_2^{(t+1)}$, $..., C_k^{(t+1)}$ at iteration $t+1$ using the following 2 steps:

Cluster Assignment. Let $T_{i,h}^t$ be a solution to the following linear program with $C_h^{(t)}$ fixed:

$$\text{minimize}_T \sum_{i=1}^{m} \sum_{h=1}^{k} T_{i,h} \cdot (\frac{1}{2} ||X_i - C_h^{(t)}||_2^2) \tag{2}$$

$$\text{subject to} \sum_{i=1}^{m} T_{i,h} \geq \tau_h, h = 1, ..., k \tag{3}$$

$$\sum_{h=1}^{k} T_{i,h} = 1, i = 1, ..., m \tag{4}$$

$$T_{i,h} \geq 0, i = 1, ..., m, h = 1, ..., k. \tag{5}$$

Cluster Update. Update $C_h^{(t+1)}$ as follows:

$$C_h^{(t+1)} = \begin{cases} \frac{\sum_{i=1}^{m} T_{i,h}^{(t)} X_i}{\sum_{i=1}^{m} T_{i,h}^{(t)}} & \text{if} \quad \sum_{i=1}^{m} T_{i,h}^{(t)} > 0, \\ C_h^{(t)} & \text{otherwise.} \end{cases}$$

These steps are repeated until $C_h^{(t+1)} = C_h^{(t)}, \quad \forall h = 1, ..., k$.

A cut-based method *Ratio cut* [14] includes cluster sizes in its cost function

$$\text{RatioCut}(P_1, ..., P_k) = \sum_{i=1}^{k} \frac{\text{cut}(P_i, \bar{P}_i)}{|P_i|}.$$

Here P_i:s are the partitions. *Size regularized cut* SRCut [8] is defined as the sum of the inter-cluster similarity and a regularization term measuring the relative size of two clusters. In [16] there is a balancing aiming term in cost function and [24] tries to find a partition close to the given partition, but so that cluster size constraints are fulfilled. There are also application-based solutions in networking [17], which aim at network load balancing, where clustering is done by self-organization without central control. In [23], energy-balanced routing between sensors is aimed so that most suitable balanced amount of nodes will be the members of the clusters.

Balanced clustering, in general, is a 2-objective optimization problem, in which two aims contradict each other: to minimize MSE and to balance cluster sizes. Traditional clustering aims at minimizing MSE without considering cluster size balance. Balancing, on the other hand, would be trivial if we did not care about MSE; simply by dividing points to equal size clusters randomly. For optimizing both, there are two alternative approaches: *Balance-constrained* and *balance-driven* clustering.

In balance-constrained clustering, cluster size balance is a mandatory requirement that must be met, and minimizing MSE is a secondary criterion. In balance-driven clustering, balance is an aim but not mandatory. It is a compromize between these two goals, namely the balance and the MSE. The solution can be a weighted compromize between MSE and the balance, or a heuristic that aims at minimizing MSE but indirectly creates a more balanced result than standard k-means. Existing algorithms are grouped into these two classes in Table 1.

In this paper, we formulate balanced k-means, so that it belongs to the first category. It is otherwise the same as standard k-means but it guarantees balanced cluster sizes. It is also a special case of constrained k-means, where cluster sizes are set equal. However, instead of using linear programming in the assignment phase, we formulate the partitioning as a pairing problem [7], which can be solved optimally by Hungarian algorithm in $O(n^3)$ time.

Table 1. Classification of some balanced clustering algorithms

Balance-constrained
Balanced k-means (proposed)
Constrained k-means [6]
Size constrained [24]
Balance-driven
FSCL [5]
FSCL with additive bias [2]
Cluster sampled data [4]
Ratio cut [14]
SRcut [8]
Submodular fractional programming [16]

2 Balanced k-Means

To describe balanced k-means, we need to define what is an assignment problem. The formal definition of assignment problem (or linear assignment problem) is as follows. Given two sets (A and S), of equal size, and a weight function $W : A \times S \to \mathbb{R}$. The goal is to find a bijection $f : A \to S$ so that the cost function is minimized:

$$\text{Cost} = \sum_{a \in A} W(a, f(a)).$$

In the context of the proposed algorithm, sets A and S correspond respectively to cluster slots and to data points, see Figure 1.

In balanced k-means, we proceed as in k-means, but the assignment phase is different: Instead of selecting the nearest centroids we have n pre-allocated slots (n/k slots per cluster), and datapoints can be assigned only to these slots, see Figure 1. This will force all clusters to be of same size assuming that $\lceil n/k \rceil = \lfloor n/k \rfloor = n/k$. Otherwise there will be ($n \mod k$) clusters of size $\lceil n/k \rceil$, and $k - (n \mod k)$ clusters of size $\lfloor n/k \rfloor$.

To find assignment that minimizes MSE, we solve an assignment problem using Hungarian algorithm [7]. First we construct a bipartite graph consisting n datapoints and n cluster slots, see Figure 2. We then partition the cluster slots in clusters of as even number of slots as possible.

We give centroid locations to partitioned cluster slots, one centroid to each cluster. The initial centroid locations can be drawn randomly from all data points. The edge weight is the squared distance from the point to the cluster centroid it is assigned to. Contrary to standard assignment problem with fixed weights, here the weights dynamically change after each k-means iteration according to the newly calculated centroids. After this, we perform the Hungarian algorithm to get the minimal weight pairing. The squared distances are stored in a $n \times n$ matrix, for the sake of the Hungarian algorithm. The update step is

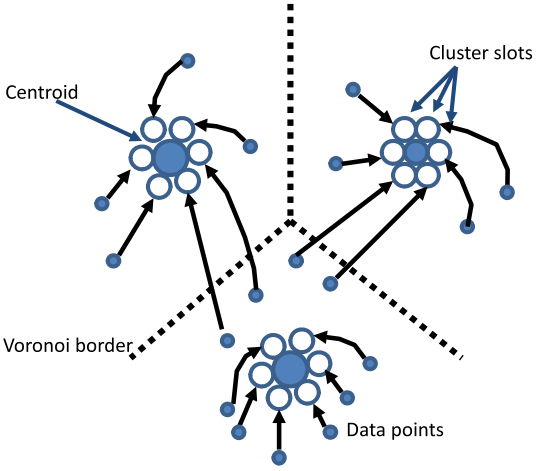


Fig. 1. Assigning points to centroids via cluster slots

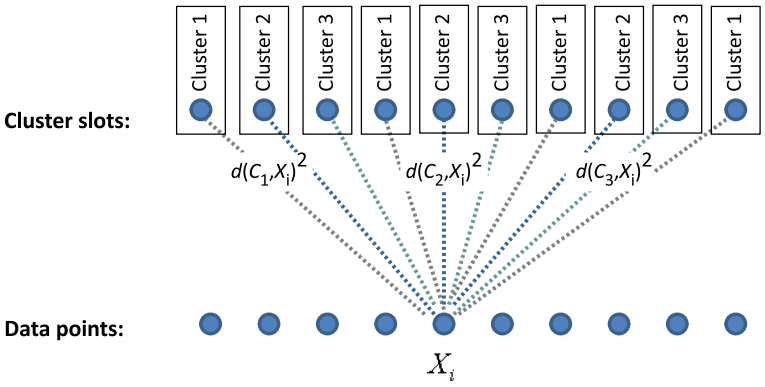


Fig. 2. Minimum MSE calculation with balanced clusters. Modeling with bipartite graph.

similar to that of k-means, where the new centroids are calculated as the means of the data points assigned to each cluster:

$$C_i^{(t+1)} = \frac{1}{n_i} \cdot \sum_{X_j \in C_i^{(t)}} X_j. \tag{6}$$

The weights of the edges are updated immediately after the update step. The pseudocode of the algorithm is in Algorithm 1. In calculation of edge weights, the number of cluster slot is denoted by a and mod is used in calculation of cluster where a cluster slots belongs to. The edge weights are calculated by

$$W(a,i) = dist(X_i, C^t_{(a \bmod k)+1})^2 \quad \forall a \in [1,n] \quad \forall i \in [1,n]. \tag{7}$$

Algorithm 1. Balanced k-means
Input: dataset X, number of clusters k
Output: partitioning of dataset.

Initialize centroid locations C^0.
$t \leftarrow 0$
repeat
  Assignment step:
    Calculate edge weights.
    Solve an Assignment problem.
  Update step:
    Calculate new centroid locations C^{t+1}
  $t \leftarrow t + 1$
until centroid locations do not change.
Output partitioning.

After convergence of the algorithm the partition of points X_i, $i \in [1, n]$, is

$$X_{f(a)} \in P_{(a \bmod k)+1}. \tag{8}$$

There is a convergence result in [6] (Proposition 2.3) for constrained k-means. The result says that the algorithm terminates in a finite number of iterations at a partitioning that is locally optimal. At each iteration, the cluster assignment step cannot increase the objective function of constrained k-means (3) in [6]. The cluster update step will either strictly decrease the value of the objective function or the algorithm will terminate. Since there are a finite number of ways to assign m points to k clusters so that cluster h has at least τ_h points, since constrained k-means algorithm does not permit repeated assignments, and since the objective of constrained k-means (3) in [6] is strictly nonincreasing and bounded below by zero, the algorithm must terminate at some cluster assignment that is locally optimal. The same convergence result applies to balanced k-means as well. The assignment step is optimal with respect to MSE because of pairing and the update step is optimal, because MSE is clusterwise minimized as is in k-means.

3 Time Complexity

Time complexity of the assignment step in k-means is $O(k \cdot n)$. Constrained k-means involves linear programming. It takes $O(v^{3.5})$ time, where v is the number of variables, by Karmarkars projective algorithm [15,22], which is the fastest interior point algorithm known to the authors. Since $v = k \cdot n$, the time complexity is $O(k^{3.5}n^{3.5})$. The assignment step of the proposed balanced k-means algorithm can be solved in $O(n^3)$ time with the Hungarian algorithm. This makes it much faster than in the constrained k-means, and allows therefore significantly bigger datasets to be clustered.

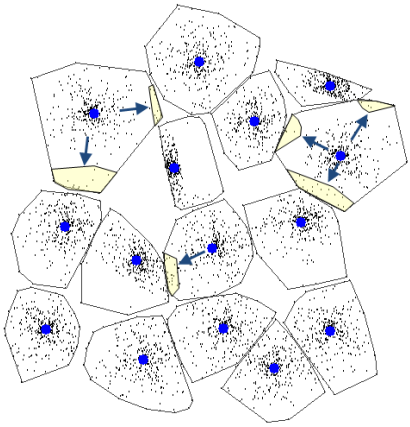


Fig. 3. Sample clustering result. Most significant differences between balanced clustering and standard k-means (non-balanced) clustering are marked and pointed out by arrows.

Table 2. MSE, standard deviation of MSE and time/run of 100 runs

Dataset	Size	Clusters	Algorithm	Best	Mean	St.dev.	Time
s2	5000	15	Balanced k-means	**2.86**	(one run)	(one run)	**1h 40min**
			Constrained k-means	-	-	-	-
s1	1000	15	Balanced k-means	2.89	(one run)	(one run)	**47s**
subset			Constrained k-means	**2.61**	(one run)	(one run)	26min
s1	500	15	Balanced k-means	3.48	3.73	0.21	**8s**
subset			Constrained k-means	**3.34**	**3.36**	0.16	30s
			K-means	2.54	4.21	1.19	0.01s
s1	500	7	Balanced k-means	14.2	15.7	1.7	10s
subset			Constrained k-means	**14.1**	**15.6**	1.6	**8s**
s2	500	15	Balanced k-means	3.60	3.77	0.12	**8s**
subset			Constrained k-means	**3.42**	**3.43**	0.08	29s
s3	500	15	Balanced k-means	3.60	3.69	0.17	**9s**
subset			Constrained k-means	**3.55**	**3.57**	0.12	35s
s4	500	15	Balanced k-means	3.46	3.61	1.68	**12s**
subset			Constrained k-means	**3.42**	**3.53**	0.20	45s
thyroid	215	2	Balanced k-means	**4.00**	**4.00**	0.001	2.5s
			Constrained k-means	**4.00**	**4.00**	0.001	**0.25s**
wine	178	3	Balanced k-means	**3.31**	3.33	0.031	0.36s
			Constrained k-means	**3.31**	**3.31**	0.000	**0.12s**
iris	150	3	Balanced k-means	**9.35**	3.39	0.43	0.34s
			Constrained k-means	**9.35**	**3.35**	0.001	**0.14s**

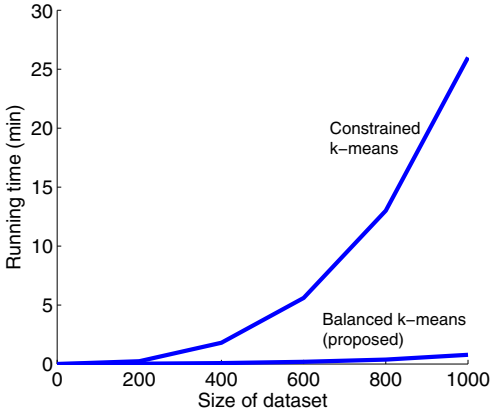


Fig. 4. Running time with different-sized subsets of s1 dataset

4 Experiments

In the experiments we use artificial datasets s1-s4, which have Gaussian clusters with increasing overlap and real-world datasets thyroid, wine and iris. The source of the datasets is `http://cs.uef.fi/sipu/datasets/`. As a platform, Intel Core i5-3470 3.20GHz processor was used. We have been able to cluster datasets of size 5000 points. One example partitioning can be seen in Figure 3, for which the running time was 1h40min. Comparison of MSE values of constrained k-means and balanced k-means is shown in Table 2, running times in Figure 4. The results indicate that constrained k-means gives slightly better MSE in many cases, but that balanced k-means is significantly faster when the size of dataset increases. For dataset of size 5000 constrained k-means could no longer provide result within one day. The difference in MSE is most likely due to the fact that balanced k-means strictly forces balance within ± 1 points, but constrained k-means does not. It may happen, that constrained k-means has many clusters of size $\lfloor n/k \rfloor$, but some smaller amount of clusters of size bigger than $\lceil n/k \rceil$.

5 Conclusions

We have presented balanced k-means clustering algorithm which guarantees equal-sized clusters. The algorithm is a special case of constrained k-means, where cluster sizes are equal, but much faster. The experimental results show that the balanced k-means gives slightly higher MSE-values to that of the constrained k-means, but about 3 times faster already for small datasets. Balanced k-means is able to cluster bigger datasets than constrained k-means. However, even the proposed method may still be too slow for practical application and therefore, our future work will focus on finding some faster sub-optimal algorithm for the assignment step.

References

1. Aloise, D., Deshpande, A., Hansen, P., Popat, P.: NP-hardness of Euclidean sum-of-squares clustering. Mach. Learn. 75, 245–248 (2009)
2. Althoff, C.T., Ulges, A., Dengel, A.: Balanced clustering for content-based image browsing. In: GI-Informatiktage 2011. Gesellschaft für Informatik e.V. (March 2011)
3. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: SODA 2007: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035. Society for Industrial and Applied Mathematics, Philadelphia (2007)
4. Banerjee, A., Ghosh, J.: On scaling up balanced clustering algorithms. In: Proceedings of the SIAM International Conference on Data Mining, pp. 333–349 (2002)
5. Banerjee, A., Ghosh, J.: Frequency sensitive competitive learning for balanced clustering on high-dimensional hyperspheres. IEEE Transactions on Neural Networks 15, 719 (2004)
6. Bradley, P.S., Bennett, K.P., Demiriz, A.: Constrained k-means clustering. Tech. rep., MSR-TR-2000-65, Microsoft Research (2000)
7. Burkhard, R., Dell'Amico, M., Martello, S.: Assignment Problems (Revised reprint). SIAM (2012)
8. Chen, Y., Zhang, Y., Ji, X.: Size regularized cut for data clustering. In: Advances in Neural Information Processing Systems (2005)
9. Demiriz, A., Bennett, K.P., Bradley, P.S.: Using assignment constraints to avoid empty clusters in k-means clustering. In: Basu, S., Davidson, I., Wagstaff, K. (eds.) Constrained Clustering: Advances in Algorithms, Theory, and Applications. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series (2008)
10. Equitz, W.H.: A New Vector Quantization Clustering Algorithm. IEEE Trans. Acoust., Speech, Signal Processing 37, 1568–1575 (1989)
11. Fränti, P., Kivijärvi, J.: Randomized local search algorithm for the clustering problem. Pattern Anal. Appl. 3(4), 358–369 (2000)
12. Fränti, P., Virmajoki, O.: Iterative shrinking method for clustering problems. Pattern Recognition 39(5), 761–765 (2006)
13. Fränti, P., Virmajoki, O., Hautamäki, V.: Fast agglomerative clustering using a k-nearest neighbor graph. IEEE Trans. on Pattern Analysis and Machine Intelligence 28(11), 1875–1881 (2006)
14. Hagen, L., Kahng, A.B.: New spectral methods for ratio cut partitioning and clustering. IEEE Transactions on Computer-Aided Design 11(9), 1074–1085 (1992)
15. Karmarkar, N.: A new polynomial time algorithm for linear programming. Combinatorica 4(4), 373–395 (1984)
16. Kawahara, Y., Nagano, K., Okamoto, Y.: Submodular fractional programming for balanced clustering. Pattern Recognition Letters 32(2), 235–243 (2011)
17. Liao, Y., Qi, H., Li, W.: Load-Balanced Clustering Algorithm With Distributed Self-Organization for Wireless Sensor Networks. IEEE Sensors Journal 13(5), 1498–1506 (2013)
18. Likas, A., Vlassis, N., Verbeek, J.: The global k-means clustering algorithm. Pattern Recognition 36, 451–461 (2003)
19. MacQueen, J.: Some methods of classification and analysis of multivariate observations. In: Proc. 5th Berkeley Symp. Mathemat. Statist. Probability, vol. 1, pp. 281–296 (1967)

20. Nallusamy, R., Duraiswamy, K., Dhanalaksmi, R., Parthiban, P.: Optimization of non-linear multiple traveling salesman problem using k-means clustering, shrink wrap algorithm and meta-heuristics. International Journal of Nonlinear Science 9(2), 171–177 (2010)
21. Pelleg, D., Moore, A.: X-means: Extending k-means with efficient estimation of the number of clusters. In: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 727–734. Morgan Kaufmann, San Francisco (2000)
22. Strang, G.: Karmarkars algorithm and its place in applied mathematics. The Mathematical Intelligencer 9(2), 4–10 (1987)
23. Yao, L., Cui, X., Wang, M.: An energy-balanced clustering routing algorithm for wireless sensor networks. In: 2009 WRI World Congress on Computer Science and Information Engineering, vol. 3. IEEE (2006)
24. Zhu, S., Wang, D., Li, T.: Data clustering with size constraints. Knowledge-Based Systems 23(8), 883–889 (2010)

Poisoning Complete-Linkage Hierarchical Clustering

Battista Biggio[1], Samuel Rota Bulò[2], Ignazio Pillai[1], Michele Mura[1], Eyasu Zemene Mequanint[1], Marcello Pelillo[3], and Fabio Roli[1]

[1] University of Cagliari, Italy
[2] FBK-irst, Trento, Italy
[3] Ca' Foscari University, Venice, Italy

Abstract. Clustering algorithms are largely adopted in security applications as a vehicle to detect malicious activities, although few attention has been paid on preventing deliberate attacks from subverting the clustering process itself. Recent work has introduced a methodology for the security analysis of data clustering in adversarial settings, aimed to identify potential attacks against clustering algorithms and to evaluate their impact. The authors have shown that single-linkage hierarchical clustering can be severely affected by the presence of a very small fraction of carefully-crafted poisoning attacks into the input data, highlighting that the clustering algorithm may be itself the weakest link in a security system. In this paper, we extend this analysis to the case of complete-linkage hierarchical clustering by devising an ad hoc poisoning attack. We verify its effectiveness on artificial data and on application examples related to the clustering of malware and handwritten digits.

1 Introduction

Clustering algorithms play an important role in data analysis by allowing us to gain insight into large sets of unlabeled data. Recently, clustering algorithms have been adopted in the context of computer security to solve different problems, *e.g.*, spotting compromised domains in DNS traffic [1], gathering information about tools and sources of attacks against Internet websites [2], detecting malicious software (*i.e.*, malware) such as computer viruses or worms [3, 4], and even identifying repackaged Android applications and Android mobile malware [5, 6].

The collection of data for most of the aforementioned scenarios is carried out in an unsupervised way; *e.g.*, malware samples such as files infected by computer viruses are often gathered from the Internet using honeypots (*i.e.*, machines that purposely expose known vulnerabilities to be infected by malware [7]), or other ad-hoc services, like VirusTotal.[1] Accordingly, the clustering algorithms used to analyze such data are exposed to possible attacks. Indeed, carefully-crafted samples might be injected into the collected data to subvert the clustering process and prevent the system from gaining useful knowledge. Due to these intrinsically adversarial scenarios, evaluating the *security* of clustering algorithms against

[1] `http://virustotal.com`

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 42–52, 2014.

carefully-designed attacks and proposing suitable countermeasures has become an important issue.

In the literature, the problem of learning in adversarial environments has been mainly addressed in the area of supervised classification [8–11], and regression [12]. Instead, only *few* works have addressed the issue of security evaluation (and the design of countermeasures) of unsupervised learning approaches such as clustering algorithms. The pioneering work in [13, 14] has focused the attention on the problem of devising specific attacks to subvert the clustering process. They showed how points could be easily *hidden* within an existing cluster by forming a *fringe* cluster, *i.e.*, by placing such points sufficiently close the border of an existing cluster. They further designed attacks consisting in adding points in a way to *bridge* two clusters, *i.e.*, by inducing the fusion of two clusters. A step further has been taken in [15], where the authors considered several potential attack scenarios in a more systematic manner. Indeed, they introduced a model of the attacker that allows one to make specific assumptions on the adversary's goal, knowledge of the attacked system, and capability of manipulating the input data, and to subsequently formalize a corresponding *optimal* attack strategy.

In this paper, we extend the clustering security analysis proposed in [15], which was focused on single-linkage hierarchical clustering, to the case of complete linkage by devising an ad hoc poisoning attack. The reason is that single- and complete-linkage hierarchical clustering algorithms are among the most used ones for the purpose of malware detection and classification [3, 4]. To cope with the computational problem of determining the optimal attack strategy, we propose some heuristics that allow us to find good approximate solutions. We finally verify the effectiveness of our approach on artificial data and on application examples related to the clustering of malware and handwritten digits.

2 Clustering in Adversarial Settings

We review here the framework proposed in [15], which introduces an adversary's model that can be used to identify and devise attacks against clustering algorithms. The adversary's model comprises the definition of the adversary's goal, knowledge of the attacked system, and capability of manipulating the input data.

Adversary's Goal. The adversary's goal is defined based on the attack specificity and the security violation pursued by the adversary. The attack specificity can be *targeted*, if it involves only the clustering of a given subset of samples; or *indiscriminate*, if it potentially affects the clustering of any sample. The security violations jeopardize the *integrity* of a system, its *availability*, or the *privacy* of its users. The *availability violations* are targeted at compromising the functionality of the system, thus causing a denial of service. In a supervised setting this amounts to achieving the largest possible classification error [10, 8, 16], while in the unsupervised case it entails attacks that induce a significant perturbation in the clustering result. The *integrity violations* aim at pursuing some specific malicious activities without significantly compromising the normal system operation. In the supervised learning setting [10, 11], these attacks camouflage

some malicious samples (*e.g.*, spam emails) to evade detection, without affecting the classification of legitimate samples. In the unsupervised setting, instead, integrity violations are attacks aiming at deflecting the grouping for specific samples, while limiting the changes to the original clustering. As an example, an attacker might change some samples in a way to hide them in a different cluster, without excessively altering the initial clusters. Finally, the *privacy violations* try to obtain information about the system's users from the clustered data.

Adversary's Knowledge. The knowledge that the adversary has about the system can be divided into: (i) *knowledge about the data*, *i.e.* the adversary knows the dataset or a surrogate set sampled from the same distribution; (ii) *knowledge of the feature space*, *i.e.* the adversary knows how features are extracted for each sample; (iii) *knowledge about the algorithm*, *i.e.* the adversary knows the clustering algorithm and thus how data is organized into clusters; (iv) *knowledge about the algorithm's parameters*, *i.e.* the adversary knows the parameters used to run the clustering algorithm. The scenario where the adversary has all the aforementioned types of knowledge is referred to as the *perfect knowledge* case.

Adversary's Capability. The adversary's capability defines in which way and to what extent the attacker can influence the clustering process. In practice, it imposes some limits to the power of the attacker. In the supervised case [10, 8], an attacker can exercise an influence on the training data and test data (*a.k.a. causative influence*) or on the test data only (*a.k.a. exploratory influence*). In the unsupervised case, instead, there is no distinction between training and test set, so the adversary can exercise only a causative influence by manipulating the samples to be clustered. The capabilities of the adversary can be circumscribed by imposing a maximum number of samples that can be manipulated, *e.g.* in the case of malware collected through *honeypots* [7] the adversary might easily send few samples without having access to the rest of the data. An additional constraint consists in limiting the extent of the modifications that the attacker can do to a sample in order to preserve its malicious functionality. Indeed, malicious samples like spam emails or malware code may not be manipulated in an unconstrained manner. Such a constraint can be expressed in terms of a suitable distance measure between the original, non-manipulated attack samples and the manipulated ones, as in [8, 17, 10, 11].

3 Poisoning Attacks

Once the adversary's model has been defined, one can design an *optimal* attack strategy that specifies how data should be manipulated to meet the adversary's goal, given the restrictions imposed by her knowledge and capabilities. In this section, we focus on *poisoning* attacks, *i.e.*, attacks targeted at violating the system's availability by indiscriminately corrupting the cluster assignment of any data point through the insertion of well-crafted *poisoning* samples in the input data. We additionally take the worst-case perspective in which the adversary has *perfect knowledge*. In more formal terms, following [15], the adversary's goal

is to maximize a distance measure between the clustering $\mathcal{C}$ obtained from the original, unchanged dataset $\mathcal{D}$ and the clustering obtained by the application of the clustering algorithm on the contaminated dataset $\mathcal{D}'$. We assume $\mathcal{D}'$ to be the union of $\mathcal{D}$ with a set of (poisoning) *attack samples* $\mathcal{A}'$, *i.e.* $\mathcal{D}' = \mathcal{D} \cup \mathcal{A}'$. Accordingly, the objective function for the adversary can be written as

$$g(\mathcal{A}') = d_c(\mathcal{C}, f_{\mathcal{D}}(\mathcal{D} \cup \mathcal{A}'))\,, \tag{1}$$

where d_c is a distance measure between clusterings, and $f_{\mathcal{D}}(\mathcal{D}')$ denotes the output of the clustering algorithm f run on the data $\mathcal{D}'$, but restricted to the samples in $\mathcal{D}$, since we are interested in measuring the clustering corruption with respect only to the original data samples. The capability of the adversary is circumscribed by imposing a maximum number of m attack samples, *i.e.* $|\mathcal{A}'| \leq \mathsf{m}$, and by imposing a box constraint on the feature values of the attack samples, *i.e.* $\boldsymbol{x}_{\mathrm{lb}} \leq \boldsymbol{a} \leq \boldsymbol{x}_{\mathrm{ub}}$ for all $\boldsymbol{a} \in \mathcal{A}'$, where $\boldsymbol{a} \leq \boldsymbol{b}$ means that the inequality holds for all vector components. In other terms, the set of attack samples $\mathcal{A}'$ is an element of $\Omega_{\mathsf{p}} = \{\{\boldsymbol{a}'_i\}_{i=1}^{\mathsf{m}} : \boldsymbol{x}_{\mathrm{lb}} \leq \boldsymbol{a}'_i \leq \boldsymbol{x}_{\mathrm{ub}} \text{ for } i = 1, \ldots, \mathsf{m}\}$. To summarize, the optimal poisoning attack strategy with perfect knowledge under the aforementioned capabilities is the solution of the following optimization problem:

$$\text{maximize}\, g(\mathcal{A}') \quad \text{s.t.}\, \mathcal{A}' \in \Omega_{\mathrm{p}}\,. \tag{2}$$

4 Poisoning Complete-Linkage Hierarchical Clustering

In this section, we focus on solving the optimization problem given by Eq. (2) for the *complete-linkage* hierarchical clustering algorithm.

Before delving into the details of our derivation, it is worth remarking here that hierarchical clustering algorithms do not output a given partitioning of the data into a set of clusters directly. They rather produce a *hierarchy* of clusterings by carrying out an *agglomerative* bottom-up procedure [18]: each point is initially considered as a separate cluster; then, at each iteration, the two *closest* clusters are fused according to a given distance measure between clusters (*i.e.*, the so-called *linkage* criterion), until a single cluster (containing all data points) is obtained. This procedure can be represented as a tree-like data structure called *dendrogram*. To obtain a given partitioning of the data, the dendrogram has to be *cut* at a certain height. Points that remain interconnected after the cut will be considered part of the same cluster. Depending on the linkage criterion, several variants of hierarchical clustering have been defined; in particular, for the *complete-linkage* and the *single-linkage* clustering algorithms the distance between any two clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ is respectively defined as the maximum and minimum Euclidean distance between all pairs of samples in $\mathcal{C}_1 \times \mathcal{C}_2$.

We denote sample-to-cluster assignments as a binary matrix $\mathtt{Y} \in \{0,1\}^{\mathsf{n}\times\mathsf{k}}$ where $\mathtt{Y}_{ik} = 1$ if the i^{th} sample is assigned to the k^{th} cluster. We also define the distance measure d_c between clusterings used in (1) as $d_c(\mathtt{Y}, \mathtt{Y}') = \|\mathtt{Y}\mathtt{Y}^\top - \mathtt{Y}'\mathtt{Y}'^\top\|_F$, where $\|\cdot\|_F$ is the Frobenius norm. This distance counts the number of times two samples have been clustered together in one clustering and not in the other,

and viceversa. It is also known as the Mirkin metric, and its relationships with other clustering indices can be found here [19]. To employ this distance measure with hierarchical clustering, we have to specify an appropriate *dendrogram cut* criterion. Similarly to [15], we take the worst-case scenario for the adversary in order to obtain the minimum performance degradation incurred by the clustering algorithm under attack. This translates into selecting the dendrogram cut minimizing the distance d_c between the uncontaminated clustering result $\mathcal{C}$ and the one induced by the cut.

The optimization problem in (2) is hard to solve due to the presence of function $f_{\mathcal{D}}$, which has in general no analytic form and a discrete output. For this reason, we propose in the following some greedy optimization strategies aimed at finding a local maximum of the objective function, by adding one attack sample at a time, *i.e.*, $|\mathcal{A}'| = \mathsf{m} = 1$.

Poisoning Attack. The idea of the optimization heuristic is to generate a set $\mathcal{S}$ of $2k$ candidate attack samples that could significantly compromise the clustering result, and retain as attack sample the one yielding the highest value of the objective function. These candidate samples are determined in a way to potentially induce a cluster to be split and to possibly induce one of those parts to be merged to another cluster. To populate the set $\mathcal{S}$, we determine for each cluster $\mathcal{V} \in \mathcal{C}$ a pair of points $(\boldsymbol{x}_1, \boldsymbol{x}_2) \in \mathcal{V} \times \mathcal{V}$ with maximum distance within the cluster (*a.k.a.* cluster diameter), *i.e.* $(\boldsymbol{x}_1, \boldsymbol{x}_2) \in \arg\max_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{V} \times \mathcal{V}} ||\boldsymbol{x} - \boldsymbol{y}||$. Now, to induce the cluster $\mathcal{V}$ to be split we aim at increasing the diameter of $\mathcal{V}$. To this end, we determine two points $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ that are collinear with $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, but outside the segment joining them, *i.e.* $\boldsymbol{z}_1 = \boldsymbol{x}_1 + \frac{1}{2}\alpha_1 \boldsymbol{l}$ and $\boldsymbol{z}_2 = \boldsymbol{x}_2 - \frac{1}{2}\alpha_2 \boldsymbol{l}$ where $\boldsymbol{l} = (\boldsymbol{x}_1 - \boldsymbol{x}_2)/\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$ and $\alpha_{1,2} > 0$. The parameters $\alpha_{1,2}$ are determined as the minimum between the cluster diameter of $\mathcal{V}$ and the distance of $\boldsymbol{x}_{1,2}$ to the closest sample not in $\mathcal{V}$, *i.e.* $\alpha_{1,2} = \min_{\boldsymbol{z} \in (\mathcal{D} \setminus \mathcal{V}) \cup \{\boldsymbol{x}_{2,1}\}} \|\boldsymbol{x}_{1,2} - \boldsymbol{z}\|$. By computing $\boldsymbol{z}_{1,2}$ for each cluster, we obtain the set of candidate attack samples $\mathcal{S}$ (see Fig. 1).

We can now select the attack sample in $\boldsymbol{z} \in \mathcal{S}$ maximizing $g(\{\boldsymbol{z}\})$ and take $\mathcal{A}' = \{\boldsymbol{z}\}$ as the optimal (in an heuristic sense) attack strategy. We call this strategy *Extend (Best)*. For the sake of computational efficiency, we also experiment a strategy that approximates the clustering matrix $\mathtt{Y}'$ directly, without explicitly running the clustering algorithm for each candidate attack point. Specifically, given a candidate attack sample $\boldsymbol{z} \in \mathcal{S}$, we split its cluster $\mathcal{V}$ in two parts, one part containing the $|\mathcal{V}|/2$ closest points to $\boldsymbol{z}$ in $\mathcal{V}$. The newly constructed cluster containing $\boldsymbol{z}$ will be merged with another cluster $\mathcal{W} \in \mathcal{C}$, if the distance between $\boldsymbol{z}$ and $\mathcal{W}$ is smaller than the diameter of $\mathcal{V}$. Given this new clustering represented with the matrix $\mathtt{Y}'$, we can evaluate the objective value of $\boldsymbol{z}$ and again retain the one with the best value among the ones in $\mathcal{S}$. We call this strategy *Extend (Hard)*. Finally, we also experiment the computation of a soft version of the clustering matrix $\mathtt{Y}'$, where the element $\mathtt{Y}_{ki}$ holds the posterior probability of class k given sample $\boldsymbol{x}_i$ computed with the Bayes rule using a likelihood estimated with a Gaussian kernel density estimator (as done in [15]). The soft matrix $\mathcal{Y}'$ is

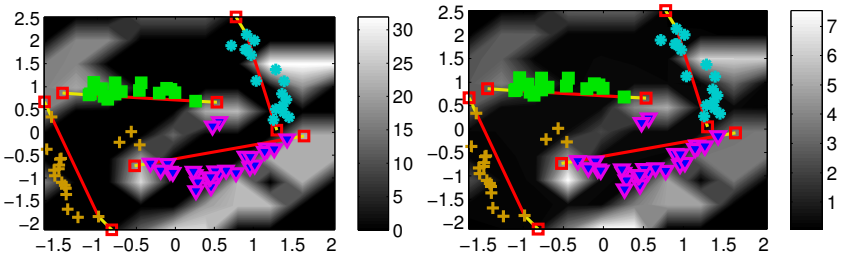


Fig. 1. Poisoning complete-linkage hierarchical clustering. In each plot, 100 samples grouped into 4 clusters are represented with different markers and colors. The red segments highlight the cluster's diameter, while the red squares are the candidate attack samples in $\mathcal{S}$. The objective function of (2), shown in the background for each greedy attack location ($|\mathcal{A}'| = 1$) is computed with hard (left plot) and soft assignments (right plot). Note how the set $\mathcal{S}$ of candidate attack points includes local maxima of the objective function.

computed for each candidate attack sample $z \in \mathcal{S}$ and the objective value in (2) evaluated. Finally, the sample with the best objective is retained. We call this last strategy *Extend (Soft)*.

5 Experiments

Following the experimental setting in [15], in this section we report an empirical evaluation of the effectiveness of our heuristic poisoning attacks on an artificial two-dimensional dataset, a realistic application example on malware clustering, and a high-dimensional problem involving clustering of handwritten digits. In each experiment, the proposed attacks are compared against the following two strategies: *Random*, that chooses the attack point at random from the minimum box enclosing the data; and *Random (Best)*, that randomly selects $2k$ attack points from the same enclosing box (being k the number of clusters), and retains the one that maximizes the objective function. The rationale is to compare our methods against random-based strategies that exhibit similar computational complexities: *Extend (Best)* and *Random (Best)* require re-running the clustering algorithm $2k$ times at each iteration to evaluate the objective and select the best candidate attack sample, while *Extend (Hard)*, *Extend (Soft)* and *Random* select the attack point without re-running the clustering algorithm.

As for evaluating the attack effectiveness, we report the value of the objective function and the number of clusters obtained at each iteration, and the two measures *Split* and *Merge* defined in [15] as follows. Let $\mathcal{C}$ and $\mathcal{C}'$ be the initial and the final clustering of the samples in $\mathcal{D}$, and $\mathtt{C}$ a binary matrix whose elements $\mathtt{C}_{kk'}$ indicate the co-occurrence of at least one sample in the kth cluster of $\mathcal{C}$ and in the k'th cluster of C', then:

$$\text{Split} = \operatorname*{mean}_i \sum_j \mathtt{C}_{ij} \;,\; \text{Merge} = \operatorname*{mean}_j \sum_i \mathtt{C}_{ij} \;. \tag{3}$$

The rationale is that *Split* evaluates the extent to which the initial clusters are split across distinct final clusters, while *merge* evaluates to what extent the final clusters include points originally belonging to distinct initial clusters.

Artificial Data. We consider here the standard two-dimensional banana-shaped dataset from PRTools.[2] A particular instance of this data is shown in Fig. 1. The number of clusters is set to $k = 4$, which corresponds to the ideal, untainted clustering $\mathcal{C}$ considered in this case. The experiment is repeated five times, each time by randomly sampling 80 data points, and adding up to 20 attack samples (*i.e.*, 20% of the data). As described in Sect. 4, the attack proceeds greedily by adding one sample at a time. After adding each attack sample, we allow the clustering algorithm to change the number of clusters from 2 to 50. The criterion used to determine the number of clusters is to minimize the distance of the current partitioning with the clustering in the absence of attack, as explained in Sect. 4. *Extend (Soft)* estimates soft clustering assignments Υ' using a Gaussian KDE, whose kernel bandwidth h is set as the average distance between each pair of data points, yielding $h \approx 2$ in each run. On this dataset, we also compare our heuristic attacks with a computationally-expensive greedy attack that selects the best attack point at each iteration (by re-running the clustering algorithm and evaluating the objective) from an equally-spaced grid of 50×50 points in $[-2.5, 2.5] \times [-2.5, 2.5]$. This attack can be thus retained very close to the optimal greedy attack, and we refer to it as *Optimal (Grid Search)*.

Results are reported in Fig. 2 (first column). From the top plot, one may note how *Extend (Best)* is able to achieve very close values of the objective function to those attained by *Optimal (Grid Search)*, denoting the effectiveness of the considered candidate attack points. *Random (Best)* performs only slightly worse, in this case, due to the fact that the feature space is only two-dimensional and bounded, and that the local maxima of the objective function typically cover large areas (see, *e.g.*, Fig. 1). *Extend (Hard)* and *Extend (Soft)* perform similarly to *Random (Best)*, and better than *Random*, confirming to some extent that predicting the output of the complete-linkage clustering algorithm after the addition of a given data point may not always be as trivial as assumed by our heuristics. The bottom plot shows how the number of selected clusters vary as the attack progresses. The main effect is an oscillation of the number of clusters, highlighting that initial clusters are fragmented, and that the resulting fragments are then merged to form distinct clusters. This effect is also confirmed by the *Split* and *Merge* values reported in Table 1.

Malware Clustering. We focus here on a real-world application example involving the behavioral malware clustering algorithm proposed in [3]. Its goal is to obtain malware clusters that can be used to automatically generate network signatures that can in turn spot botnet command-and-control (C&C) and other malware-related communications at the network perimeter. With respect to the original algorithm, we made the same simplifications done in [15], and use the

[2] `http://prtools.org`

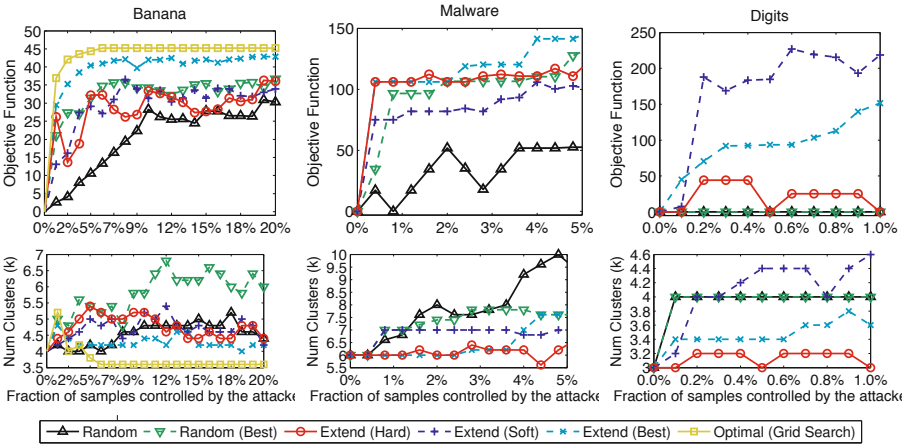


Fig. 2. Results for the Banana-shaped (first column), the Malware (second column), and the Digit (third column) datasets. Plots in the top and the bottom row respectively show how the objective function $d_c(\mathtt{Y}, \mathtt{Y}')$ and the number of clusters vary as the fraction of attack samples increases.

Table 1. Split and Merge averaged values and standard deviations for the Banana-shaped dataset (at 20% poisoning), the Malware dataset (at 5% poisoning), and the Digit dataset (at 1% poisoning)

	Banana (20%)		Malware (5%)		Digits (1%)	
	Split	*Merge*	*Split*	*Merge*	*Split*	*Merge*
Random	1.70 ± 0.27	1.56 ± 0.31	1.10 ± 0.09	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Random (Best)	2.20 ± 0.32	1.52 ± 0.31	1.33 ± 0.12	1.31 ± 0.39	1.00 ± 0.00	1.00 ± 0.00
Extend (Hard)	2.10 ± 0.13	1.97 ± 0.41	1.33 ± 0.12	1.15 ± 0.08	1.00 ± 0.00	1.13 ± 0.18
Extend (Soft)	1.90 ± 0.38	1.81 ± 0.18	1.27 ± 0.09	1.11 ± 0.17	1.60 ± 0.15	1.07 ± 0.15
Extend (Best)	2.15 ± 0.22	2.05 ± 0.11	1.83 ± 0.00	1.36 ± 0.14	1.27 ± 0.28	1.07 ± 0.15
Optimal	2.00 ± 0.31	2.28 ± 0.41				

complete-linkage criterion instead of the single linkage. Each malware is represented by six feature values, normalized in $[0, 1]$: (i) number of GET requests; (ii) number of POST requests; (iii) average URL length; (iv) average number of parameters in the request; (v) average amount of data sent by POST requests; and (vi) average response length. We use a dataset of 1,000 malware samples chosen from *Dataset 1* of [3], which includes malware collected from different sources, like MWCollect[3] and Malfease[4]. The experiments are repeated five times, by randomly selecting a subset of 475 malware samples in each run. As in [3, 15], the initial set of clusters $\mathcal{C}$ is selected as the partitioning that minimizes the value of the Davies-Bouldin Index [20], yielding approximately 9 clusters in each run. During the attack, a maximum of 25 attack samples are added (*i.e.*, 5% of the data), while the clustering algorithm can vary the number of clusters from

[3] Collaborative Malware Collection and Sensing, `https://alliance.mwcollect.org`.
[4] Project Malfease, `http://malfease.oarci.net`.

2 to 50. The value of h for the KDE used in *Extend (Soft)* is set to 0.2, as it is close to the average distance between each pair of samples in each run.

Results are shown in Fig. 2 (second column). The effect of the attack is essentially the same as for the Banana-shaped data, as witnessed by the variation in the number of clusters (bottom plot) and from the values of *Split* and *Merge* in Table 1: the initial clusters are fragmented to form different clusters. *Extend (Best)* outperforms again the other methods, as shown by the values of the objective function (top plot). *Extend (Soft)* performs slightly worse than *Random (Best)*, instead, while *Random* tends to increase the number of clusters but not the objective function. This happens since most of the attack points are clustered separately in new, additional clusters without affecting the initial clusters at all.

Handwritten Digits. We finally consider clustering of handwritten digits from the MNIST dataset [21],[5] where each digit is represented as a grayscale image of 28×28 pixels. Each pixel is considered here as a feature value (normalized in $[0, 1]$ by dividing its value by 255). The feature space has thus 784 dimensions. For simplicity, as in [15], we restrict our analysis on a subset of the data made up of the three digits '0', '1', and '6'. Three initial clusters, each representing one of the considered digits, are obtained by first computing the average digit for each class, and then selecting 700 samples per class, by retaining the closest samples to the corresponding average digit. We run the experiments five times, each time by randomly choosing 330 samples per digit from the corresponding set of 700 pre-selected samples. While the attack proceeds, the attacker can inject up to 10 attack samples (1% of the data), while the clustering algorithm can select from 2 to 100 clusters. The value of h for the KDE used in *Extend (Soft)* is set as $h \approx 1$, based on the average distance between all pairs of samples in each run.

Results are shown in Fig. 2 (third column). Notably, in this case *Extend (Soft)* outperforms the other methods, including *Extend (Best)*. *Extend (Soft)* deals indeed with less sharp variations of the objective function, that may in turn allow it to identify a better combination of attack points in the end. The random-based methods are completely ineffective, instead. This is due to the high dimensionality of the feature space, which drastically reduces the chances of finding a good local maxima by selecting the attack points at random. In particular, it can be noted from the bottom plot in Fig. 2 (third column) that the number of clusters induced by the random-based attacks increases from 3 to 4, while the objective function (top plot) remains at zero, and *Split* and *Merge* do not vary (see Table 1). This essentially means that all the corresponding attack points are clustered together in a single cluster, separated from the initial ones. Finally, it is worth noting that *Merge* approximately equals 1 in Table 1 for all the considered attacks, highlighting that even the effective (non-random) attacks are mostly able to split the initial clusters but not to form clusters that aggregate samples initially belonging to different clusters.

[5] Publicly available at `http://cs.nyu.edu/~roweis/data.html`.

6 Conclusions and Future Work

In this paper, we addressed the problem of evaluating the security of clustering algorithms in adversarial settings. We showed with real-world experiments that complete-linkage clustering may be significantly vulnerable to deliberate attacks. In general, finding the optimal attack strategy for an arbitrary clustering algorithm is a difficult problem. Therefore, we have to rely on heuristic algorithms in order to carry out our analysis. For the sake of efficiency, these heuristics should be heavily dependent on the targeted clustering algorithm, as in our case. Nevertheless, it would be interesting to devise more general methods that can use the clustering algorithm as a black box and find a solution by performing a stochastic search on the solution space (*e.g.*, by simulated annealing), or an educated exhaustive search (*e.g.*, by using branch-and-bound techniques).

Acknowledgements. This work has been partly supported by the project "Security of pattern recognition systems in future internet" (CRP-18293) funded by Regione Autonoma della Sardegna, L.R. 7/2007, Bando 2009.

References

1. Perdisci, R., Corona, I., Giacinto, G.: Early detection of malicious flux networks via large-scale passive DNS traffic analysis. IEEE Trans. Dependable and Secure Comp. 9(5), 714–726 (2012)
2. Pouget, F., Dacier, M., Zimmerman, J., Clark, A., Mohay, G.: Internet attack knowledge discovery via clusters and cliques of attack traces. J. of Information Assurance and Security 1(1) (2006)
3. Perdisci, R., Ariu, D., Giacinto, G.: Scalable fine-grained behavioral clustering of http-based malware. Computer Networks 57(2), 487–500 (2013)
4. Rieck, K., Trinius, P., Willems, C., Holz, T.: Automatic analysis of malware behavior using machine learning. J. Comput. Secur. 19(4), 639–668 (2011)
5. Hanna, S., Huang, L., Wu, E., Li, S., Chen, C., Song, D.: Juxtapp: A scalable system for detecting code reuse among Android applications. In: Flegel, U., Markatos, E., Robertson, W. (eds.) DIMVA 2012. LNCS, vol. 7591, pp. 62–81. Springer, Heidelberg (2013)
6. Burguera, I., Zurutuza, U., Nadjm-Tehrani, S.: Crowdroid: behavior-based malware detection system for android. In: SPSM 2011, pp. 15–26 (2011)
7. Spitzner, L.: Honeypots: Tracking Hackers. Addison-Wesley Professional (2002)
8. Biggio, B., Fumera, G., Roli, F.: Security evaluation of pattern classifiers under attack. IEEE Trans. Knowledge and Data Eng. 26(4), 984–996 (2014)
9. Brückner, M., Kanzow, C., Scheffer, T.: Static prediction games for adversarial learning problems. J. Mach. Learn. Res. 13, 2617–2654 (2012)
10. Huang, L., Joseph, A.D., Nelson, B., Rubinstein, B., Tygar, J.D.: Adversarial machine learning. In: ACM Workshop AISec 2011, pp. 43–57 (2011)
11. Barreno, M., Nelson, B., Sears, R., Joseph, A.D., Tygar, J.D.: Can machine learning be secure? In: ASIACCS 2006, pp. 16–25 (2006)
12. Großhans, M., Sawade, C., Brückner, M., Scheffer, T.: Bayesian games for adversarial regression problems. In: ICML, vol. 28 (2013)

13. Dutrisac, J.G., Skillicorn, D.: Hiding clusters in adversarial settings. In: ISI 2008, pp. 185–187 (2008)
14. Skillicorn, D.B.: Adversarial knowledge discovery. IEEE Intelligent Systems 24, 54–61 (2009)
15. Biggio, B., Pillai, I., Rota Bulò, S., Ariu, D., Pelillo, M., Roli, F.: Is data clustering in adversarial settings secure? In: ACM Workshop AISec 2013, pp. 87–98 (2013)
16. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines. In: ICML (2012)
17. Kolcz, A., Teo, C.H.: Feature weighting for improved classifier robustness. In: CEAS (2009)
18. Jain, A.K., Dubes, R.C.: Algorithms for clustering data. Prentice-Hall, Inc., Upper Saddle River (1988)
19. Meilă, M.: Comparing clusterings: An axiomatic view. In: ICML, pp. 577–584 (2005)
20. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. Journal of Intelligent Information Systems 17(2-3), 107–145 (2001)
21. LeCun, Y., Jackel, L., Bottou, L., Brunot, A., Cortes, C., Denker, J., Drucker, H., Guyon, I., Müller, U., Säckinger, E., Simard, P., Vapnik, V.: Comparison of learning algorithms for handwritten digit recognition. In: Int'l Conf. on Artificial Neural Networks, pp. 53–60 (1995)

A Comparison of Categorical Attribute Data Clustering Methods

Ville Hautamäki[1], Antti Pöllänen[1], Tomi Kinnunen[1], Kong Aik Lee[2], Haizhou Li[2], and Pasi Fränti[1]

[1] School of Computing, University of Eastern Finland, Finland
[2] Institute for Infocomm Research, A*STAR, Singapore
villeh@cs.uef.fi

Abstract. Clustering data in Euclidean space has a long tradition and there has been considerable attention on analyzing several different cost functions. Unfortunately these result rarely generalize to clustering of categorical attribute data. Instead, a simple heuristic k-modes is the most commonly used method despite its modest performance. In this study, we model clusters by their empirical distributions and use expected entropy as the objective function. A novel clustering algorithm is designed based on local search for this objective function and compared against six existing algorithms on well known data sets. The proposed method provides better clustering quality than the other iterative methods at the cost of higher time complexity.

1 Introduction

The goal of *clustering* [1] is to reveal hidden structures in a given data set by grouping similar data objects together while keeping dissimilar data objects in separated groups. Let X denote the set of data objects to be clustered. The classical clustering problem setting considers data objects in a D-dimensional vector space, $X \subset \mathbb{R}^D$. The most commonly used objective function for such data is *mean squared error* (MSE). A generic solution is the well-known k-means method [2], which consists of two steps that are iterated until convergence. In *assignment step* (or E-step), all vectors are assigned to new clusters and *re-estimation step* (or M-step), model parameters are updated based on the new assigments.

Different from vector space data, data in educational sciences, sociology, market studies, biology and bioinformatics often involves *categorical attributes*, also known as *nominal* data. For instance, a data object could be a single questionnaire form that consists of multiple-choice questions. Possible outcomes of the answers can be encoded as integers. In this way, each questionnaire would be represented as an element of $\mathbb{N}^D$, where D is the number of questions. Unfortunately, since, the categories do not have any natural ordering, applying clustering methods developed for metric space data cannot be applied as such.

Hamming distance is a distance function designed for categorical data. It counts the number of attributes where two vectors disagree, i.e., having different

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 53–62, 2014.

attribute values. Cost functions and algorithms based on Hamming distance include *k-medoids* [3] and *k-modes* [4], both being extensions of the classical k-means [2]. In k-medoids, cluster representative (*discrete median*) is a vector in the cluster that minimizes the sum of distances from all other vectors to the cluster representative. In k-modes, the representative is the *mode* of the cluster, calculated independently for every attribute. Mode is the most frequently occurring value, in one attribute, over all the vectors in the cluster.

Using minimum Hamming distance as the assignment rule, one is also faced with the so-called zero probability condition [5]. It is one of the the assumptions behind the convergence proof of the classical k-means algorithm, stating that that the probability of assigning a vector to more than one cluster must be zero. With real valued data this condition holds. However, in tne case of categorical attribute clustering based on Hamming distance this condition is clearly not met. In the extreme case, when two D-dimensional vectors are maximally different, their Hamming distance is D. Consequently, the Hamming distance can take up only D unique values and it is likely that a vector is equally close to more than one cluster. Moreover, in the k-modes method, the cluster representative (mode) is not unique either. Tie-breaking needs to be employed in both the E- and the M-steps.

Tie-breaking problem in the cluster assignment (E-step) phase can be solved by testing each vector one by one whether its move to a new cluster will improve the objective function value. If such a cluster is found, the cluster parameters are immediately updated. Convergence of the algorithm can then be detected when there is no movement of vectors. One way to tackle the tie-breaking problem in the M-step is to represent the cluster by its *probability mass function* (pmf), that is, the relative frequencies of each category value in the cluster. For example, choices for educational background could have values $P(\text{elementary school}) = 0.2$, $P(\text{high school}) = 0.7$ and $P(\text{vocational school}) = 0.1$. In a sense, k-modes can be considered as a quantized version of the pmf-based cost functions. In this example, "high school" would be the cluster representative.

A number of different objective functions have been proposed, based on the the idea of modeling each cluster by its pmf: *k-histograms* [6, 7], *k-distributions* [8], *minimum description length* (MDL) [9], *mutual information* [10] and *expected entropy* [11–14]. Expected entropy is the average entropy of the entire clustering. If the pmf of the cluster is sharply peaked, its entropy is small. Therefore minimizing the expected entropy leads to compact clusters.

Despite the availability of multiple pmf-based methods, it is unclear which objective function and method would be best suited for a given application. In this work, we compare six well known categorical clustering methods in diverse categorical data sets using expected entropy as a clustering quality measure. Data sets vary from small sets of only 47 data points to large data set of more than 47k entries. In addition, we propose a new local search algorithm that directly optimizes the expected entropy.

2 Modeling Cluster by Its Distribution

In hard clustering, the goal is to divide a data set X, of size N, into disjoint clusters $\mathcal{V} = \{V_1, V_2, \ldots, V_M\}$, where $V_i \subset X, \cup_{i=1}^{M} V_i = X$, and $V_i \cap V_j = \emptyset \quad \forall\, i \neq j$.

In categorical clustering, data set consists of vectors $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)$, where each x_d takes values from a discrete set (categories). The number of categories in dimension d is denoted by C_d. We assume, without a loss of generality that $x_d \in \{1, \ldots, C\}$, where $C = \max_{d=1\ldots D} C_d$.

Entropy [15] is a measure of "surprise" in the data. High entropy signifies flat distribution whereas low entropy signifies peaked distribution. Formally, entropy for discrete distribution is defined as:

$$H(X) \triangleq -\sum_{\boldsymbol{x} \in X} p(\boldsymbol{x}) \log p(\boldsymbol{x}), \tag{1}$$

where $p(\boldsymbol{x}) = p(x_1, \ldots, x_D)$. Here, $p(\boldsymbol{x})$ denotes *estimated probability* of the joint event $(x_1, \ldots, x_D)$. In the rest of the discussion, by entropy we will mean *estimated entropy*, also known as *empirical entropy*.

Our goal is to minimize the so-called *expected entropy*) [12]:

$$H(\mathcal{V}) \triangleq \sum_{m=1}^{M} \frac{|V_m|}{N} H(V_m), \tag{2}$$

where $|V_m|$ is the cardinality of V_m and $H(V_m)$ is the entropy of the cluster V_m. Note that by setting $M = 1$, we obtain $H(\mathcal{V}) = H(X)$, and by setting $M = N$, we obtain $H(\mathcal{V}) = 0$, where each vector is in its own cluster. All other values are between these two extremes.

3 Algorithms

We evaluate two different types of clustering approaches, *iterative* and *agglomerative*. In iterative algorithms, clustering cost is improved in each iteration by repartitioning the datasets. The selected algorithms are summarized in Table 1. In agglomerative algorithms, instead, clusters are merged one by one until a desired number of clusters is reached. Two agglomerative methdos are considered: ACE [14], which optimizes the expected entropy (2), and ROCK [16] which optimizes its own cost function.

3.1 Prototype Based Algorithms

Prototype-based iterative methods [3, 4] select one vector from each cluster as a representative, analogous to centroid vector in conventional k-means. The k-modes and k-medoid methods use minimum Hamming distance to assign vectors to clusters. In the classical k-means, squared Euclidean distance was used. In the M-step, the goal is to find such a prototype per cluster that minimizes Hamming distance from each vector in the cluster to the prototype vector. In k-medoid, one vector from the cluster is selected as the prototype and in k-modes, most frequently observed category per dimension is selected.

Table 1. Summary of k-means type methods experimented in this study, classified according to cluster representative type and distance measure

Method	Representative	Measure
k-distributions [8]	Distribution	Product of m-estimates
k-histograms [6, 7]	Distribution	non-matching frequencies
k-modes [4]	Mode	Hamming distance
k-medoids [3]	Medoid	Hamming distance
k-entropies [**this paper**]	Distribution	Entropy change

3.2 k-Distributions

In k-distributions [8], there are no cluster prototypes but the histograms are used to represent clusters. In the E-step, a vector is assigned to the cluster that maximizes the likelihood $p(\boldsymbol{x}|V_m)$. The likelihood can be factorized into each dimension separately assuming that dimensions are independent. Some categories may have zero count, the histogram is therefore processed by *Laplacian smoothing* [17].

The expected entropy is not directly optimized by k-distributions. No proof of convergence exists, but experimentally we have noticed that the method seems to converge, albeit slowly. In the following, we attempt to give an explanation of the slow convergence. It would benefit if the similarity measure between a vector and cluster remains relatively stable when only small changes are made in the cluster partitioning. Unfortunately, this is not the case with k-distributions. Let us consider a case where we map a vector to a cluster, where one dimension has a non-matching category (no vector in the cluster has that category). When a new vector having this non-matching category is added to the cluster, comparing likelihood before and after addition we notice a large difference. For example, the likelihood from a vector after addition of the cluster with 15 vectors and 3 categories is 3.5 times more than before the addition. Thus, vectors end up changing clusters very often, leading to a slow convergence.

3.3 k-Representatives and k-Histograms

K-representatives [7] first assigns randomly all vectors to clusters and computes normalized histograms as representatives of each cluster. Frequencies are normalized so that they sum up to one. The distance measure from vector to cluster is Hamming distance weighted by the frequency. The method assigns new vectors to clusters based on the distance measure and recomputes the histograms. Process continues until no re-assignments of vectors are detected.

Unfortunately, contrary to the claim in [7], we found out that algorithm does not always converge[1], We, therefore, do not consider k-representatives method

[1] Proof by explicit construction of a 5-dimensional data set that k-representatives does not converge: `http://cs.uef.fi/sipu/krepresentatives.pdf`.

further. In iterative clustering with immediate update, vector is moved from one cluster to another if the move *decreases* the cost function. We considere here k-histograms [6] cost, which is the sum of k-representatives distance measures. It is a non-negative cost function, thus, the algorithm converges in a finite number of steps. The k-histograms method uses the immediate update strategy, otherwise it is the same as k-representatives.

3.4 Agglomerative Methods

A *robust clustering algorithm for categorical attributes* (ROCK) [16] defines a cost function based on the idea of neighbours and links. Neighbourhood of each vector is decided based on thresholded distance between vectors. We use Hamming distance. A link between two vectors is made if they share at least one neighbour. The goal of ROCK is to maximize pairwise links between vectors within the clusters, and minimize links between clusters. In each iteration, ROCK merges two the clusters that maximizes this criterion.

In *Agglomerative Categorical clustering with Entropy criterion* ACE method [14], expected entropy is optimized. In each iteration, ACE merges two clusters, V_i and V_j, so that the *incremental entropy* is minimized:

$$I_m(V_i, V_j) = H(V_i \cup V_j) - H(V_i) - H(V_j). \tag{3}$$

3.5 The Proposed Method

We propose to optimize the expected entropy directly. We start by randomly assigning each vector to a cluster. The method then iterates over all vectors and tests whether moving it to a new cluster improves the expected entropy. The assignment that maximally improves is selected. The algorithm converges when no vector changes its cluster assignment. It is easy to see that this strategy converges as each iteration is forced to either improve on the previous solution, or keep the existing one and stop. The proposed method is summarized in Algorithm 1.

Algorithm 1. The proposed method (k-entropies)

```
Randomly assign all vectors to M clusters.
Model clusters as their probability mass function (pmf).
repeat
  for x ∈ X do
    V_i ← Assign x according to minimum cost (2).
    Estimate prototype of the cluster V_i as the pmf of the cluster.
  end for
until No change in vector assignments.
```

3.6 Summary

All the algorithms, mentioned above are summarized in Table 2. The time and space complexities for ACE and ROCK are referenced from the respective publications, and the others have been derived by ourselves. The quadratic space and time complexity of both ACE and ROCK makes them rather impractical for large data sets. Here, I denotes the number iterations, C_{avg} the average number of categories, T_L the cost of computing logarithm, $R_{\max}$ the maximum number of neighbours, and R_{avg} the average number of neighbours.

Table 2. Summary of clustering algorithms

Algorithm	Type	Time complexity	Space complexity
ACE [14]	Agglomerative	$O(N^2 \log N + N^2 D C_{\text{avg}})$	$O(N^2)$
ROCK [16]	Agglomerative	$O(N^2 \log N + N^2)$	$O(\min\{N^2, N R_{\max} R_{\text{avg}}\})$
k-medoids [3]	k-means	$O(INMD)$	$O(N)$
k-modes [4]	k-means	$O(INMD)$	$O(N)$
k-distributions [8]	k-means	$O(INMDT_L)$	$O(N)$
k-histograms [6]	Immediate update	$O(INMD)$	$O(N)$
k-entropies	Immediate update	$O(INMDC_{\text{avg}}T_L)$	$O(N)$

4 Experiments

Experimental comparison were performed using six different categorical data sets (Mushroom, Votes, Soybean, CENSUS, SPECT hearth and Plants) obtained from UCI Machine Learning archive [18]. Data sets are summarized in Table 3.

Only two methods optimize directly the expected entropy: ACE and k-entropies (proposed method). We are interested to find out how the other methods perform in terms of expected entropy as a clustering objective function, where low entropy is desired. For iterative schemes, the number of iterations I depends on initialization, data set and cost function, it can have importance on how fast the algorithm is in practice. Number of iterations I measures the empirical convergence speed of the algorithm.

Mushroom data set includes 8124 observations from 23 different mushroom species. it has 21 attributes, and 119 categories. Dimensions with large number of missing values were discarded in our tasks. Dimensions of the vectors encode forms and colours of the mushrooms. **Congressional votes** data set includes votes from the US Congress of 1984. Each vector gives the votes of one of the 435 member of the US congress. In total, proposals were collected with possible outcome of {yes, no, other}, where other means that politicians opinion on the said proposal is not known. Total number of categories is 46. **Soybean** data set contains observations from different soybeans. It contains 47 vectors, 35 dimensions and 72 categories. **CENSUS** data set is selected to evaluate scalability of

the compared methods. Data set size is 2,458,285, has 68 dimensions and 396 categories. This data set contains both nominal and ordinal data types. In our experiments, special processing for ordinal data is not used. **SPECT hearth** is data on cardiac *single proton emission computed tomography* (SPECT) images. Each SPECT image was summarized to 22 binary pattern features. Data set contains 267 patients, and 44 categories. **Plants** data set is transaction data about different growth locations, containing 34781 vectors (plants), 70 dimensions and 140 categories.

Table 3. Data set summary

Data set	Vectors	D	Categories	Entropy
Mushroom	8124	21	119	21.44
Votes	435	16	46	13.98
Soybean	47	35	72	18.80
CENSUS	2458285	68	396	55.17
SPECT hearth	267	22	44	13.68
Plants	34781	70	140	25.35

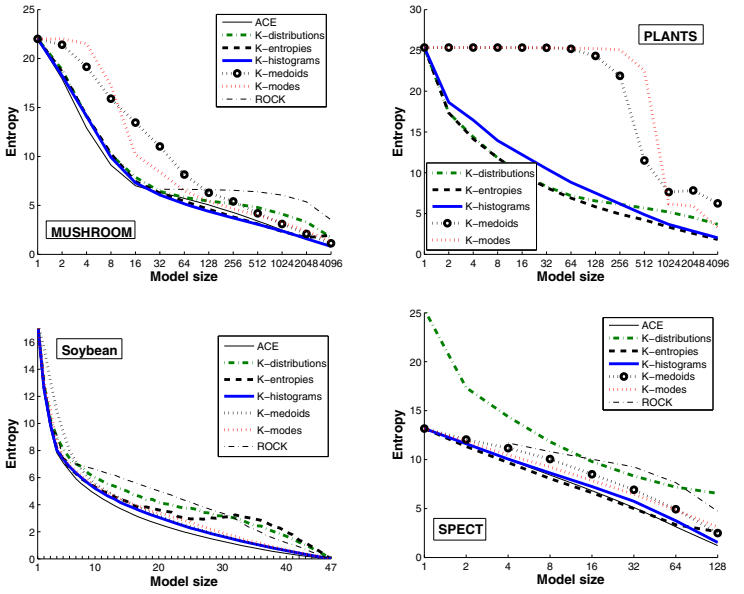


Fig. 1. Expected entropy as a function of model size. In order from top left to bottom right: mushroom, plants, soybean and spect data sets.

4.1 Quality of Clustering

Fig. 1 shows the expected entropy as a function of model size. First glance validates our intuition: methods that are based on optimizing distribution perform similarly. In general, the order of performance is: ACE first, then k-entropies and after that k-histograms. K-distributions gives different results for SPECT data set, when comparing to other sets.

The prototype based methods, k-medoids and k-modes optimize sum of Hamming distances and perform similarly, as expected. They cluster to the mushroom and plants data sets differenty than the pmf-based methods. ROCK also seems to follow its own trend. If no links exist between two clusters, then there is no way to merge them. This behaviour is visible in mushroom and SPECT data sets, in smaller model sizes ROCK is not able to obtain any results. Plants is transaction data, where most attributes have zero values, resulting all zero vector as a prototype with k-modes and k-medoids.

4.2 Summary of Experiments

Summary of average expected entropies and processing times with standard deviations is shown in Table 4 and 5, when repeating all experiments 10 times.

Table 4. Summary obtained average expected entropies and standard deviations.

Algorithm	Soybean $M=4$		Mushroom $M=16$		Votes $M=2$		SPECT $M=8$		Plants $M=1024$		CENSUS $M=16$	
	$H(\mathcal{V})$	std	$H(\mathcal{V})$	std	$H(\mathcal{V})$	std	$H(\mathcal{V})$	std	$H(\mathcal{V})$	std	$H(\mathcal{V})$	std
ACE	**7.83**	**0**	**7.01**	n/a	9.79	0.16	8.43	0.07	n/a	n/a	n/a	n/a
ROCK	9.20	**0**	n/a	n/a	**9.30**	**0**	10.82	**0**	n/n	n/a	n/a	n/a
k-medoids	10.94	1.57	13.46	0.71	10.00	0.95	10.06	0.44	7.64	0.33	32.61	0.87
k-modes	8.66	1.06	10.19	1.45	9.70	0.03	9.25	0.29	6.19	0.18	31.00	1.26
k-distributions	9.00	0.93	7.87	0.44	9.59	0.01	8.25	0.13	5.19	0.09	28.58	**0.27**
k-representatives	8.30	0.83	7.51	0.30	9.60	0.01	8.64	0.13	n/a	n/a	n/a	n/a
k-histograms	8.04	0.53	7.31	**0.18**	9.60	0.01	8.64	0.15	3.67	**0.02**	29.17	0.48
k-entropies	8.15	0.56	7.31	**0.18**	9.58	**0**	**8.06**	0.07	**3.33**	0.05	**28.51**	0.50

Table 5. Summary obtained average processing times (in seconds) and standard deviations

Algorithm	Mushroom		Plants		CENSUS	
	Time	std	Time	std	Time	std
ACE	2565.76	n/a	n/a	n/a	n/a	n/a
ROCK	n/a	n/a	n/a	n/a	n/a	n/a
k-medoids	0.06	0	30.89	8.12	184.95	18.63
k-modes	0.08	0.01	63.60	10.75	188.65	24.17
k-distributions	1.83	0.17	5043.80	1637.70	1748.34	530.97
k-representatives	0.30	0.07	n/a	n/a	n/a	n/a
k-histograms	0.19	0.04	467.46	110.44	900.35	81.81
k-entropies	12.51	1.43	8669.55	1000.84	6068.72	1116.87

Entry with n/a means that algorithm was not able to produce a result for that configuration, either due to non-convergence or running out of memory. Model sizes were selected for each data set separately, either by looking at the expected entropy as function of model size plot, or by information from the data set descriptions. For the plants data set we selected 1024, because for smaller model sizes k-modes and k-medoids completely fail.

We notice that for Soybean and Mushroom data sets ACE is the best as it directly optimizes the expected entropy. However, for the votes and SPECT data sets the proposed method provides better clustering, than ACE. The usability of ACE and ROCK are limited to their space complexity: those methods are not able to cluster largest sets at all. The proposed method is the best in terms of quality for the SPECT, plants and CENSUS data sets.

K-representatives results were also obtained for illustrative purposes for the datasets it converged on. It is slower than k-histograms, which can be attributed to the non-convergence behaviour of the algorithm. In terms of expected entropy, k-representatives iteration strategy did not provide any visible advantage over the immediate update of the k-histograms.

When comparing the proposed method and ACE in terms of processing time, we see that the proposed method is a clear winner. However, other methods that do not directly optimize expected entropy are clearly much faster.

5 Conclusions

We have compared existing pmf-based categorical clustering methods and found them to be very similar in terms of expected entropy. We also found out that the prototype-based methods (k-medoids and k-modes), while being the fastest methods, are not able to reach the lowest expected entropy obtained by the pmf-based methods. Thus, those methods are not recommended for clustering categorical data sets. On the other hand, ACE, while providing the best overall results, is not well-suited for large data sets, because of its quadratic time and space complexities. The proposed k-entropies method yielded the best results for the larger datasets. As a future work, we plan to investagate ways to obtain a k-means type clustering algorithm for the expected entropy cost.

Acknowledgements. This work was supported by Academy of Finland (projects 253000 and 253120).

References

1. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)
2. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. I, pp. 281–297. University of California (1967)

3. Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis. John Wiley Sons, New York (1990)
4. Huang, Z.: Extensions to k-means algorithm for clustering large data sets with categorical values. Data Mining Knowledge Discovery 2(3), 283–304 (1998)
5. Gersho, A., Gray, R.M.: Vector Quantization and Signal Compression. Kluwer Academic Publisher, Boston (1992)
6. He, Z., Xu, X., Deng, S., Dong, B.: K-histograms: An efficient clustering algorithm for categorical dataset. CoRR abs/cs/0509033 (2005)
7. San, O.M., Huynh, V.N., Nakamori, Y.: An alternative extension of the k-means algorithm for clustering categorical data. International Journal of Applied Mathematics and Computer Science 14(2), 241–247 (2004)
8. Cai, Z., Wang, D., Jiang, L.: K-distributions: A new algorithm for clustering categorical data. In: Huang, D.-S., Heutte, L., Loog, M. (eds.) ICIC 2007. LNCS (LNAI), vol. 4682, pp. 436–443. Springer, Heidelberg (2007)
9. Chakrabarti, D., Papadimitrou, S., Modha, D.S., Faloutsos, C.: Fully automatic cross-associations. In: Proceedings of the ACM SIGKDD Conference (2004)
10. Andritsos, P., Tsaparas, P., Miller, R.J., Sevcik, K.C.: LIMBO: Scalable clustering of categorical data. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 123–146. Springer, Heidelberg (2004)
11. Barbará, D., Li, Y., Couto, J.: Coolcat: an entropy-based algorithm for categorical clustering. In: CIKM 2002: Proceedings of the Eleventh International Conference on Information and Knowledge Management, pp. 582–589. ACM, New York (2002)
12. Li, T., Ma, S., Ogihara, M.: Entropy-based criterion in categorical clustering. In: ICML 2004: Proceedings of the Twenty-First International Conference on Machine Learning, p. 68. ACM, New York (2004)
13. Li, T.: A unified view on clustering binary data. Machine Learning 62(3), 199–215 (2006)
14. Chen, K., Liu, L.: The "best k" for entropy-based categorical data clustering. In: Proceedings of the 17th International Conference on Scientific and Statistical Database Management (SSDBM 2005), Berkeley, USA, pp. 253–262 (2005)
15. Cover, T., Thomas, J.: Elements of Information Theory. Wiley-Interscience (1991)
16. Guha, S., Rastogi, R., Shim, K.: Rock: A robust clustering algorithm for categorical attributes. Information Systems 25(5), 345–366 (2000)
17. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
18. Asuncion, A., Newman, D.: UCI machine learning repository (2007)

Improving Approximate Graph Edit Distance Using Genetic Algorithms

Kaspar Riesen[1], Andreas Fischer[2], and Horst Bunke[3]

[1] Institute for Information Systems, University of Applied Sciences and Arts Northwestern Switzerland, Riggenbachstrasse 16, 4600 Olten, Switzerland
kaspar.riesen@fhnw.ch
[2] Biomedical Science and Technologies Research Centre, Polytechnique Montreal 2500 Chemin de Polytechnique, Montreal H3T 1J4, Canada
andreas.fischer@polymtl.ca
[3] Institute of Computer Science and Applied Mathematics, University of Bern, Neubrückstrasse 10, 3012 Bern, Switzerland
bunke@iam.ch

Abstract. Many flexible methods for graph dissimilarity computation are based on the concept of edit distance. A recently developed approximation framework allows one to compute graph edit distances substantially faster than traditional methods. Yet, this novel procedure considers the local edge structure only during the primary optimization process. Hence, the speed up is at the expense of an overestimation of the true graph edit distances in general. The present paper introduces an extension of this approximation framework. Regarding the node assignment from the original approximation as a starting point, we implement a search procedure based on a genetic algorithm in order to improve the approximation quality. In an experimental evaluation on three real world data sets a substantial gain of distance accuracy is empirically verified.

1 Introduction

Graph matching refers to the process of evaluating the structural similarity of graphs. A large number of methods for graph matching have been proposed in recent years (see [1, 2] for exhaustive surveys). Due to its ability to cope with arbitrarily structured graphs with unconstrained label alphabets for both nodes and edges, the concept of graph edit distance [3] can be applied to virtually any kind of graphs. Therefore, graph edit distance has been used in the context of classification and clustering tasks in diverse applications [4–6].

Given two graphs, the source graph g_1 and the target graph g_2, the basic idea of graph edit distance is to transform g_1 into g_2 using some distortion operations. A standard set of distortion operations is given by *insertions*, *deletions*, and *substitutions* of both nodes and edges. We denote the substitution of two nodes u and v by $(u \rightarrow v)$, the deletion of node u by $(u \rightarrow \varepsilon)$, and the insertion of node v by $(\varepsilon \rightarrow v)$[1]. A sequence of edit operations $e_1, \ldots, e_k$ that transform g_1 completely into g_2 is called an *edit path* between g_1 and g_2.

[1] For edges we use a similar notation.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 63–72, 2014.

Let $\Upsilon(g_1, g_2)$ denote the set of all possible edit paths between two graphs g_1 and g_2. To find the most suitable edit path out of $\Upsilon(g_1, g_2)$, one introduces a cost for each edit operation, measuring the strength of the corresponding operation. The *edit distance* of two graphs is then defined by the minimum cost edit path between two graphs.

The computation of exact graph edit distance is usually carried out by means of a tree search algorithm which explores the space of all possible mappings of the nodes and edges of the first graph to the nodes and edges of the second graph. A widely used method is based on the A* algorithm [7] which is a best-first search algorithm. The computational complexity of the exact edit distance algorithm, whether or not heuristic functions are used to govern the tree traversal process, is exponential in the number of nodes of the involved graphs. Consequently, exact edit distance can be computed for graphs of a rather small size only.

In recent years, a number of methods addressing the high computational complexity of graph edit distance computation have been proposed (e.g. [8–11]). The authors of the present paper also introduced an algorithmic framework which allows the approximate computation of graph edit distance in a substantially faster way than traditional methods [12]. Yet, the substantial speed-up in computation time is at the expense of an overestimation of the actual graph edit distance. The reason for this overestimation is that the algorithm is able to consider only local, rather than global, edge structure during the optimization process. The main objective of the present paper is to significantly reduce the overestimation of edit distances in our approximation framework. To this end, the distance approximation found by the procedure of [12] is systematically improved using a search procedure based on genetic algorithms.

Genetic algorithms have been proposed in the context of error-tolerant graph matching in various publications [13–15]. The basic idea of this approach is to formalize matchings as states (*chromosomes*) with a corresponding performance (*fitness*). An initial pool of these chromosomes, i.e. matchings, evolves iteratively into other generations of matchings. To this end, different genetic operations are applied to the current matchings. Though the search space is explored in a random fashion, genetic algorithms can be designed so as to favour promising chromosomes, i.e. well fitting matchings, and further improve them by specific genetic operations.

The remainder of this paper is organized as follows. Next, in Sect. 2 the original framework for graph edit distance approximation [12] is summarized. In Sect. 3 the extension of this specific framework using a genetic search procedure is introduced. An experimental evaluation on diverse data sets is carried out in Sect. 4, and in Sect. 5 we draw some conclusions and outline some possible tasks and extensions for future work.

2 Bipartite Graph Edit Distance Approximation

In the framework presented in [12], for matching two graphs g_1 and g_2 with nodes $V_1 = \{u_1, \ldots, u_n\}$ and $V_2 = \{v_1, \ldots, v_m\}$, respectively, a cost matrix $\mathbf{C}$ is first established as follows:

$$\mathbf{C} = \left[\begin{array}{cccc|cccc} c_{11} & c_{12} & \cdots & c_{1m} & c_{1\varepsilon} & \infty & \cdots & \infty \\ c_{21} & c_{22} & \cdots & c_{2m} & \infty & c_{2\varepsilon} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \infty \\ c_{n1} & c_{n2} & \cdots & c_{nm} & \infty & \cdots & \infty & c_{n\varepsilon} \\ \hline c_{\varepsilon 1} & \infty & \cdots & \infty & 0 & 0 & \cdots & 0 \\ \infty & c_{\varepsilon 2} & \ddots & \vdots & 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \infty & \vdots & \ddots & \ddots & 0 \\ \infty & \cdots & \infty & c_{\varepsilon m} & 0 & \cdots & 0 & 0 \end{array}\right]$$

Entry c_{ij} thereby denotes the cost of a node substitution $u_i \rightarrow v_j$, $c_{i\varepsilon}$ denotes the cost of a node deletion $u_i \rightarrow \varepsilon$, and $c_{\varepsilon j}$ denotes the cost of a node insertion $\varepsilon \rightarrow v_j$.

Obviously, the left upper corner of the cost matrix represents the costs of all possible node substitutions, the diagonal of the right upper corner the costs of all possible node deletions, and the diagonal of the bottom left corner the costs of all possible node insertions. Note that each node can be deleted or inserted at most once. Therefore any non-diagonal element of the right-upper and left-lower part is set to ∞. The bottom right corner of the cost matrix is set to zero since substitutions of the form $(\varepsilon \rightarrow \varepsilon)$ should not cause any costs. In the definition of cost matrix $\mathbf{C}$, to each entry c_{ij}, i.e. to each cost of a node edit operation, the minimum sum of edge edit operation costs, implied by the corresponding node operation, is added (i.e. the matching cost arising from the local edge structure is encoded in the individual entries of $\mathbf{C}$).

On the basis of the square cost matrix $\mathbf{C}$ a bipartite assignment algorithm is executed (first step). The result returned by this bipartite optimization procedure corresponds to the minimum cost mapping m of the nodes and their local edge structure of g_1 to the nodes and their local edge structure of g_2. Mapping m can be seen as partial edit path $\pi = e_1, \ldots, e_l$, where each edit operation $e_i \in \pi$ reflects an operation on nodes from V_1 and/or V_2 (deletions, insertions or substitutions). In a second step the edit path π between g_1 and g_2 is completed according to mapping m. Note that edit operations on edges are implied by edit operations on their adjacent nodes, i.e. whether an edge is substituted, deleted, or inserted, depends on the edit operations performed on its adjacent nodes. Hence, given the set of node operations $e_1, \ldots, e_l$, the global edge structures from g_1 and g_2 can be edited accordingly. The cost of the complete edit path π is finally returned as an approximate graph edit distance. We denote the approximated edit distance between graphs g_1 and g_2 according to mapping m with $d(g_1, g_2, m)$.

Note that the edit path corresponding to $d(g_1, g_2, m)$ considers the edge structure of g_1 and g_2 in a global and consistent way while the optimal node mapping m from step 1 is able to consider the structural information in an isolated way only (single nodes and their adjacent edges). This is due to the fact that during the optimization process no information about neighboring node mappings is available. Hence, in comparison with optimal search methods for graph edit distance, our novel algorithmic framework might cause additional edge operations

in the second step, which would not be necessary in a globally optimal graph matching. Hence, the distances found by this specific framework are – in the optimal case – equal to, or – in a suboptimal case – larger than the exact graph edit distance.

For the remainder of this paper we denote this graph edit distance approximation algorithm with *BP* (*Bipartite*).

3 Improving Graph Edit Distance Approximations Using Genetic Algorithms

In several experimental evaluations we observed that the suboptimality of BP is very often due to a few incorrectly assigned nodes in m. That is, only few node assignments from the first step are responsible for the additional edge operations in the second step (and the resulting overestimation of the true edit distance). Our novel procedure ties in at this observation. Rather than returning the approximate edit distance directly, a genetic search procedure based on mapping m is started.

The chromosomes in our genetic search procedure are mappings related to our original node assignment m. In order to build an initial population $P(0)$ containing chromosomes (mappings), we compute N random variations $\{m_1^{(0)}, \ldots, m_N^{(0)}\}$ of m. A single variation $m_i^{(0)} \in P(0)$ of m is computed as follows. Every node assignment $u_i \to v_j$ in m is possibly omitted with a certain probability p (referred to as *mutation probability*). That is, in an alternative mapping we enforce nodes u_i and v_j to be assigned to other nodes than v_j and u_i, respectively. This is ensured by means of an update of the cost matrix $\mathbf{C}$ such that entry $c_{i,j}$ (corresponding to the assignment $u_i \to v_j$) is set to ∞. Given the updated cost matrix (with ∞-entries at certain positions) an optimal node assignment is computed using our former procedure. This results in a new mapping $m_i^{(0)}$ which does not contain $(u_i \to v_j)$ any more. Note that $m_i^{(0)}$ corresponds to an optimal node assignment based on the altered cost matrix. Hence, $m_i^{(0)}$ is consistent, i.e. every node of g_1 is assigned to a single node of g_2 (or deleted) and every node of g_2 is assigned to a single node of g_1 (or inserted).

This mutation procedure is repeated N times to mapping m in order to get N different mappings $P(0) = \{m_1^{(0)}, \ldots, m_N^{(0)}\}$ and thus N different approximations of the true graph edit distance[2]. Note that all of these approximate edit distance values are still equal to, or larger than, the exact distance values. Hence, without knowing the exact graph edit distance, the fitness of every assignment $m_i^{(0)}$ can be rated according to its specific distance value $d(g_1, g_2, m_i^{(0)})$, viz. the lower $d(g_1, g_2, m_i^{(0)})$ the better the fitness of $m_i^{(0)}$.

Given the initial population $P(0)$ the following iterative procedure is carried out next. A new population $P(t+1)$ of mappings is built upon a subset E of $P(t)$,

[2] Note that the original mapping m is initially added to $P(0)$ such that the approximation found by our extension is guaranteed to be at least as accurate as the original approximation $d(g_1, g_2, m)$.

often referred to as *parents*. In order to select the parents from a given population $P(t)$, the $(f \cdot N)$ best approximations, i.e. the approximations in $P(t)$ with lowest distance values, are selected ($f \in]0, 1]$). In our framework, all approximations from E are added without any modifications to the next population $P(t+1)$. This ensures that the best solution found so far will not be lost during the search procedure (known as *survival of the fittest*).

In order to derive the remaining mappings of the new population $P(t+1)$, the following procedure is repeated $(N - |E|)$-times. Two mappings m' and m'' from the pool of parents E are randomly selected and eventually combined to one mapping m. To this end, the cost matrices $\mathbf{C}' = c'_{i,j}$ and $\mathbf{C}'' = c''_{i,j}$ corresponding to mappings m' and m'', respectively, are merged by means of

$$\mathbf{C}_m = \max\{c'_{i,j}, c''_{i,j}\}$$

Based on $\mathbf{C}_m$ an optimal mapping m is computed and eventually added to $P(t+1)$. Due to the definition of $\mathbf{C}_m$ the node mappings omitted in at least one of the mappings m' and m'' will also be prevented in the merged mapping m. The detour via optimal assignment computation on a cost matrix $\mathbf{C}_m$ again ensures that the merged mapping m is consistent with the underlying graphs (nodes of both graphs are uniquely assigned to nodes of the other graph or deleted/inserted).

The two main steps of the genetic algorithm (selection of parents $E \subseteq P(t)$ and computation of a new generation of mappings $P(t+1)$ based on E) are repeated until the best distance approximation has not been improved during the last τ iterations. It is well known that genetic algorithms are not deterministic. Therefore, we repeat the complete search procedure s times from the scratch and return the overall best approximation found in these s runs (which makes the algorithmic procedure more stable and reduces the risk of finding a poor approximation due to a poor random initialization).

Given that the genetic search procedure stops after t iterations on average, the two main steps of our former approximation framework, namely the computation of an optimal mapping m based on a cost matrix and the derivation of the corresponding edit distance, have to be carried out $(s \cdot t \cdot N)$-times. Hence, one can expect that our extended framework increases the run time by the magnitude of $(s \cdot t \cdot N)$ compared to our original framework.

The complete algorithmic procedure is given in Alg. 1. Note that the first three lines of Alg. 1 correspond to the original framework *BP*, while line 4 to 18 describe the proposed extension, denoted by BPGA from now on.

4 Experimental Evaluation

For experimental evaluations, three data sets from the IAM graph database repository[3] for graph based pattern recognition and machine learning are used [16]. The first graph data set involves graphs that represent molecular compounds (AIDS). We construct graphs from the AIDS Antiviral Screen Database

[3] `www.iam.unibe.ch/fki/databases/iam-graph-database`

Algorithm 1. BPGA(g_1, g_2) (Meta Parameters: N, p, τ, f, s)

```
1: Build cost matrix C according to the input graphs g1 and g2
2: Compute optimal node assignment m on C
3: Derive edit path and approximate edit distance based on m
4: for i = 1, ..., s do
5:    build initial population P(0) of mappings {m_1^(0), ..., m_N^(0)} based on m using mutation
      probability p
6:    d_best = min_{i=1,...,N} {d(g1, g2, m_i^(0))}
7:    t = 0; l = 0
8:    while t − l < τ do
9:       select a subset E ⊆ P(t) of parents (|E| = f · N)
10:      build a new population P(t + 1) = {m_1^(t+1), ..., m_N^(t+1)} from E
11:      d = min_{i=1,...,N} {d(g1, g2, m_i^(t+1))}
12:      t = t + 1
13:      if d < d_best then
14:         d_best = d; l = t
15:      end if
16:   end while
17: end for
18: return d_best
```

of Active Compounds [17]. This data set consists of two classes (*active, inactive*), which represent molecules with activity against HIV or not. The molecules are converted into graphs in a straightforward manner by representing atoms as nodes and the covalent bonds as edges. Nodes are labeled with the number of the corresponding chemical symbol and edges by the valence of the linkage.

The second graph data set consists of graphs representing fingerprint images (FP) [18]. In order to obtain graphs from fingerprint images, the 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 nodes. Additional nodes are inserted in regular intervals between ending points and bifurcation points. Finally, undirected edges are inserted to link nodes that are directly connected through a ridge in the skeleton. Each node is labeled with a two-dimensional attribute giving its position. The edges are attributed with an angle denoting the orientation of the edge with respect to the horizontal direction.

The third data set consists of graphs representing symbols from architectural and electronic drawings (GREC) [19]. 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 nodes and labeled with a two-dimensional attribute giving their position. The nodes 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.

Our procedure BPGA has five meta parameters to be defined by the user (see Table 1 for a survey). In the following evaluations only two of them are altered in order to evaluate their impact on the approximation quality, viz. the population size N as well as the mutation probability p. The three remaining parameters (τ, f, s) are freezed to constants. In fact, in preliminary experimental evaluations it turns out that these parameters – given that they do not fall below a certain threshold – do nearly not affect the resulting approximations. We choose minimum values for τ, f, s such that stable and reasonable results on all of the three data sets can be observed. In the Table 1 the meta parameters and their respective values are summarized.

Table 1. Meta Parameters of BPGA

Parameter	Meaning	Value
N	Population Size	$\{50, 100\}$
p	Mutation probability that a given node mapping in m is prevented (needed to build $P(0)$)	$\{0.1, 0.3, 0.5, 0.7\}$
τ	Termination when best solution has not been improved during the last τ iterations	6
f	Percentage of chromosomes selected from $P(t)$ as parents to build $P(t+1)$	0.25
s	Number of runs	3

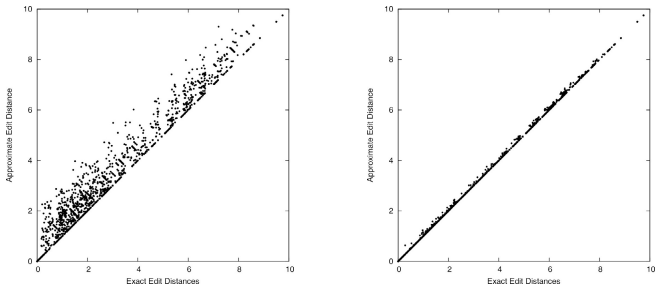
In Table 2 the achieved results are shown. On each data set and for each graph edit distance algorithm two characteristic numbers are computed, viz. the mean relative overestimation of the exact graph edit distance ($\varnothing o$) and the mean run time to carry out one graph matching ($\varnothing t$). The algorithms employed are A* and BP (reference systems) and eight differently parametrized versions of our novel procedure BPGA ($N \in \{50, 100\}$; $p \in \{0.1, 0.3, 0.5, 0.7\}$).

First we focus on the degree of overestimations and regard the results of BPGA with $N = 50$ only. The original framework (BP) overestimates the graph distance by 12.68% in average on the AIDS data. On the Fingerprint and GREC data the overestimations amount to 6.38% and 2.98%, respectively. These values can be substantially reduced with our extended framework. For instance on the AIDS data, the mean relative overestimation can be reduced to 2.01% in the best case ($p = 0.5$). That is, the mean relative overestimation of our novel framework is approximately six times smaller than the one of the original approximation framework. On the GREC data set the mean relative overestimation is reduced from 2.98% to 0.83% in the best case ($p = 0.3$) and on the Fingerprint data the overestimation can be heavily reduced from 6.38% to 0.13% ($p = 0.3$ or $p = 0.5$). We observe that a mutation probability between 0.3 and 0.5 works well on all three data sets.

Comparing the mean run time of our novel procedure with the original framework on the AIDS data, we observe that our extension takes approximately 300 times longer for one matching in average with $N = 50$ (approx. 167- and 200-times longer matching times in average on the Fingerprint and GREC data,

Table 2. The mean relative overestimation of the exact graph edit distance ($\varnothing o$) and the mean run time for one matching ($\varnothing t$) using a specific graph edit distance algorithm

Algorithm	Data Set					
	AIDS		FP		GREC	
	$\varnothing$ o	$\varnothing$ t	$\varnothing$ o	$\varnothing$ t	$\varnothing$ o	$\varnothing$ t
A* (Exact)	-	5.63	-	5.00	-	3.10
BP	12.68	0.0004	6.38	0.0006	2.98	0.0004
BPGA(50, 0.1)	2.96	0.12	0.20	0.10	1.00	0.08
BPGA(50, 0.3)	2.18	0.11	0.13	0.10	0.83	0.08
BPGA(50, 0.5)	2.01	0.12	0.14	0.10	0.83	0.08
BPGA(50, 0.7)	2.12	0.11	0.15	0.10	0.89	0.08
BPGA(100, 0.1)	2.33	0.23	0.14	0.20	0.82	0.16
BPGA(100, 0.3)	1.53	0.22	0.09	0.21	0.66	0.17
BPGA(100, 0.5)	1.42	0.23	0.09	0.20	0.68	0.17
BPGA(100, 0.7)	1.54	0.23	0.11	0.20	0.74	0.16



(a) BP (b) BPGA(100, 0.3)

Fig. 1. Exact (x-axis) vs. approximate (y-axis) graph edit distance

respectively). The observed run time increase perfectly lies within the expected multiplication of the average run time by $(s \cdot t \cdot N)$. However, compared to the exact algorithm our extension is still very fast (approximately 40 to 50 times faster on all data sets with $N = 50$).

Increasing the population size N to 100 allows us to further decrease the overestimation. Yet, the reduction is at the prize of approximately doubling the mean runtime when compared to $N = 50$ on all data sets. Also with $N = 100$ a mutation probability between 0.3 and 0.5 seems to be the best choice on all data sets.

The substantial improvement of the approximation accuracy can also be observed in the scatter plots in Fig. 1. These scatter plots give us a visual representation of the accuracy of the suboptimal methods on the Fingerprint data set[4]. We plot for each pair of graphs their exact (horizontal axis) and approximate

[4] On the other data sets similar results can be observed.

(vertical axis) distance value. The reduction of the overestimation using our proposed extension is clearly observable and illustrates the power of our extended framework.

5 Conclusion and Future Work

In the present paper we propose an extension of our previous graph edit distance approximation algorithm (BP). The major idea of our work is to use the suboptimal graph edit distance and the underlying node assignment in a genetic search procedure to improve the approximation accuracy. With several experimental results we show that this extension leads to a substantial reduction of the overestimations typical for BP. Though the run times are increased when compared to our former framework (as expected), they are still far below the run times of the exact algorithm.

We see three important lines of research for future work. First, we want to implement other search methods than genetic algorithms (e.g. floating search [20]). Second, there seems to be room for developing other merging methods to build a new assignment based on two given assignments (without the need to compute optimal assignments based on $\mathbf{C}$). Finally, the experimental evaluation will be extended (more data sets, more exhaustive evaluations of the meta parameters, etc.).

Acknowledgements. This work has been supported by the *Hasler Foundation* Switzerland and the *Swiss National Science Foundation* project P300P2-151279.

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. Vento, M.: A one hour trip in the world of graphs, looking at the papers of the last ten years. In: Kropatsch, W.G., Artner, N.M., Haxhimusa, Y., Jiang, X. (eds.) GbRPR 2013. LNCS, vol. 7877, pp. 1–10. Springer, Heidelberg (2013)
3. Bunke, H., Allermann, G.: Inexact graph matching for structural pattern recognition. Pattern Recognition Letters 1, 245–253 (1983)
4. Neuhaus, M., Bunke, H.: An error-tolerant approximate matching algorithm for attributed planar graphs and its application to fingerprint classification. In: Fred, A., Caelli, T.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.) SSPR&SPR 2004. LNCS, vol. 3138, pp. 180–189. Springer, Heidelberg (2004)
5. Ambauen, R., Fischer, S., Bunke, H.: Graph edit distance with node splitting and merging and its application to diatom identification. In: Hancock, E., Vento, M. (eds.) GbRPR 2003. LNCS, vol. 2726, pp. 95–106. Springer, Heidelberg (2003)
6. Robles-Kelly, A., Hancock, E.: Graph edit distance from spectral seriation. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(3), 365–378 (2005)
7. Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions of Systems, Science, and Cybernetics 4(2), 100–107 (1968)

8. Boeres, M.C., Ribeiro, C.C., Bloch, I.: A randomized heuristic for scene recognition by graph matching. In: Ribeiro, C.C., Martins, S.L. (eds.) WEA 2004. LNCS, vol. 3059, pp. 100–113. Springer, Heidelberg (2004)
9. Sorlin, S., Solnon, C.: Reactive tabu search for measuring graph similarity. In: Brun, L., Vento, M. (eds.) GbRPR 2005. LNCS, vol. 3434, pp. 172–182. Springer, Heidelberg (2005)
10. Justice, D., Hero, A.: A binary linear programming formulation of the graph edit distance. IEEE Trans. on Pattern Analysis ans Machine Intelligence 28(8), 1200–1214 (2006)
11. 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&SPR 2006. LNCS, vol. 4109, pp. 163–172. Springer, Heidelberg (2006)
12. Riesen, K., Bunke, H.: Approximate graph edit distance computation by means of bipartite graph matching. Image and Vision Computing 27(4), 950–959 (2009)
13. Cross, A., Wilson, R., Hancock, E.: Inexact graph matching using genetic search. Pattern Recognition 30(6), 953–970 (1997)
14. Wang, I., Fan, K.C., Horng, J.T.: Genetic-based search for error-correcting graph isomorphism. IEEE Transactions on Systems, Man, and Cybernetics (Part B) 27(4), 588–597 (1997)
15. Suganthan, P.: Structural pattern recognition using genetic algorithms. Pattern Recognition 35(9), 1883–1893 (2002)
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.) SSPR&SPR 2008. LNCS, vol. 5342, pp. 287–297. Springer, Heidelberg (2008)
17. DTP, AIDS antiviral screen (2004), `http://dtp.nci.nih.gov/docs/aids/aids_data.html`
18. Watson, C., Wilson, C.: NIST Special Database 4, Fingerprint Database. National Institute of Standards and Technology (March 1992)
19. 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)
20. Pudil, P., Novovicova, J., Kittler, J.: Floating search methods in feature-selection. Pattern Recognition Letters 15(11), 1119–1125 (1994)

Approximate Graph Edit Distance Guided by Bipartite Matching of Bags of Walks

Benoit Gaüzère[1], Sébastien Bougleux[2], Kaspar Riesen[3,⋆], and Luc Brun[1]

[1] ENSICAEN, GREYC CNRS UMR 6072, France
{benoit.gauzere,luc.brun}@ensicaen.fr
[2] Université de Caen Basse-Normandie, GREYC CNRS UMR 6072, France
bougleux@unicaen.fr
[3] University of Applied Sciences and Arts Northwestern Switzerland
kaspar.riesen@fhnw.ch

Abstract. The definition of efficient similarity or dissimilarity measures between graphs is a key problem in structural pattern recognition. This problem is nicely addressed by the graph edit distance, which constitutes one of the most flexible graph dissimilarity measure in this field. Unfortunately, the computation of an exact graph edit distance is known to be exponential in the number of nodes. In the early beginning of this decade, an efficient heuristic based on a bipartite assignment algorithm has been proposed to find efficiently a suboptimal solution. This heuristic based on an optimal matching of nodes' neighborhood provides a good approximation of the exact edit distance for graphs with a large number of different labels and a high density. Unfortunately, this heuristic works poorly on unlabeled graphs or graphs with a poor diversity of neighborhoods. In this work we propose to extend this heuristic by considering a mapping of bags of walks centered on each node of both graphs.

1 Introduction

Graphs provide a generic data structure which allows to encode fine properties of a large variety of objects such as shapes or molecules. The use of a graph representation to address pattern recognition problems implies to define a similarity measure between graphs. A widely used approach consists in using the graph edit distance, which allows to measure the distortion required to transform one graph into another. The distortion between two graphs G and G' can be encoded by an edit path defined as a sequence of operations transforming G into G'. Such a sequence may include node or edge insertions, removals and substitutions. Given a non-negative cost function $c(.)$, associated to each operation, the cost of an edit path is defined as the sum of its elementary operation's costs. The optimal edit path is defined as the one associated to the minimal cost among all edit paths transforming G into G'. This minimal cost then corresponds to the edit distance between G and G'. Unfortunately, beside its appealing properties, the computational time of the graph edit distance is known to grow exponentially with

⋆ Kaspar Riesen is supported by the Hasler Foundation Switzerland.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 73–82, 2014.

the number of implied nodes [9,2]. A close relationship exists between graph edit distance and morphism between graphs. Indeed, Bunke [1] has shown that under special conditions on the costs of node and edge insertions, removals and substitutions, computing the graph edit distance is equivalent to compute a maximum common subgraph of two graphs. More generally any mapping between the set of nodes and edges of two graphs induces an edit path which substitutes all mapped nodes and edges, and inserts or removes the non-mapped nodes/edges of the two graphs. Conversely, given an edit path between two graphs such that each node and each edge is substituted only once, one can define a mapping between the substituted nodes and edges of both graphs.

This close relationship between mappings and edit distance constitutes the main principle of the heuristic proposed by Riesen and Bunke [7] in order to decrease the exponential growth of the computational cost of the graph edit distance according to the number of considered nodes. This heuristic builds a mapping between the node sets of two graphs using a bipartite assignment algorithm, and deduces an edit path from this mapping. The cost of this edit path, which may not be optimal, is considered as an approximation of the exact edit distance. The optimal bipartite assignment algorithm is based on a cost function defined between the neighborhoods of each pair of nodes of the two graphs. The idea behind this heuristic being that a mapping between nodes with similar neighborhoods should induce an edit path with a low cost. However, this heuristic may work poorly on unlabeled graphs and more generally in cases where neighborhoods do not allow to easily differentiate the nodes.

In this paper we propose to extend this heuristic by considering a bipartite assignment algorithm between bags of walks incident to each node of both graphs. Hence, within this framework, a mapping of direct neighborhoods is similar to a mapping of bags of walks of length 1.

Our paper is structured as follows: Section 2 defines the bipartite assignment problem, and Section 3 defines the computation of an approximate edit distance from a bipartite assignment algorithm together with the heuristic defined in [7]. Section 4 defines an efficient computation of the bag of walks associated to each node of a graph, together with the costs of substituting, inserting or removing such a bag. Finally, Section 5 presents experiments on molecule datasets showing the accuracy gain obtained using our approach.

2 Assignment Problem

2.1 Linear Sum Assignment Problem (LSAP)

Let $\mathcal{X} = \{x_i\}_i$ and $\mathcal{Y} = \{y_i\}_i$ be two sets with $|\mathcal{X}| = |\mathcal{Y}| = n$. Assigning the n elements of $\mathcal{X}$ to the n elements of $\mathcal{Y}$ can be described by a bijective mapping $\mathcal{X} \to \mathcal{Y}$, reduced to a permutation of $\{1, \ldots, n\}$ if indices of elements are considered. Provided a matrix $\mathbf{C} \in \mathbb{R}_+^{n \times n}$ so that $C_{i,j} = c(x_i \to y_j) = c(y_j \to x_i)$ measures the cost of assigning element $x_i \in \mathcal{X}$ to element $y_j \in \mathcal{Y}$, the Linear Sum Assignment Problem (LSAP) finds an optimal permutation $\hat{\varphi} \in \underset{\varphi \in S_n}{\operatorname{argmin}} \sum_{i=1}^{n} C_{i,\varphi(i)}$, where S_n is

the set of all permutations of $\{1,\ldots,n\}$. Recall that any permutation φ can be associated to a permutation matrix $\mathbf{P}\in\{0,1\}^{n\times n}$ satisfying $P_{i,j}=\delta_{i,\varphi(i)}$, where $\delta_{i,j}$ is the Kronecker delta ($\delta_{i,j}=1$ if $i=j$ and 0 else). Note that $\mathbf{P}$ is doubly stochastic (sum of rows is equal to 1 and similarly for columns). Then, the LSAP corresponds to find an optimal permutation matrix

$$\hat{\mathbf{P}}\in\underset{\mathbf{P}\in\mathcal{P}_n}{\operatorname{argmin}}\ \sum_{i=1}^{n}\sum_{j=1}^{n}C_{i,j}\,P_{i,j}, \tag{1}$$

where $\mathcal{P}_n$ denotes the set of all $n\times n$ permutation matrices.

The LSAP may also be formulated as a maximization problem, and is also known as the maximum weighted bipartite matching problem. It can be solved by the Hungarian or Kuhn-Munkres algorithm in $O(n^3)$ time complexity [5,6], and it has been generalized in many directions, see [3] for more details.

2.2 LSAP with Insertion and Removal of Elements

Let $\mathcal{X}$ and $\mathcal{Y}$ be two sets, with $n=|\mathcal{X}|$ and $m=|\mathcal{Y}|$. As before, each element $x_i\in\mathcal{X}$ can be assigned to an element $y_j\in\mathcal{Y}$ according to a given substitution cost matrix $\mathbf{C}(\mathcal{X},\mathcal{Y})\in\mathbb{R}_+^{n\times m}$ with $[\mathbf{C}(\mathcal{X},\mathcal{Y})]_{i,j}=c(x_i\to y_j)$. Also, assume that each element of both $\mathcal{X}$ and $\mathcal{Y}$ can be deleted, or equivalently inserted, that is assigned to the null element denoted by ϵ. Removal and insertion of an element $x_i\in\mathcal{X}$ have the same cost $c(x_i\to\epsilon)=c(\epsilon\to x_i)$, and similarly for the elements of $\mathcal{Y}$. Removal-insertion costs associated to the n elements of $\mathcal{X}$ can be represented by the matrix $\mathbf{C}_\epsilon(\mathcal{X})\in\mathbb{R}^{n\times n}$, with $[\mathbf{C}_\epsilon(\mathcal{X})]_{i,j}=c(x_i\to\epsilon)$ if $i=j$ and $+\infty$ else. Similarly consider $\mathbf{C}_\epsilon(\mathcal{Y})\in\mathbb{R}^{m\times m}$. In other terms, each set is augmented with null elements, $\mathcal{X}_\epsilon=\mathcal{X}\cup\{\epsilon_i\}_{i=1,\ldots,m}$ and $\mathcal{Y}_\epsilon=\mathcal{Y}\cup\{\epsilon_i\}_{i=1,\ldots,n}$, such that $|\mathcal{X}_\epsilon|=|\mathcal{Y}_\epsilon|=n+m$. Following [7], the optimal linear sum assignment $\mathcal{X}_\epsilon\to\mathcal{Y}_\epsilon$, according to the cost matrix

$$\mathbf{C}_\epsilon(\mathcal{X},\mathcal{Y})=[C_{i,j}]_{i,j}=\begin{bmatrix}\mathbf{C}(\mathcal{X},\mathcal{Y}) & \mathbf{C}_\epsilon(\mathcal{X})\\ \mathbf{C}_\epsilon(\mathcal{Y}) & \mathbf{0}\end{bmatrix}\in[0,+\infty]^{(n+m)\times(n+m)}, \tag{2}$$

substitutes at most $\min(n,m)$ elements of $\mathcal{X}$ to at most $\min(n,m)$ elements of $\mathcal{Y}$, with insertion or removal of the remaining ones. Since the substitution of empty elements should not cause any cost, we always have $c(\epsilon_i\to\epsilon_j)=0$ (lower right submatrix of $\mathbf{C}_\epsilon(\mathcal{X},\mathcal{Y})$). An optimal assignment $\mathcal{X}_\epsilon\to\mathcal{Y}_\epsilon$ can then be defined as a matrix $\mathbf{P}$ minimizing the total cost functional

$$A(\mathbf{C}_\epsilon(\mathcal{X},\mathcal{Y}),\mathbf{P})=\underbrace{\sum_{i=1}^{n}\sum_{j=1}^{m}C_{i,j}\,P_{i,j}}_{\text{substitution}}+\underbrace{\sum_{i=1}^{n}C_{i,m+i}\,P_{i,m+i}+\sum_{j=1}^{m}C_{n+j,j}\,P_{n+j,j}}_{\text{removal/insertions}} \tag{3}$$

among the set $\mathcal{P}_{n,m,\epsilon}$ of all doubly substochastic matrices

$$\mathbf{P}=\begin{bmatrix}\mathbf{Q} & \mathbf{R}\\ \mathbf{S} & \mathbf{0}\end{bmatrix}\in\{0,1\}^{(n+m)\times(n+m)},$$

where $\mathbf{Q} \in \{0,1\}^{n \times m}$ represents the (partial) assignment $\mathcal{X} \to \mathcal{Y}$, and $\mathbf{R} \in \{0,1\}^{n \times n}$ and $\mathbf{S} \in \{0,1\}^{m \times m}$ are diagonal matrices representing removal and insertions. Columns and rows of $\mathbf{P}$ are constrained to satisfy

$$P_{i+j,j} + \sum_{i=1}^{n} P_{i,j} = 1, \ \forall j = 1, \ldots, m, \quad \text{and} \quad P_{i,j+i} + \sum_{j=1}^{m} P_{i,j} = 1, \ \forall i = 1, \ldots, n.$$

According to Section 2.1, the computation cost of the assignment is $O((n+m)^3)$. This assignment problem with edition is used to design approximate graph edit distances, as described in the following section.

3 Approximate Graph Edit Distance Based on the LSAP

We consider simple labeled graphs denoted by $G = (V, E, \mu, \nu)$, where V is the finite set of nodes, $E \subset V \times V$ is the set of edges, $\mu : V \to \mathcal{L}_\mathcal{V}$ is the node labeling function, and $\nu : E \to \mathcal{L}_\mathcal{E}$ is the edge labeling function. $\mathcal{L}_\mathcal{V}$ and $\mathcal{L}_\mathcal{E}$ are label sets for both nodes and edges (e.g. the vector space $\mathbb{R}^n$ or a set of symbolic labels).

As mentioned in Section 1, a major drawback of graph edit distance is its computational complexity. In fact, the problem of finding the minimum cost edit path between G and G' can be reformulated as an instance of a *Quadratic Assignment Problem* (*QAP*), known to be $\mathcal{NP}$*-complete*. Hence, exact computation of the graph edit distance is limited to graphs of rather small size in practice.

3.1 Graph Edit Distance Approximation

The graph edit distance approximation framework introduced in [7] reduces the QAP of graph edit distance computation to an instance of an LSAP which can be, in contrast with QAPs, efficiently solved. The algorithmic framework mainly consists of the following three steps.

Step 1. First, the graphs to be matched are subdivided into individual nodes plus local structures whereon a cost matrix $\mathbf{C}_\epsilon$, as defined in Eq. (2), is built.

Formally, let us consider an input graph $G = (V, E, \mu, \nu)$ together with a bag of bags of structural patterns $B = \{B_i\}_{i=1,\ldots,|V|}$. Every bag B_i is associated to a node $u_i \in V$ and characterizes the local structure of G around node u_i. The target graph $G' = (V', E', \mu', \nu')$ and its corresponding bags of structural patterns $B' = \{B'_i\}_{i=1,\ldots,|V'|}$ are given analogously. We define a cost $c(B_i \to B'_j)$ for the substitution of two bags of patterns, and a cost $c(B_i \to \epsilon)$ as well as a cost $c(\epsilon \to B'_j)$ for the removal and insertion of a bag, respectively. Given the cost model and following the scheme outlined in Section 2 we build the cost matrix $\mathbf{C}_\epsilon(B, B')$, encoding the cost of substitutions, insertions, and removals of bags of structural patterns.

Step 2. In the second step of the approximation framework, an assignment algorithm is applied to the square cost matrix $\mathbf{C}_\epsilon(B, B')$ in order to find a minimum cost assignment between both set of bags (possibly including removals and/or insertions of bags):

$$\hat{\mathbf{P}} \in \underset{\mathbf{P} \in \mathcal{P}_{|B|,|B'|,\epsilon}}{\operatorname{argmin}} \; A\left(\mathbf{C}_\epsilon(B, B'), \mathbf{P}\right). \tag{4}$$

Note that each bag B_i is associated to a single node u_i, and therefore, the optimal assignment defined by Eq. (4) provides an optimal assignment between the nodes of both graphs with respect to their bags of patterns. That is, the permutation $\hat{\mathbf{P}}$ provides a mapping $\psi : V \cup \{\varepsilon\} \to V' \cup \{\varepsilon\}$ of the nodes V of G to the nodes V' of G'. Due to the definition of the cost matrix, which allows both insertions and removals of elements, the mapping ψ includes node assignments of the form $u_i \to u'_j$, $u_i \to \varepsilon$, $\varepsilon \to u'_j$, and $\varepsilon \to \varepsilon$.

Step 3. Clearly, the mapping ψ can be interpreted as a partial edit path between the graphs G and G' considering edit operations on nodes only. Thus, in the last step this partial edit path is completed with respect to the edges. This can be accomplished since edit operations on edges are implied by edit operations on their nodes. That is, whether an edge is substituted, removed, or inserted, depends on the edit operations performed on its nodes. Hence, given the set of node operations in ψ, the global edge structures from G and G' can be edited accordingly. The cost of the complete edit path is finally returned as an approximate graph edit distance between graphs G and G'.

3.2 Defining Bags of Structural Patterns

Note that the edit path corresponding to the approximate edit distance value considers the edge structure of G and G' in a global and consistent way while the optimal permutation $\hat{\mathbf{P}}$ is able to consider the structural information in an isolated way only (bags of local structural patterns). This is due to the fact that during the optimization process of the specific LSAP, no information about neighboring node mappings is available. Hence, the definition of powerful structural patterns is a crucial task in this approximation framework.

In [7], every bag B_i of structural patterns represents the set of edges incident to node $v_i \in V$. Formally, assume that node v_i has incident edges E_{v_i}, then we define $B_i = \{\, (v_i, v_k) \in E_{v_i} : v_k \in V \,\}$. The present paper introduces a major generalization of this formalism. That is, rather than "the star neighborhood" of every node, bags of walks centered on each node are considered as bags of structural patterns. Both the computation of these bags of walks and the definition of an adequate cost model on them are described in the next section.

4 Walks and Approximate GED for Labeled Graphs

Recall that a walk of length k in a simple graph $G = (V, E, \mu, \nu)$, or k-walk, is a sequence $(u_i)_i$ of $(k+1)$ nodes of V such that $(u_i, u_{i+1}) \in E$ for all $i = 1, \ldots, k$. Any k-walk $(u_i)_i$, in a labeled graph, can be associated to a sequence

$$s = (s_l)_l = (\mu(u_0)\,\nu(u_0,u_1)\,\mu(u_1)\,\nu(u_1,u_2)\,\cdots\,\mu(u_{k-1})\,\nu(u_{k-1},u_k)\,\mu(u_k))$$

of $(2k+1)$ labels, alternating node and edge labels. Let B_i be the bag of sequences of $(2k+1)$ labels associated to all k-walks starting at node $v_i \in V$. Now given two graphs G and G', together with their bags $B = \{B_i\}_{i=1,\ldots,|V|}$ and $B' = \{B'_i\}_{i=1,\ldots,|V'|}$ of bags of label sequences, for each pair of bags $(B_i, B'_j) \in B \times B'$, the substitution cost $c(B_i \to B'_j)$ can be defined by comparing the label sequences. This is equivalent to the comparison of two bags of labeled k-walks, starting at nodes v_i and v'_j respectively.

By assuming that the substitution of node or edge labels does not depend on the labels themselves when they are different, the edit cost between two sequences $s \in B_i$ and $s' \in B'_j$ can simply be defined from the number of common labels at the same position in both sequences:

$$c(s \to s') = c_{\text{ns}} \sum_{l=0}^{k} \delta_{2l+1} + c_{\text{es}} \sum_{l=1}^{k} \delta_{2l}, \tag{5}$$

where $\delta_l = 0$ if $s_l = s'_l$ and 1 else, and c_{ns} and c_{es} denote node and edge substitution costs, respectively. When $s = s'$, the associated k-walks are equivalent, or similar, and $c(s \to s') = 0$. In other cases, different labels at the same position in s and s' appear at least once, the k-walks are said to be different. Since to compute this cost, k-walks needs to be explicitly extracted, it is difficult to derive a cost between bags which is computationally attractive. So we propose to restrict the knowledge of each k-walk to its terminal nodes (begin and end nodes), together with their labels, which allows to consider the cost

$$\hat{c}(s \to s') = \begin{cases} 0 & \text{if } s = s' \\ (\delta_1 + \delta_{2k+1} + k - 1)\, c_{\text{ns}} + k\, c_{\text{es}} & \text{else,} \end{cases} \tag{6}$$

so that non-terminal node labels and also edge labels are treated as if they were pairwise different when sequences are different. Obviously the cost $\hat{c}$ satisfies $c(s \to s') \leqslant \hat{c}(s \to s')$ for any s and s'.

Any optimal mapping between the walks of two bags according to Eq. (6) should include a mapping of similar walks with 0 cost. The cost of an optimal mapping between two bags of walks may thus be rewritten as:

$$[\mathbf{C}(B,B')]_{i,j} = 0 \cdot |B_i \cap B'_j| + \min_{\mathbf{P} \in \mathcal{P}_{|B_i \setminus B'_j|,|B'_j \setminus B_i|,\epsilon}} A\left(\mathbf{C}_\epsilon(B_i \setminus B'_j, B'_j \setminus B_i), \mathbf{P}\right), \tag{7}$$

which separates similar and different k-walks. Determining if k-walks (sequences) are similar can be achieved through the construction of the direct product of the two corresponding labeled graphs (Section 4.1). This also allows to derive assignment costs for the remaining different k-walks (Section 4.2).

4.1 Similar Walks

The direct product of two labeled graphs $G = (V, E, \mu, \nu)$ and $G' = (V', E', \mu', \nu')$ is the graph $G \times G' = (V_\times, E_\times, \mu_\times, \nu_\times)$. The node set and the edge set are given by $V_\times = \left\{(v_i, v'_j) \in V \times V' : \mu(v_i) = \mu'(v'_j)\right\}$ and $E_\times$, where $E_\times$ is defined by

$$\left\{\left((v_i, v'_j), (v_k, v'_l)\right) \in V_\times \times V_\times : (v_i, v_k) \in E \wedge (v'_j, v'_l) \in E' \wedge \nu(v_i, v_k) = \nu'(v'_j, v'_l)\right\}$$

such that $\mu_\times((v_i, v'_j)) = \mu(v_i) = \mu'(v'_j)$ for all node $(v_i, v'_j) \in V_\times$, and similarly $\nu_\times((v_i, v'_j), (v_k, v'_l)) = \nu(v_i, v_k) = \nu'(v'_j, v'_l)$ for all edge $((v_i, v'_j), (v_k, v'_l)) \in E_\times$. In particular, a walk from node (v_i, v'_j) to node (v_k, v'_l) in $G \times G'$ corresponds to a walk from v_i to v_k in G, and to a similar walk from v'_j to v'_l in G', both having the same sequence of node and edge labels by construction ([4] for an overview). This allows to partially match the two bags with a zero cost according to Eq. (7). Recall that the number of k-walks, between any pair of nodes of a graph, can be computed by $\mathbf{W}^k$, where $\mathbf{W}$ is the adjacency matrix of the graph. So, the number of k-walks common to the two graphs G and G' can be deduced from $\mathbf{W}^k_\times$, where $\mathbf{W}_\times$ defines the adjacency matrix of the direct product graph. Note that a walk in G similar to p walks in G' will be duplicated p times in the direct graph product.

4.2 Different Walks

Given a k value, and two different k-walks s and s', $c(s, s')$ can only take four different values depending on the values of δ_1 and δ_{2k+1}. This last point drastically simplifies the optimal assignment of the bags B_i and B'_j defined by Eq. (7), which can be efficiently approximated through histograms encoding terminal node's labels of sequences.

Let $h_i : \mathcal{L}_\mathcal{V} \to \mathbb{N}$ be the histogram function which assigns to each label $l \in \mathcal{L}_\mathcal{V}$, the number of k-walks ending at a node of label l in the bag B_i. This number of k-walks can be efficiently computed using $\mathbf{W}^k$. Similarly consider histograms h'_j and $h^\times_{(i,j)}$. From the definition of the direct product, we have $h^\times_{(i,j)} = z_i\, z'_j$, where z_i (resp. z'_j) defines the number of k-walks in B_i (resp. B'_j), for each node label, which are similar to at least one k-walk in B'_j (resp. B_i). The number of k-walks, in each bag, which can be matched with 0 cost, is thus given by $\min\{z_i, z'_j\}$. The remaining k-walks in B_i is then given by $h_{i\setminus j} = h_i - \min\{z_i, z'_j\}$. Similarly we consider $h_{j\setminus i} = h_j - \min\{z_i, z'_j\}$. Since computing z_i and z'_j may be computationally costly using an implicit enumeration of walks, $h_{i\setminus j}$ is approximated by $\hat{h}_{i\setminus j} = h_i - \min\{h_i, h'_j, \lfloor (h^\times_{(i,j)})^{1/2} \rfloor\}$, and similarly for $h'_{j\setminus i}$. According to (6), the cost of assigning the bag B_i to the bag B'_j is finally given by:

$$\begin{aligned}
[\mathbf{C}(B, B')]_{i,j} = {} & ((\delta_1 + k - 1)\, c_{\text{ns}} + k\, c_{\text{es}}) \sum_{l=1}^{|\mathcal{L}_\mathcal{V}|} \min\left\{\hat{h}_{i\setminus j}(l), \hat{h}'_{j\setminus i}(l)\right\} \\
& + ((\delta_1 + k)\, c_{\text{ns}} + k\, c_{\text{es}}) \min\left\{r_{i,j}, r'_{j,i}\right\} \\
& + ((\delta_1 + k)\, c_{\text{nri}} + k\, c_{\text{eri}}) \left|r_{i,j} - r'_{j,i}\right|,
\end{aligned} \tag{8}$$

where c_{nri} and c_{eri} denote node and edge removal/insertion costs, and $r_{i,j}$ corresponds to the k-walks of B_i not similar to a k-walk of B'_j, and whose terminal nodes need also to be substituted:

$$r_{i,j} = \sum_{l=1}^{|\mathcal{L}_\mathcal{V}|} \hat{h}_{i\setminus j}(l) - \min\left\{\hat{h}_{i\setminus j}(l), \hat{h}'_{j\setminus i}(l)\right\}, \tag{9}$$

and similarly for $r'_{j,i}$. The first line of (8) corresponds to substituted k-walks ending with the same node label ($\delta_{2k+1} = 0$ in Eq. 6), the second line corresponds to substituted k-walks ending with a different node label ($\delta_{2k+1} = 1$ in Eq. (6)), and third line to the remaining k-walks to be removed/inserted. From Eq. (8) and Eq. (9), the cost of removing/inserting a bag, *i.e.* the cost of removing/inserting all its k-walks, is given by $[\mathbf{C}_\epsilon(B)]_{i,i} = ((k+1)\, c_{\text{nri}} + k\, c_{\text{eri}})\, |B_i|$, and $[\mathbf{C}_\epsilon(B')]_{i,i} = ((k+1)\, c_{\text{nri}} + k\, c_{\text{eri}})\, |B'_i|$. The costs given by (8) and this last equation allow to construct the cost matrix $\mathbf{C}_\epsilon(B, B')$ in order to build the optimal assignment of bags of walks B and B'. An efficient approximation of the GED is then deduced from this optimal assignment (Sec. 3.1).

5 Experiments

Our new heuristic to compute an approximate edit distance has been tested on 4 graph datasets[1] encoding molecular graphs. For all these experiments, insertion/removal costs have been arbitrarily set to 3 for both edges and nodes and substitution cost to 1 for edges and nodes, regardless of node's or edge's labels. Graphs included within the 4 datasets have different characteristics: Alkane and PAH are only composed of unlabeled graphs whereas MAO and Acyclic correspond to labeled graphs. In addition, Alkane and Acyclic correspond to acyclic graphs having a low number of nodes (8 to 9 nodes in average) whereas MAO and PAH correspond to larger cyclic graphs (about 20 nodes in average). Tables 1 and 2 show a comparison of the accuracy of our proposition with state of the art method [7] and exact edit distance. First, Table 1 shows the percentage of distance matrix entries corresponding to a gain (i.e. computed edit distance is lower), a loss or no changes on the accuracy of our approximation method versus the one proposed by [7]. As we can see in column "Gain", our approach provides a more accurate approximation of the edit distance for 45% to 98% of molecules' pairs while we observe a loss on the accuracy for only $< 1\%$ to 27% of computed edit distances, depending on the dataset. Same conclusions are observed in Table 2 which shows the average edit distance for each dataset and each method together with the average time required to compute the associated edit distance matrix. We can note that the time required to compute our edit distances is higher, but still comparable, than the one required by [7]. However, one can note that computation times obtained for the lines 1 and 2 have been computed using a Java implementation [8] whereas line 3 corresponds to a Matlab implementation. Finally, results for A* method for MAO and PAH datasets are not displayed since it takes too much time to compute. These first results allow us to highlight the gain on the accuracy induced by using our matching approach instead of the one initially proposed by [7]. In addition, we can note that taking

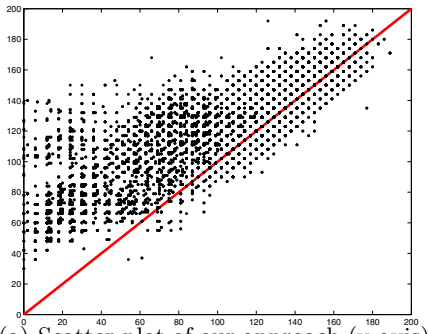
[1] These datasets are available at `http://iapr-tc15.greyc.fr/links.html`

Table 1. Accuracy comparison between our approach and [7]

Dataset	Gain	Loss	Equality	Size
Alkane	45%	28%	27%	3
PAH	73%	14%	13%	4
MAO	98%	2%	$< 1\%$	4
Acyclic	56%	24%	21%	4

Table 2. Average edit distance ($\bar{d}$) and average time in seconds ($\bar{t}$) for each method and each dataset (BW = bags of walks)

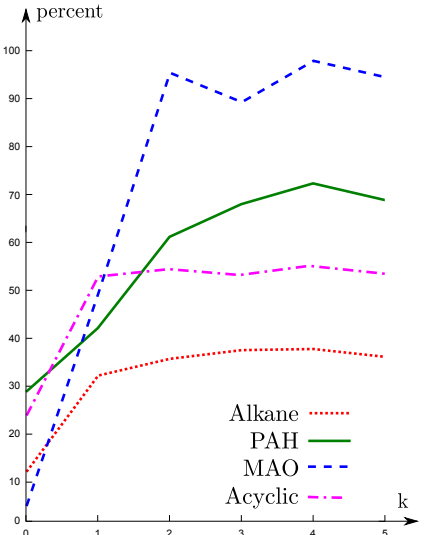
	Alkane		**Acyclic**		**MAO**		**PAH**	
	$\bar{d}$	$\bar{t}$	$\bar{d}$	$\bar{t}$	$\bar{d}$	$\bar{t}$	$\bar{d}$	$\bar{t}$
A*	15	28 800	17	172 800	-	-	-	-
[7]	35	20	35	22	105	20	138	40
BW	33	55	31	86	49	129	120	390



(a) Scatter-plot of our approach (x-axis) and [7] (y-axis)

Method	**MAO**			**PAH**		
k	**1**	**3**	**5**	**1**	**3**	**5**
[7]	68%	62%	54%	59%	63%	61%
BW	93%	90%	71%	79%	78%	74%

(b) Classification results



(c) Percentage of distance matrix's entries corresponding to an accuracy gain using our approach versus the size of considered walks (k) for each dataset

Fig. 1. Classification of MAO and PAH using k-ppv and walk size comparisons

into account a larger radius than the direct neighborhood (i.e. walk size > 1) allows us to increase the percentage of distance matrix's entries corresponding to an accuracy gain using our approximation, with maximum percentage obtained for walks of size equals to 3 or 4 (Figure 1(c)). However, we can note that the accuracy decreases when considering walks up to 5 nodes. This observation can be explained by the tottering phenomenon which induces non representative walks into the computation of the cost matrix. In addition, we can note that this observation is stronger for Acyclic and Alkane datasets which are more prone to tottering since they are both composed of smaller molecules than PAH and MAO.

In order to validate our proposition on prediction problems, we predicted the classes of PAH and MAO molecules thanks to a k-ppv algorithm, with k equals to 1, 3 and 5. Table in Figure 1(b) shows the percentage of correctly classified molecules using a 10-fold cross validation. As observed in previous experiments, the gain on the accuracy provided by our approximation (line 2, Table in Figure 1(b)) allows us to obtain significantly better classification results than the ones obtained by the approximation method proposed in [7] (line 1, Table in Figure 1(b)). This classification experiment shows thus the relevance of our contribution for prediction problems. This accuracy gain is also shown by the scatter plot of our approximation (x-axis) and the approximation of [7] (y-axis) on PAH and MAO datasets (Figure 1(a)). Points over the diagonal corresponds to a better accuracy of our approach than the one obtained by [7].

6 Conclusion

We have presented in this paper a natural extension of a well known heuristic computing an approximate graph edit distance between labeled graphs. Our heuristic is based on an assignment of bags of walks incident to each node. Experiments show that the proposed heuristic brings a significant decrease of the graph edit distance compared to the previous heuristic at a cost which remains much lower than the computational cost of the exact edit distance. Moreover, according to our experiments our heuristic provides a significant gain on classification results using a kppv classifier. Our future work will consist to test other types of patterns and to compare explicit vs implicit enumeration of patterns.

References

1. Bunke, H.: On a relation between graph edit distance and maximum common subgraph. Pattern Recognition Letters 18(9), 689–694 (1997)
2. Bunke, H., Allermann, G.: Inexact graph matching for structural pattern recognition. Pattern Recognition Letters 1, 245–253 (1983)
3. Burkard, R., Dell'Amico, M., Martello, S.: Assignment Problems. SIAM (2009)
4. Hammack, R., Imrich, W., Klavžar, S.: Hanbook of Product Graphs, 2nd edn. Discrete Mathematics and its Applications. CRC Press, Taylor & Francis (2011)
5. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics Quaterly 2, 83–97 (1955)
6. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics 5(1), 32–38 (1957)
7. Riesen, K., Bunke, H.: Approximate graph edit distance computation by means of bipartite graph matching. Image and Vision Computing 27, 950–959 (2009)
8. Riesen, K., Emmenegger, S., Bunke, H.: A novel software toolkit for graph edit distance computation. In: Kropatsch, W.G., Artner, N.M., Haxhimusa, Y., Jiang, X. (eds.) GbRPR 2013. LNCS, vol. 7877, pp. 142–151. Springer, Heidelberg (2013)
9. Sanfeliu, A., Fu, K.: A distance measure between attributed relational graphs for pattern recognition. Systems, Man and Cybernetics 13(3), 353–363 (1983)

A Hausdorff Heuristic for Efficient Computation of Graph Edit Distance

Andreas Fischer[1], Réjean Plamondon[1], Yvon Savaria[1],
Kaspar Riesen[2], and Horst Bunke[3]

[1] Département de Génie Électrique, École Polytechnique de Montréal
2900 boul. Édouard-Montpetit, Montréal, Québec H3T 1J4, Canada
{andreas.fischer,rejean.plamondon,yvon.savaria}@polymtl.ca
[2] Institute for Informations Systems, University of Applied Sciences and Arts
Northwestern Switzerland, Riggenbachstrasse 16, 4600 Olten, Switzerland
kaspar.riesen@fhnw.ch
[3] Institute of Computer Science and Applied Mathematics, University of Bern
Neubrückstrasse 10, 3012 Bern, Switzerland
bunke@iam.unibe.ch

Abstract. Graph edit distance is a flexible and powerful measure of dissimilarity between two arbitrarily labeled graphs. Yet its application is limited by the exponential time complexity involved when matching unconstrained graphs. We have recently proposed a quadratic-time approximation of graph edit distance based on Hausdorff matching, which underestimates the true distance. In order to implement verification systems for the approximation algorithm, efficiency improvements are needed for the computation of the true distance. In this paper, we propose a Hausdorff heuristic that employs the approximation algorithm itself as a heuristic function for efficient A* computation of the graph edit distance. In an experimental evaluation on several data sets of the IAM graph database, substantial search space reductions and runtime speedups of one order of magnitude are reported when compared with plain A* search.

Keywords: Graph matching, graph edit distance, A* search, Hausdorff distance.

1 Introduction

Graphs are one of the most general data structures in pattern recognition for representing objects. Individual parts of the objects are represented with nodes which are linked with edges to represent binary relationships. Both nodes and edges can be labeled with attributes, for instance in form of feature vectors. This high representational power of graphs has proven successful in pattern recognition and led to widespread applications [1, 2], for example in bioinformatics [3], image classification [4], and computer network analysis [5].

The complexity of the graph data structure usually leads to a high computational complexity when matching two objects. Therefore, many graph matching

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 83–92, 2014.

algorithms impose certain constraints on the graphs. For example spectral methods [6, 7], which are based on efficient eigendecomposition of the adjacency or Laplacian matrix of a graph, primarily target unlabeled graphs or allow only severely constrained label alphabets. Other examples include restrictions to ordered graphs [8] and graphs with unique node labels [9].

Graph edit distance (GED) [10] is a flexible measure of dissimilarity between two graphs, which is able to cope with unconstrained graphs. In particular, arbitrary labels are allowed on both nodes and edges. Originally proposed for string matching [11], the concept of edit distance is to apply a series of edit operations to one object in order to transform it into the other. The edit distance then corresponds with the minimum cost among all possible edit paths.

However, the flexibility of GED comes at the cost of an exponential time complexity with respect to the graph size. The search space of all possible edit paths is usually traversed with a best-first A* search [12]. By using a heuristic function to estimate the future cost of an incomplete edit path, the efficiency of the search procedure can be greatly improved [13–15] but the computational complexity remains the same.

In order to overcome the limitation of exponential time complexity, polynomial approximation of GED is a promising line of current research. In [16], the Hungarian algorithm [17] is used to obtain a cubic-time approximation of GED by assigning nodes and their local edge structure of one graph to nodes and their local edge structure of the other graph. Although only local structure is considered, a high approximation quality is achieved and a strong performance is reported for the task of pattern classification on different graph data sets [16].

Following the same idea of matching nodes and their local edge structure, we have recently proposed an even faster quadratic-time approximation of GED in [18, 19] based on Hausdorff matching [20]. Similar to the comparison of finite subsets of a metric space by means of Hausdorff distance, each node of one graph is compared with each node of the other graph only once to determine its best matching cost, hence the quadratic time complexity. As expected, the deviation from the true edit distance has proven to be larger when compared with the cubic-time approximation. Still, the proposed Hausdorff edit distance (HED) has achieved promising results for the task of pattern classification on diverse graph data sets [18, 19]. It combines the high flexibility of GED to cope with unconstrained graphs with a low quadratic time complexity, which makes HED applicable to a wide range of real-world applications.

So far, a direct comparison of the proposed approximation algorithms with the true edit distance could only be performed for relatively small graphs due to the exponential time complexity of GED. There is a need to improve the efficiency of GED for verification experiments, which can measure the approximation quality in the case of larger graphs observed in many real-world applications. As mentioned above, a common approach to improve the efficiency is the development of accurate heuristic functions for A* computation of GED, which can greatly reduce runtime and memory usage by avoiding a complete traversal of the exponential search space of all possible edit paths. Note that only

admissible heuristic functions, which underestimate the true edit distance, can be used for exact computation of GED. Suboptimal variants of A* search like Bayesian A* search [21, 22] do not guarantee a globally optimal solution. Instead, they are interesting for approximating GED as suggested in [23] for beam search and weighted path length search.

In this paper, we propose a Hausdorff heuristic based on HED to improve the efficiency of GED. Since HED underestimates the true edit distance, it is an admissible heuristic function for A* search. The performance of the proposed heuristic is experimentally evaluated on several data sets from the IAM graph database [24] and is compared with plain A* search. Substantial search space reductions and runtime speedups of one order of magnitude are reported.

The remainder of this paper is organized as follows. First, HED is reviewed in Section 2. Afterwards, the proposed Hausdorff heuristic is presented in Section 3 and experimental results are reported in Section 4. Finally, we draw conclusions in Section 5.

2 Hausdorff Edit Distance

In this section, we review the Hausdorff edit distance (HED) [18, 19]. After providing some basic definitions in Section 2.1, HED is defined in Section 2.2.

2.1 Basic Definitions

A *graph* g is a four-tuple $g = (V, E, \mu, \nu)$. V is the finite set of nodes, $E \subseteq V \times V$ is the set of edges, $\mu : V \to L_V$ is the node labeling function, and $\nu : E \to L_E$ is the edge labeling function. L_V and L_E are label sets for nodes and edges, for instance symbolic labels $\{\alpha, \beta, \gamma, \ldots\}$ or the vector space $\mathbb{R}^n$.

Given two graphs $g_1 = (V_1, E_1, \mu_1, \nu_1)$ and $g_2 = (V_2, E_2, \mu_2, \nu_2)$, *edit operations* transform nodes and edges of g_1 into nodes and edges of g_2. Three node edit operations are usually considered for $u \in V_1$ and $v \in V_2$, namely *substitutions* $(u \to v)$, *deletions* $(u \to \epsilon)$, and *insertions* $(\epsilon \to v)$. The same set of edit operations is considered for edges $p \in E_1$ and $q \in E_2$, *i.e.* substitutions $(p \to q)$, deletions $(p \to \epsilon)$, and insertions $(\epsilon \to q)$.

A *cost function* $\mathcal{C}$ assigns non-negative costs to node and edge edit operations. An example for Euclidean labels is the Euclidean cost function with substitution cost $\mathcal{C}(u \to v) = ||\mu_1(u) - \mu_2(v)||$ and $\mathcal{C}(p \to q) = ||\nu_1(p) - \nu_2(q)||$. The cost for deletion and insertion is often set to a constant value. Without loss of generality, we will assume $\mathcal{C}(u \to \epsilon) = \mathcal{C}(\epsilon \to v) = C_n$ and $\mathcal{C}(p \to \epsilon) = \mathcal{C}(\epsilon \to q) = C_e$ in the following for all types of cost functions.

2.2 HED Definition

The Hausdorff edit distance $HED(g_1, g_2, \mathcal{C})$ between two graphs g_1 and g_2 is defined with respect to the cost function $\mathcal{C}$ as follows:

$$HED(g_1, g_2, \mathcal{C}) = \sum_{u \in V_1} \min_{v \in V_2 \cup \{\epsilon\}} f(u, v, \mathcal{C}) + \sum_{v \in V_2} \min_{u \in V_1 \cup \{\epsilon\}} f(u, v, \mathcal{C}) \qquad (1)$$

It consists of two summation terms that each calculate nearest neighbor distances between the two node sets similar to the Hausdorff distance between finite subsets of a metric space. Nearest neighbors are determined with respect to the node function $f(u, v, \mathcal{C})$, which is defined as

$$f(u,v,\mathcal{C}) = \begin{cases} C_n + \sum_{i=1}^{|P|} \frac{C_e}{2} & \text{for node deletion } (u \to \epsilon) \\ C_n + \sum_{i=1}^{|Q|} \frac{C_e}{2} & \text{for node insertion } (\epsilon \to v) \\ \frac{\mathcal{C}(u \to v) + \frac{HED(P,Q,\mathcal{C})}{2}}{2} & \text{for node substitution } (u \to v) \end{cases} \tag{2}$$

where P is the set of edges adjacent to u and Q is the set of edges adjacent to v. In the case of deletions, the node deletion cost C_n and half of the implied edge deletion cost is accumulated. Node insertion costs are obtained accordingly. In the case of substitution, half of the substitution cost is considered, which itself consists of the node substitution cost $\mathcal{C}(u \to v)$ and half of the implied edge cost.

In order to obtain an estimate of the implied edge cost, the edge sets P and Q are matched in the same manner as the node sets, *i.e.* based on a Hausdorff edit distance

$$HED(P,Q,\mathcal{C}) = \sum_{p \in P} \min_{q \in Q \cup \{\epsilon\}} g(p,q,\mathcal{C}) + \sum_{q \in Q} \min_{p \in P \cup \{\epsilon\}} g(p,q,\mathcal{C}) \tag{3}$$

with the corresponding edge function

$$g(p,q,\mathcal{C}) = \begin{cases} C_e & \text{for edge deletion } (p \to \epsilon) \\ C_e & \text{for edge insertion } (\epsilon \to q) \\ \frac{\mathcal{C}(p \to q)}{2} & \text{for edge substitution } (p \to q) \end{cases} \tag{4}$$

The two divisions by 2 for node substitutions in Equation 2 ensure that HED approximates GED. Only half of the substitution cost is considered because the substitutions, which are not required to be bidirectional, appear in both summation terms in Equation 1. Only half of the implied edge cost is considered because each edge edit operation is implied by exactly two nodes. In effect, an optimal edit cost is assigned to each node without taking the assignments of the other nodes into account. Therefore HED is always less than or equal to GED.

In order to limit the underestimation, lower bounds are used with respect to the number of elements in the set. For $HED(g_1, g_2, \mathcal{C})$, we use a lower bound of $||V_1| - |V_2|| \cdot C_n$ and for $HED(P, Q, \mathcal{C})$, we use a lower bound of $||P| - |Q|| \cdot C_e$. For more details on HED, we refer to [18, 19].

3 Hausdorff Heuristic

In this section, a Hausdorff heuristic based on HED is presented for efficient A* computation of GED. First, GED computation is discussed in Section 3.1. Afterwards, the integration of HED as a heuristic into the A* search algorithm is detailed in Section 3.2.

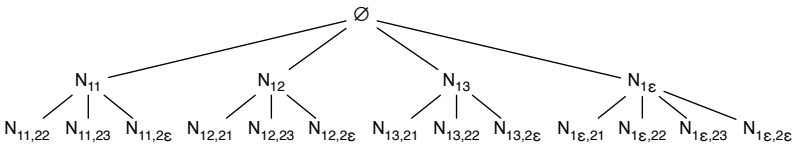


Fig. 1. GED search tree

3.1 GED Computation

The search space for GED is usually spanned by all possible node edit operations that transform V_1 into V_2. Edge edit operations are implied by the node edit operations as soon as both nodes of an edge in g_1 have been assigned to two nodes in g_2.

An example is shown in Figure 1 for $V_1 = \{u_1, u_2\}$ and $V_2 = \{v_1, v_2, v_3\}$. The root of the search tree is an empty node assignment $\emptyset$. At the first level, the first node of V_1 is either assigned to one of the nodes in V_2 or it is deleted. At the second level, the second node of V_1 is assigned in the same way considering all remaining nodes in V_2. At this leaf level, all remaining node insertions are added to the node assignment N. Clearly, the number of entries in the tree is exponential with respect to the number of nodes of the graphs.

Using A* best-first search, the non-expanded node assignments are kept in a sorted *open* list, which is ordered by the cost function

$$f(N) = g(N) + h(N) \tag{5}$$

where $g(N)$ is the cost of all current node edit operations and implied edge edit operations, and $h(N)$ is a heuristic function that estimates the future cost of the node assignment. Admissible heuristics are less than or equal to the real cost. At each step of the search, the currently best node assignment from *open* with the lowest cost function $f(N)$ is removed and its successors are added to *open*. As soon as the currently best node assignment is complete, *i.e.* it transforms V_1 into V_2, the cost of the assignment is returned as GED.

3.2 HED Heuristic

The proposed Hausdorff heuristic estimates the future cost of a node assignment N by means of HED. We consider the subgraph g_1' of g_1 that contains all free nodes $F_1 \subseteq V_1$ according to N and the subgraph g_2' of g_2 that contains all free nodes $F_2 \subseteq V_2$. Then, we calculate the heuristic function

$$h(N) = HED(g_1', g_2', \mathcal{C}) \tag{6}$$

with respect to the underlying cost function $\mathcal{C}$. Because HED underestimates the edit distance between g_1' and g_2', the heuristic function $h(N)$ underestimates the future cost of the node assignment N and is therefore an admissible heuristic for A* computation of GED.

Algorithm 1. Hausdorff heuristic

Require: graphs g_1, g_2, cost function $\mathcal{C}$, node assignment N
Ensure: minimum future cost c
1: **for all** free nodes $u \in F_1 \subseteq V_1$ according to N **do**
2: $c_1(u) \leftarrow f(u, \epsilon, \mathcal{C})$
3: **end for**
4: **for all** free nodes $v \in F_2 \subseteq V_2$ according to N **do**
5: $c_2(v) \leftarrow f(\epsilon, v, \mathcal{C})$
6: **end for**
7: **for all** nodes u in F_1 **do**
8: **for all** nodes v in F_2 **do**
9: $cost \leftarrow f(u, v, \mathcal{C})$
10: $c_1(u) \leftarrow \min(cost, c_1(u))$
11: $c_2(v) \leftarrow \min(cost, c_2(v))$
12: **end for**
13: **end for**
14: $cost \leftarrow \sum_{u \in F_1} c_1(u) + \sum_{v \in F_2} c_2(v)$
15: **return** $\max(cost, ||F_1| - |F_2|| \cdot C_n)$

A straight-forward computation of Equation 6 is detailed in Algorithm 1. First, the minimum edit costs of each free node $u \in F_1$ and $v \in F_2$ are calculated in lines 1-13. Then, line 14 performs the summation according to Equation 1 and line 15 applies the lower bound (see Section 2.2).

We would like to point out two implementation details, which are important for efficiency. First, the sorted *open* list of the A* search is implemented as a binary search tree that allows to add new elements with $O(\log(n))$ time complexity. Secondly, the function $f(u, v, \mathcal{C})$ from Equation 2 is independent from the node assignment N. Therefore it is pre-computed before the A* search is executed, which leads to an efficient quadratic time complexity of $O(|F_1| \cdot |F_2|)$ for Algorithm 1 with a low constant factor. In particular, the time complexity is independent from the number of edges adjacent to the nodes.

4 Experimental Evaluation

In this section, we report experimental results achieved with the proposed Hausdorff heuristic. The results are compared with plain A* search as a reference, *i.e.* GED computation with the trivial heuristic $h(N) = 0$.

In the following, the selection of suitable graphs is discussed in Section 4.1 and performance results are provided in Section 4.2.

4.1 Graph Selection

Several data sets from the IAM graph database [24] are considered that differ in object domain and graph structure. First, the *Letter I-III* data sets containing graphs of letter drawings, which are artificially distorted with three distortion degrees. Secondly, the *Fingerprints* data set containing graphs of fingerprints from

Table 1. Data set statistics. Number of selected graphs, median number of nodes and edges, minimum and maximum number of nodes.

Data Set	Graphs	$\|V\|_{med}$	$\|E\|_{med}$	$\|V\|_{min}$	$\|V\|_{max}$
Letters I	200	6	4	4	8
Letters II	200	6	4	4	9
Letters III	200	5	5	4	8
Fingerprints	186	4	6	2	8
Molecules I	95	8	7	4	9
Molecules II	103	9	8	3	9

Table 2. Search space reduction. Average size of the *open* list after A* search.

Data Set	Reference	Hausdorff Heuristic	Reduction Factor
Letters I	568.2	27.3	20.8
Letters II	2,829.0	300.3	9.4
Letters III	2,955.1	386.0	7.7
Fingerprints	6,282.3	2,104.8	3.0
Molecules I	61,994.3	6,094.3	10.2
Molecules II	111,852.1	18,911.7	5.9

the NIST-4 reference database [25]. Thirdly, the *Molecules I* data set based on molecular compounds from the Chemical Carcinogenesis Research Information System (CCRIS) database [26]. And finally, the *Molecules II* data set which is based on molecular compounds from the AIDS Antiviral Screen Database of Active Compounds [27]. Cost functions and their parameter values are adopted from previous work [16, 19].

Due to the exponential time complexity of GED, only relatively small graphs can be included in the evaluation. Besides the runtime, the required memory space is also a limiting factor since the size of the *open* list may grow exponentially during A* search. For each data set, we have selected the first n graphs with less than 10 nodes such that each computation of the $n \cdot n$ edit distances with plain A* search is feasible with less than one million node assignments. If possible $n = 200$ was chosen. Table 1 lists the resulting data set statistics.

4.2 Results

The performance results for search space reduction are listed in Table 2. The proposed Hausdorff heuristic is compared with plain A* search as a reference on the six graph data sets. The average number of elements in the *open* list after A* search is indicated. This number is directly related to the memory space required for GED. Despite the choice of rather small graphs for experimental evaluation, the average size of $111,852.1$ for the Molecules II data set is already very high with respect to the imposed limit of one million node assignments. The experiment was performed with 4GB RAM.

Table 3. Computational speedup. CPU runtime in seconds.

Data Set	Reference	Hausdorff Heuristic	Speedup Factor
Letters I	18.5	1.1	17.1
Letters II	109.7	11.2	9.8
Letters III	110.5	14.6	7.6
Fingerprints	196.4	61.3	3.2
Molecules I	523.5	36.8	14.2
Molecules II	1,224.6	165.3	7.4

The search space reduction achieved with the Hausdorff heuristic is about one order of magnitude in all cases. The best result is obtained on the Letters I data set, where 20.8 times less memory is required to compute GED. We assume that this large reduction factor is related to the suitability of the label domain for the Hausdorff heuristic. The Letters I data set contains weakly distorted drawings of letters whose line endings are labeled with their Cartesian coordinates. Using an Euclidean cost function, this type of node label is effective for selecting the nearest neighbor of $u \in V_1$ in V_2 and vice versa in Equation 1, even if the local edge structure is similar. Furthermore, we report strong results for molecular compounds using a Dirac cost function for matching chemical symbols. On the Molecules I data set, a reduction factor of 10.2 is reported.

The performance results for runtime reduction are provided in Table 3. Experiments were conducted with an Intel Core i7 processor with 2.0GHz CPU using a Java implementation. The runtime is indicated in seconds for matching all $n \cdot n$ selected graphs. The speedups are closely related to the search space reductions, which indicates that the computation of the Hausdorff heuristic does not lead to a significant overhead. In all cases, a CPU runtime reduction of one order of magnitude is obtained. The best result is achieved for the Letters I data set, where 17.1 times less CPU time is required to compute GED when using the Hausdorff heuristic.

5 Conclusions

In this paper, we have proposed a Hausdorff heuristic for efficient A* computation of graph edit distance (GED). The heuristic is based on the Hausdorff edit distance (HED), a quadratic-time approximation of GED, which underestimates the true distance and is hence admissible as a heuristic function for A* search. Based on a domain-specific cost function, the proposed Hausdorff heuristic can cope with unconstrained graphs. In particular, arbitrary labels are allowed on both nodes and edges.

An experimental evaluation is reported for six data sets from the IAM graph database. In all cases, the proposed heuristic has achieved substantial reductions of one order of magnitude in memory space and runtime. The best performance is reported for graphs from the Letters I data set, where 20.8 times less memory

and 17.1 times less CPU time were required to compute GED when using the Hausdorff heuristic instead of plain A* search.

In future work, we aim to include larger graphs in the experimental evaluation and to compare and combine different heuristic functions. Our overall aim is to implement an efficient verification system that is able to calculate the true edit distance for large graphs in order to evaluate the approximation quality of HED and other GED approximations. We expect that in addition to software acceleration, a massive hardware acceleration will be necessary to compute GED for larger graphs.

Acknowledgments. This work has been supported by the SNSF grant P300P2-151279 to A. Fischer, the NSERC grant RGPIN-915 to R. Plamondon, a Canada Research Chair grant to Y. Savaria, and a grant from the Hasler Foundation Switzerland to K. Riesen.

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. Vento, M.: A one hour trip in the world of graphs, looking at the papers of the last ten years. In: Kropatsch, W.G., Artner, N.M., Haxhimusa, Y., Jiang, X. (eds.) GbRPR 2013. LNCS, vol. 7877, pp. 1–10. Springer, Heidelberg (2013)
3. Borgwardt, K., Ong, C., Schönauer, S., Vishwanathan, S., Smola, A., Kriegel, H.P.: Protein function prediction via graph kernels. Bioinformatics 21(1), 47–56 (2005)
4. Harchaoui, Z., Bach, F.: Image classification with segmentation graph kernels. In: Proc. Int. Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
5. Shoubridge, P., Kraetzl, M., Wallis, W.D., Bunke, H.: Detection of abnormal change in time series of graphs. Journal of Interconnection Networks 3(1-2), 85–101 (2002)
6. Umeyama, S.: An eigendecomposition approach to weighted graph matching problems. IEEE Trans. on Pattern Analysis and Machine Intelligence 10(5), 695–703 (1988)
7. Wilson, R., Hancock, E., Luo, B.: Pattern vectors from algebraic graph theory. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(7), 1112–1124 (2005)
8. Jiang, X., Bunke, H.: Optimal quadratic-time isomorphism of ordered graphs. Pattern Recognition 32(17), 1273–1283 (1999)
9. Dickinson, P., Bunke, H., Dadej, A., Kraetzl, M.: Matching graphs with unique node labels. Pattern Analysis and Applications 7(3), 243–254 (2004)
10. Sanfeliu, A., Fu, K.S.: A distance measure between attributed relational graphs for pattern recognition. IEEE Trans. on Systems, Man, and Cybernetics 13(3), 353–363 (1983)
11. Wagner, R.A., Fischer, M.J.: The string-to-string correction problem. Journal of the Association for Computing Machinery 21(1), 168–173 (1974)
12. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. on Systems, Science, and Cybernetics 4(2), 100–107 (1968)

13. Berretti, S., Del Bimbo, A., Vicario, E.: Efficient matching and indexing of graph models in content-based retrieval. IEEE Trans. on Pattern Analysis and Machine Intelligence 23(10), 1089–1105 (2001)
14. Gregory, L., Kittler, J.: Using graph search techniques for contextual colour retrieval. In: Caelli, T.M., Amin, A., Duin, R.P.W., Kamel, M.S., de Ridder, D. (eds.) SSPR&SPR 2002. LNCS, vol. 2396, pp. 186–194. Springer, Heidelberg (2002)
15. Riesen, K., Fankhauser, S., Bunke, H.: Speeding up graph edit distance computation with a bipartite heuristic. In: Proc. Int. Workshop on Mining and Learning with Graphs, pp. 21–24 (2007)
16. Riesen, K., Bunke, H.: Approximate graph edit distance computation by means of bipartite graph matching. Image and Vision Computing 27(4), 950–959 (2009)
17. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics 5(1), 32–38 (1957)
18. Fischer, A., Suen, C.Y., Frinken, V., Riesen, K., Bunke, H.: A fast matching algorithm for graph-based handwriting recognition. In: Kropatsch, W.G., Artner, N.M., Haxhimusa, Y., Jiang, X. (eds.) GbRPR 2013. LNCS, vol. 7877, pp. 194–203. Springer, Heidelberg (2013)
19. Fischer, A., Suen, C., Frinken, V., Riesen, K., Bunke, H.: Approximation of graph edit distance based on Hausdorff matching. Pattern Recognition (submitted)
20. Huttenlocher, D.P., Klanderman, G.A., Kl, G.A., Rucklidge, W.J.: Comparing images using the Hausdorff distance. IEEE Trans. on Pattern Analysis and Machine Intelligence 15, 850–863 (1993)
21. Coughlan, J.M., Yuille, A.L.: Bayesian A* tree search with expected O(N) node expansions: applications to road tracking. Neural Computation 14(8), 1929–1958 (2002)
22. Cazorla, M., Escolano, F., Gallardo, D., Rizo, R.: Junction detection and grouping with probabilistic edge models and Bayesian A*. Pattern Recognition 35(9), 1869–1881 (2002)
23. 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&SPR 2006. LNCS, vol. 4109, pp. 163–172. Springer, Heidelberg (2006)
24. 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.) SSPR&SPR 2008. LNCS, vol. 5342, pp. 287–297. Springer, Heidelberg (2008)
25. Watson, C., Wilson, C.: NIST Special Database 4, Fingerprint Database. National Institute of Standards and Technology (1992)
26. Kazius, J., McGuire, R., Bursi, R.: Derivation and validation of toxicophores for mutagenicity prediction. Journal of Medicinal Chemistry 48(1), 312–320 (2005)
27. DTP: AIDS antiviral screen (2004), `http://dtp.nci.nih.gov/docs/aids/aids_data.html`

Flip-Flop Sublinear Models for Graphs

Brijnesh Jain

Technische Universität Berlin, Germany
brijnesh.jain@gmail.com

Abstract. Extending linear classifiers from feature vectors to attributed graphs results in sublinear classifiers. In contrast to linear models, the classification performance of sublinear models depends on our choice as to which class we label as positive and which as negative. We prove that the expected classification accuracy of sublinear models may differ for different class labelings. Experiments confirm this finding for empirical classification accuracies on small samples. These results give rise to flip-flop sublinear classifiers that consider both class labelings during training and select the model for prediction that better fits the training data.

Keywords: graph matching, classification, perceptron learning.

1 Introduction

Linear models are one of the most simple prediction methods that make strong assumptions about the structure of the underlying data and yields stable, but possibly inaccurate predictions [4]. In addition, linear methods form a basis for understanding and devising nonlinear ones.

Application of linear methods, however, is confined to real-valued feature vectors. In [5,12], linear models have been generalized to sublinear models for graphs. Similarly as for linear models, an understanding of sublinear methods is essential for understanding extensions of non-sublinear models on graphs [6,7].

Here, we are interested in understanding the relationship between the performance of sublinear models and the different ways with which we can label the classes. In two-class problems, it is common practice to label one class as positive and the other as negative. For linear models, the classification performance is independent of how we label both classes. The reason is that each vector has an additive inverse. The existence of an inverse allows us to interpret the class regions separated by a hyperplane $\mathcal{H}$ in two ways: the normal of $\mathcal{H}$ points to the positive class. The additive inverse of a normal of $\mathcal{H}$ is also a normal pointing towards the opposite direction. Thus, normal and its additive inverse define the same class regions but with different class labels. As a consequence, there is a dual to each linear function that defines the same class regions but with flipped labels. Since a well-defined addition on graphs is unknown within the framework of sublinear models, the question arises whether there is also a dual for each sublinear function on graphs.

This contribution proves that in almost all cases there is no dual of a sublinear function. Empirical evaluation on relatively small samples confirm that

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 93–102, 2014.

the classification performance of sublinear models depend on whether we label a given class as positive or negative. These findings suggest to devise flip-flop sublinear models that choose the class labeling resulting in better classification accuracy. In experiments we show that flip-flop sublinear models perform better than standard sublinear models.

2 Sublinear Models on Attributed Graphs

This section introduces sublinear models for graphs as proposed by [12].

2.1 The Space of Attributed Graphs

Let $\mathbb{A}$ be a set of node and edges attributes. For the sake of convenience, we assume that $\mathbb{A}$ is the Euclidean space $\mathbb{R}^d$, though the theory presented in this paper can be adapted to the case where node and edges attributes come from arbitrary and possibly disjoint sets. We consider graphs of the form $X = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V}$ represents a set of vertices, $\mathcal{E}$ a set of edges, and $\mathcal{A} \subseteq \mathbb{A}$ a set of attributes of the nodes and edges. Node attributes take the form $\boldsymbol{x}_{ii} \in \mathcal{A}$ for each node $i \in \mathcal{V}$ and edges attributes are given by $\boldsymbol{x}_{ij} \in \mathcal{A}$ for each edge $(i, j) \in \mathcal{E}$. By $\mathcal{X}_\mathcal{G}$ we denote the space of all graphs with attributes from $\mathbb{A}$.

Without loss of generality, we may assume that edges must have non-zero attributes. Then each graph can be regarded as a complete graph, where non-edges are treated as edges with zero-attribute. Note that vertices may have zero as well as non-zero attributes. Including non-edges as edges with zero attribute allows us to express graphs X by a matrix $\boldsymbol{X} = (\boldsymbol{x}_{ij})$ with elements $\boldsymbol{x}_{ij} \in \mathcal{A}$.

2.2 Sublinear Dot Product

We equip the space $\mathcal{X}_\mathcal{G}$ with a graph similarity, called sublinear dot product.

Suppose that X is a graph with matrix representation $\boldsymbol{X}$. The particular form of the matrix depends on how the nodes of X are ordered. Since there is no canonical ordering of the nodes, each re-ordering may result in a different matrix representation of X. Let $[\boldsymbol{X}]$ denote the equivalence class of all matrices obtained by permuting the nodes of graph X in all possible ways. We write $\boldsymbol{X}' \in X$ to denote that $\boldsymbol{X}'$ is a representtive of the equivalence class $[\boldsymbol{X}]$.

To formulate the sublinear dot product of two graphs X and Y, we assume that both graphs have the same number of nodes. If this is not the case, we can safely add isolated nodes with zero attribute to the smaller graph until both graphs have the same number of nodes.

Let $\boldsymbol{X} = (\boldsymbol{x}_{ij})$ and $\boldsymbol{Y} = (\boldsymbol{y}_{ij})$ be matrix representations of X and Y, resp., of the same size. Then we define the dot product of $\boldsymbol{X}$ and $\boldsymbol{Y}$ by

$$\boldsymbol{X}^T \boldsymbol{Y} = \sum_{i,j} \boldsymbol{x}_{ij}^T \boldsymbol{y}_{ij},$$

where $\boldsymbol{x_{ij}}^T \boldsymbol{y}_{ij}$ denotes the dot product between attribute vectors $\boldsymbol{x}_{ij}$ and $\boldsymbol{y}_{ij}$. Observe that the dot products $\boldsymbol{x_{ij}}^T \boldsymbol{y}_{ij}$ correspond to node and edge similarities.

The sublinear dot product maximizes the dot product over all possible matrix representations and is of the form

$$X \cdot Y = \max \left\{ \boldsymbol{X}^T \boldsymbol{Y} \,:\, \boldsymbol{X} \in X, \boldsymbol{Y} \in Y \right\}.$$

As shown in [8], we can equivalently express $X \cdot Y$ by

$$X \cdot Y = \max \left\{ \boldsymbol{X}^T \boldsymbol{Y} \,:\, \boldsymbol{X} \in X \right\} = \max \left\{ \boldsymbol{X}^T \boldsymbol{Y} \,:\, \boldsymbol{Y} \in Y \right\},$$

where $\boldsymbol{Y} \in Y$ is an arbitrarily chosen matrix representation for the first equation and $\boldsymbol{X} \in X$ for the second equation. We call $\boldsymbol{X}$ and $\boldsymbol{Y}$ optimally aligned, if

$$X \cdot Y = \boldsymbol{X}^T \boldsymbol{Y}.$$

The sublinear dot product extends the dot product from vectors to graphs. It is straightforward to verify that the function $f_Y(X) = X \cdot Y$ as a pointwise maximizer of dot products is sublinear, that is convex and positively homogeneous. For this reason, we call $X \cdot Y$ sublinear. Though the sublinear dot product is not linear, it shares similar geometrical properties and generalizes the concept of maximum common subgraph [8]. It can be reduced to a special case of the graph-edit distance and is widely used in different guises as a common choice of proximity measure for graphs [2,3,17].

2.3 Sublinear Models for Graphs

Sublinear Functions. Sublinear functions on graphs are functions of the form

$$f(X) = W \cdot X + b, \tag{1}$$

where W is the weight graph and $b \in \mathbb{R}$ is the bias. We assign a graph X to class $y = +1$ if $f(X) \geq 0$ and to class $y = -1$ if $f(X) < 0$. Then the equation $f(X) = 0$ defines a decision surface

$$\mathcal{H}_f = \{X \in \mathcal{X}_\mathcal{G} \,:\, f(X) = 0\} \subseteq \mathcal{X}_\mathcal{G}$$

that separates both class regions.

The Learning Problem. The goal of learning consists in finding a weight graph W and bias b such that the s-linear discriminant $f(X) = W \cdot X + b$ minimizes the expected risk

$$E(f) = \int_{\mathcal{X}_\mathcal{G}} L(f(X), y)\, dP(X, y), \tag{2}$$

where $P(X, y)$ is the joint probability distribution on $\mathcal{X}_\mathcal{G} \times \mathcal{Y}$ and $L(\hat{y}, y)$ is a differentiable loss function that measures the cost of predicting class $\hat{y}$ when the actual class is y.

Since the distribution $P(X, y)$ is usually unknown, the expected risk $E(f)$ can not be computed directly. Instead, we approximate the expected risk by minimizing the empirical risk

$$E_N(f) = \frac{1}{N} \sum_{i=1}^{N} L(f(X_i), y_i)$$

on the basis of N training examples $(X_i, y_i) \in \mathcal{X}_\mathcal{G} \times \mathcal{Y}$.

Subgradient Learning Rules. To minimize the empirical risk $E_N(f)$, we present the margin perceptron algorithm for graphs. For this let $f(X) = W \cdot X + b$ be a sublinear function. We define a tube $\mathcal{T}_{f,\lambda}$ around the decision boundary $\mathcal{H}_f$ consisting of all graphs X with $|f(X)| \leq \lambda$, where $\lambda \geq 0$ is the margin parameter. Suppose that (X, y) is a new training example. The margin perceptron updates the weight graph and bias if one of the two cases occurs: (1) f misclassifies X, or (2) f correctly classifies X, but X lies in the tube $\mathcal{T}_{f,\lambda}$. Both conditions are met when $y \cdot f(X) \leq 0$. In this case the update rule is of the form

$$\begin{aligned} \boldsymbol{W} &\leftarrow \boldsymbol{W} + \eta \cdot y \cdot \boldsymbol{X} \\ b &\leftarrow b + \eta \cdot y, \end{aligned}$$

where η is the learning rate and $\boldsymbol{W} \in W$ and $\boldsymbol{X} \in X$ are optimally aligned matrix representations of W and X, that is $W \cdot X = \boldsymbol{W}^T \boldsymbol{X}$.

As shown in [12], the update rule of the graph perceptron minimizes the empirical risk $E_N(f)$, where the underlying loss function is of the form

$$L(f(X), y) = \max\{0, \lambda - y \cdot f(X)\}.$$

For $\lambda = 0$, we obtain the graph perceptron algorithm as a special case. Convergence is discussed in [11,12].

3 Flip-Flop Sublinear Models

The main result of this contribution is Theorem 1 stating that the performance of sublinear models depends on which of both classes is labeled as positive class. As an implication of Theorem 1, we introduce flip-flop sublinear models.

Each sublinear function $f(X) = W \cdot X + b$ separates the graph space $\mathcal{X}_\mathcal{G}$ into two class regions of the form

$$\mathcal{R}_+(f) = \{X \in \mathcal{X}_\mathcal{G} \,:\, f(X) \geq 0\} \quad \text{and} \quad \mathcal{R}_-(f) = \{X \in \mathcal{X}_\mathcal{G} \,:\, f(X) < 0\},$$

where $\mathcal{R}_+(f)$ is the region for the positive and $\mathcal{R}_-(f)$ the region of the negative class. We define the class-dual of f as a sublinear function $f'(X) = W' \cdot X + b'$ separating $\mathcal{X}_\mathcal{G}$ into class regions of the form

$$\mathcal{R}_+(f') = \mathcal{R}_-(f) \quad \text{and} \quad \mathcal{R}_-(f') = \mathcal{R}_+(f).$$

By definition, the class-dual f' is a sublinear function that implements the same class regions as f but with flipped labels. In vector spaces, each linear function $h(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$ with non-zero weight $\boldsymbol{w}$ has a unique class-dual, which is of the form $h'(\boldsymbol{x}) = -\boldsymbol{w}^T\boldsymbol{x} - b$. This statement is invalid in graph spaces as shown by the next result (a proof is presented in [13]).

Theorem 1. *There is no class-dual of a sublinear function with probability one.*

Suppose that the graph space is partitioned in two class regions $\mathcal{R}_+$ and $\mathcal{R}_-$. Consider the expected classification accuracy of the sublinear function f

$$\mathbb{C}_{ref}[f] = \int [f(X), y]\, dP(X, y),$$

where $[f(X), y] = 1$ if f correctly classifies X as y, and 0 otherwise. The subscript of $\mathbb{C}_{ref}$ refers to the current labeling of the classes as the reference labels. If there is a sublinear model f^* with $\mathbb{C}_{ref}[f^*] = 1$, then f^* perfectly separates both classes. When flipping the labels, we assign graphs from class region $\mathcal{R}_+$ negative and graphs from class region $\mathcal{R}_-$ positive labels. Under the assumption that $\mathbb{C}_{ref}[f^*] = 1$, Theorem 1 yields

$$\max_f \mathbb{C}_{flip}[f] < \mathbb{C}_{ref}[f^*] = 1,$$

where $\mathbb{C}_{flip}$ is the expected classification accuracy when the original class labels have been flipped. In a more general setting, we arrive at the following result:

Corollary 1. *For two-class problem, we generally have*

$$\max_f \mathbb{C}_{ref}[f] \neq \max_f \mathbb{C}_{flip}[f],$$

where the maximum is taken over all sublinear functions on graphs.

Corollary 1 gives rise to flip-flop sublinear models that selects the labeling resulting in a better separation of the data:

Algorithm 1. (Flip Flop Classifier)

Input:

Sample $\mathcal{S} \subseteq \mathcal{X}_\mathcal{G} \times \mathcal{Y}$.

Procedure:

Learn sublinear classifier $f(X)$ on the basis of $\mathcal{S}$ with accuracy $\alpha_\mathcal{S}(f)$.
Construct dual sample $\mathcal{S}'$ according to $(X, y) \in \mathcal{S} \Leftrightarrow (X, -y) \in \mathcal{S}'$.
Learn sublinear classifier f' on the dual sample $\mathcal{S}'$ with accuracy $\alpha_{\mathcal{S}'}(f')$.

Return:

Classifier and labeling that yields a higher classification accuracy

$$f^* = \arg\max_{f, f'} \left\{\alpha_\mathcal{S}(f), \alpha_{\mathcal{S}'}(f')\right\}$$

---

4 Experiments

Experiments on two-class problems aims at investigating the behavior of sublinear models under different class-labelings. Experiments on multi-class problems aim at assessing the performance of flip-flop sublinear models in a practical setting when class-labeling is random-like.

4.1 Data

We selected subsets of the following training data sets from the IAM graph database repository [16]: letter (low, medium, high), fingerprint, grec, and coil. The *letter* data sets compile distorted letter drawings from the Roman alphabet that consist of straight lines. Lines of a letter are represented by edges and endpoints of lines by vertices. The distortion levels are low, medium, and high. Fingerprint images of the *fingerprints* data set are converted into graphs, where vertices represent endpoints and bifurcation points of skeletonized versions of relevant regions. Edge represent ridges in the skeleton. The *grec* data set consists of graphs representing symbols from noisy versions of architectural and electronic drawings. Vertices represent endpoints, corners, intersections, or circles. Edges represent lines or arcs. The $coil_{13,16}$ and $coil_{42,44}$ data sets are subsets of the coil-100 data set consisting of objects corresponding to the subscripted pairs of indices $3, 16$ and $42, 44$ (starting at index 0). The first pair of indices refers to images representing two different types of rubber cats and the second pair to images representing two different types of cups. After preprocessing, the images are represented by graphs, where vertices represent endpoints of lines and edges represent lines. All datasets consist of a fixed training-, validation-, and test-set.

4.2 Experiments – Label Dependency

Data. We considered the following two-class problems: (1) letters A and H of the letter-high data set, (2) letters E and F of the letter-high data set, (3) classes 0 and 1 of the fingerprint data set, (4) $\text{coil}_{13,16}$, and (5) $\text{coil}_{42,44}$.

Experimental Protocol. For each data set, we randomly sampled 50% off all data for training in a stratified manner. The remaining examples formed the test set. Then we applied the graph-perceptron algorithm ($\lambda = 0$) using the graduated assignment algorithm [3] for computing the sublinear dot product. We recorded the classification accuracy on the training- and test-examples. The learning-rates of the perceptron algorithm were taken from [12]. We repeated this experiments 50 times. Next, for each data set, we flipped the labels of both classes and repeated the same experiment again 50 times.

Results and Discussion. Table 1 summarizes the results. The Shapiro-Wilk test at significance level $\alpha = 0.05$ rejected in about half of the cases the hypothesis that the classification accuracies are normally distributed. For this reason,

Table 1. Results of the graph-perceptron algorithm for two-class classification problems using the original and the flipped class labeling. Shown are the average classification accuracies over 50 trials, the standard deviation and the p-values obtained from the Mann-Whitney U-Test. Rows marked with $'+'$ refer to a class labeling with better results than rows marked with $'-'$. The quantity d/N is the ratio of the dimension d of the largest matrix representation of a training graph and the number N of training examples.

			training			test		
	d/N		avg	std	p-val	avg	std	p-val
Letter-High$_{A,H}$	0.3	+	100.0	0.0	0.000	96.2	2.0	0.000
		−	95.4	1.4		89.2	2.7	
Letter-High$_{E,F}$	0.3	+	98.4	0.9	0.007	90.3	2.8	$\times$0.259
		−	97.6	1.6		89.8	2.2	
Fingerprint$_{0,1}$	0.8	+	99.8	0.1	0.000	96.3	0.3	0.000
		−	73.2	3.3		72.0	3.2	
Coil$_{13,16}$(cats)	59.1	+	100.0	0.0	$\times$0.230	89.2	7.0	0.000
		−	99.6	1.3		75.2	9.4	
Coil$_{42,44}$ (cups)	41.0	+	100.0	0.0	0.000	87.3	5.5	0.000
		−	97.4	4.2		70.2	6.85	

we applied the Mann-Withney U-Test for testing the null hypothesis whether the classification accuracies of the original labeling and the flipped labeling come from the same distribution. In all but two cases (marked as $\times$), the resulting p-values were less than the significance levels $\alpha = 0.01$ and $\alpha = 0.05$. In these 8 out of 10 cases, we rejected the null hypothesis and accepted the alternative hypothesis that the accuracies come from different distributions.

The results show that the average accuracies are different in 8 out of 10 cases and that the differences are significant. From this we conclude that the average accuracy of a graph-perceptron depends on our choice as to which class we label as positive and negative. Recall that Theorem 1 and its implications consider the expected classification accuracy. Empirical classification accuracies based on finite samples of relatively small size according to the ratio d/N confirm that the theoretical findings are of practical relevance.

In all cases, a class labeling with better average accuracy results in a better average generalization performance. This also holds when the difference on the training set is not statistically significant as in the Coil$_{13,16}$ data set. Conversely, a statistically significant difference on the training set does not guarantee a statistically significant difference on the test set as shown for Letter-High$_{E,F}$.

As expected generalization performance was lower than the performance on the training examples. Notable is the strong decrease of generalization performance for both Coil data sets indicating overfitting. Inspecting the ratio d/N of the dimension of the largest matrix representation of a graph in the training set and the number N of training examples shows that the dimension is roughly 40 and 60 times higher than the number of training examples. According to Covers

Function Counting Theorem, we can always find a separating decision surface for the training set provided the classes are labeled favorably and the training examples are in general position. In addition, we can also expect good results on the training data, when the class labeling is unfavorable. Due to the high dimension and the low number of training examples, it is likely that the learned models do not generalize well.

4.3 Experiments – Flip-Flop Sublinear Models

Data. We considered the following multi-class problems: letter (low, medium, high), fingerprint, and grec.

Experimental Protocol. To cope with multiple classes, we applied the flip-flop perceptron and the flip-flop margin perceptron algorithm using a one-versus-all approach. For computing the sublinear dot product, we again applied the graduated assignment algorithm [3]. The learning-rates and margin parameters were taken from [12]. For each data set, we trained both flip-flop sublinear models on the union of the training and validation data. We assessed the generalization performance on the test data. We repeated this experiment 10 times. We used the given splits instead of random splits of the training and test data in order to make the results comparable to other methods.

Results and Discussion. Table 2 summarizes the training and test results of the flip flop perceptron and flip flop margin perceptron algorithm for multi-class problems.

We first compare the training results of the four different graph perceptron algorithms. We observe that on average both flip-flop perceptrons better separate the training data than their standard counterparts. We also observe that for flip-flop classifiers, the margin perceptron does not yield the best training results on three out of five data sets (Letter M, F'print, GREC). This is in contrast to the standard versions of both perceptron algorithms. Finally, we see that the differences are small compared to those obtained in our first experiments on two-class problems. One reason for this is that our experiments on two-class problems consider the extreme case of an unfavorable vs. a favorable labeling, whereas these experiments consider the natural labeling vs. the favorable labeling. The natural labeling in a one-against-all classification approach labels the corresponding single class as positive and all other classes as negative. For most classes, this labeling turns out to be the favorable one, such that flipping the labeling is necessary only in few cases. Thus, for most one-against-all dichotomies in these experiments, the natural labeling coincides with the favorable labeling.

Next, we compare the test results of the four different graph perceptron algorithms. The first observation to be made is that both flip-flop perceptrons perform better on average than their standard counterparts on all but the GREC data set.

Table 2. Training and test classification accuracies for multi-class problems. The number of classes is shown in parentheses next to the identifier of the respective data set. Shown are the average accuracy, standard deviation, and maximum accuracy over 10 runs of perceptron (perc), margin perceptron (mperc), flip-flop perceptron (ffperc), and flip-flop margin perceptron (ffmperc) for training and test data. Generalization performance is compared against graph Bayes ($bayes_2$), generalized learning graph quantization (glgq), similarity-kernel SVM (sk-svm), support vector machine applied on dissimilarity embeddings after dimension reduction using PCA (pca+svm), and optimized dissimilarity space embedding (odse.v2). Results for entries with a dash – are not available. Results marked with an asterique * are not comparable, because the grec data set as used in [1] differs from the on publicly available at [16].

train	Letter L (15)		Letter M (15)		Letter H (15)		F'print (4)		GREC (22)	
	avg	max	avg	max	avg	max	avg	max	avg	max
ffperc	$97.0^{\pm 0.5}$	97.6	$93.3^{\pm 0.4}$	94.0	$86.7^{\pm 0.8}$	87.7	$87.1^{\pm 0.5}$	88.0	$99.5^{\pm 0.3}$	100.0
ffmperc	$98.9^{\pm 0.2}$	99.2	$93.1^{\pm 0.2}$	93.5	$88.9^{\pm 0.5}$	89.7	$86.5^{\pm 0.7}$	87.3	$99.4^{\pm 0.2}$	99.8
perc	$96.9^{\pm 0.3}$	97.5	$92.0^{\pm 0.6}$	93.1	$84.3^{\pm 0.7}$	85.3	$80.0^{\pm 1.7}$	81.8	$98.2^{\pm 0.3}$	98.6
mperc	$96.9^{\pm 0.3}$	97.2	$92.6^{\pm 0.5}$	93.7	$86.4^{\pm 0.5}$	87.1	$80.7^{\pm 2.9}$	84.3	$99.0^{\pm 0.2}$	99.1

test	Letter L (15)		Letter M (15)		Letter H (15)		F'print (4)		GREC (22)	
	avg	max	avg	max	avg	max	avg	max	avg	max
ffperc	$95.6^{\pm 0.5}$	96.1	$89.4^{\pm 0.6}$	90.3	$83.5^{\pm 0.9}$	84.3	$82.3^{\pm 0.7}$	83.4	$96.1^{\pm 0.4}$	96.6
ffmperc	$97.0^{\pm 0.4}$	97.5	$90.4^{\pm 0.8}$	91.7	$85.6^{\pm 0.7}$	86.5	$83.1^{\pm 0.5}$	84.0	$96.9^{\pm 0.3}$	97.4
perc [12]	$94.5^{\pm 0.7}$	96.0	$86.1^{\pm 1.1}$	87.5	$80.7^{\pm 1.1}$	82.5	$76.8^{\pm 1.6}$	79.1	$96.3^{\pm 0.5}$	97.0
mperc [12]	$95.5^{\pm 0.3}$	95.7	$88.7^{\pm 0.6}$	89.5	$84.1^{\pm 0.5}$	84.8	$79.5^{\pm 2.6}$	82.4	$97.5^{\pm 0.6}$	98.1
$bayes_2$ [9]	–		–		80.4		79.2		89.9	
glgq [10]	–		–		88.4		84.8		97.5	
sk-svm [1]	99.2		94.7		92.8		81.7		92.2*	
pca+svm [1]	99.2		94.9		92.1		82.2		92.0*	
odse.v2 [14]	99.0		96.8		96.2		–		97.9	

Let us compare the results of the flip-flop classifiers against $bayes_2$ and glgq. Both are also classifiers based on the graph orbifold framework. $Bayes_2$ is based on bell-shaped distributions around center graphs and glgq is an extension of generalized learning vector quantization to the graph domain. The results show that both flip-flop classifiers are superior than $bayes_2$ but perform worse compared with glgq.

Finally, we compare both flip-flop classifiers against other state-of-the-art algorithms. From the results we see that sk-svm, pca+svm, and odse.v2 are clearly superior on all letter data set and therefore more robust against noise. The picture changes when it comes to the F'print and GREC data set. The graph perceptrons algorithms are slightly better on F'print and comparable to odse.v2 on GREC. These findings are similar to linear in vector spaces: graph perceptrons are simple methods that yield possibly inaccurate results. Further improvements are possible in two ways: first, by more extensively exploring the hyperparameters, and second, by controlling the VC dimension via the number of nodes and edges of the weight graph (for details see [12]).

5 Conclusion

Theorem 1 states that there is no dual of a sublinear model with probability one. An immediate consequence of this result is that the expected classification performance of sublinear models depends on our choice as to which class we label as positive and which as negative. Experiments on finite samples of relatively small size compared to the dimensionality of the data confirm the theoretical findings for empirical classification performance. This justifies flip-flop classifiers that consider both labelings during training and select the model with the better classification performance of the training data.

References

1. Bunke, H., Riesen, K.: Improving vector space embedding of graphs through feature selection algorithms. Pattern Recognition 44(9), 1928–1940 (2011)
2. Cour, T., Srinivasan, P., Shi, J.: Balanced graph matching. In: NIPS (2006)
3. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. IEEE Trans. on Pattern Analysis and Machine Intelligence 18(4), 377–388 (1996)
4. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning. Springer, New York (2001)
5. Jain, B., Wysotzki, F.: Multi-Layer Perceptron Learning in the Domain of Graphs. IJCNN 3, 1993–1998 (2003)
6. Jain, B., Wysotzki, F.: Structural perceptrons for attributed graphs. In: Fred, A., Caelli, T.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.) SSPR&SPR 2004. LNCS, vol. 3138, pp. 85–94. Springer, Heidelberg (2004)
7. Jain, B.: Structural Neural Learning Machines. PhD thesis, TU Berlin (2005)
8. Jain, B., Obermayer, K.: Structure Spaces. The Journal of Machine Learning Research 10, 2667–2714 (2009)
9. Jain, B., Obermayer, K.: Maximum likelihood for gaussians on graphs. In: Jiang, X., Ferrer, M., Torsello, A. (eds.) GbRPR 2011. LNCS, vol. 6658, pp. 62–71. Springer, Heidelberg (2011)
10. Jain, B., Obermayer, K.: Generalized learning graph quantization. In: Jiang, X., Ferrer, M., Torsello, A. (eds.) GbRPR 2011. LNCS, vol. 6658, pp. 122–131. Springer, Heidelberg (2011)
11. Jain, B., Obermayer, K.: Learning in Riemannian Orbifolds. arXiv preprint arXiv:1204.4294 (2012)
12. Jain, B.: Sublinear Models for Graphs. arXiv preprint arXiv:1204.4294 (2014)
13. Jain, B.: Flip-Flop Sublinear Models for Graphs: Proof of Theorem 1. arXiv preprint arXiv:cs.LG (2014)
14. Livi, L., Rizzi, A., Sadeghian, A.: Optimized Dissimilarity Space Embedding for Labeled Graphs. Information Sciences (2014)
15. 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)
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.) SSPR&SPR 2008. LNCS, vol. 5342, pp. 287–297. Springer, Heidelberg (2008)
17. Umeyama, S.: An eigendecomposition approach to weighted graph matching problems. IEEE Transactions on PAMI 10(5), 695–703 (1988)

Node Centrality for Continuous-Time Quantum Walks

Luca Rossi[1], Andrea Torsello[2], and Edwin R. Hancock[3]

[1] School of Computer Science, University of Birmingham, UK
[2] Department of Environmental Science, Informatics, and Statistics, Ca' Foscari University of Venice, Italy
[3] Department of Computer science, University of York, UK

Abstract. The study of complex networks has recently attracted increasing interest because of the large variety of systems that can be modeled using graphs. A fundamental operation in the analysis of complex networks is that of measuring the *centrality* of a vertex. In this paper, we propose to measure vertex centrality using a continuous-time quantum walk. More specifically, we relate the importance of a vertex to the influence that its initial phase has on the interference patterns that emerge during the quantum walk evolution. To this end, we make use of the quantum Jensen-Shannon divergence between two suitably defined quantum states. We investigate how the importance varies as we change the initial state of the walk and the Hamiltonian of the system. We find that, for a suitable combination of the two, the importance of a vertex is almost linearly correlated with its degree. Finally, we evaluate the proposed measure on two commonly used networks.

Keywords: Vertex Centrality, Complex Network, Quantum Walk, Quantum Jensen-Shannon Divergence.

1 Introduction

In recent years, an increasing number of researchers have turned their attention to the study complex networks [1]. Complex network are ubiquitous in a large number of real-world systems. A non-exhaustive list of examples includes metabolic networks [2], protein interactions [3], brain networks [4] and scientific collaboration networks [5]. A fundamental task in complex network analysis is that of measuring the centrality of a vertex, i.e., its importance. To this end, a number of centrality indices have been introduced in the literature [1, 6–9]. Each of these captures different but equally significant aspects of vertex importance.

Perhaps the most intuitive centrality measure is degree centrality [7]. This is defined as the number of links incident upon a node, i.e., the degree of the node. The degree centrality naturally interprets the number of edges incident on a vertex as a measure of its "popularity", or, alternatively, as the risk of a node being infected in an epidemiological scenario. Closeness centrality [10], on the other hand, links the importance of a vertex to its proximity to the remaining vertices

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 103–112, 2014.

of the graph. More precisely, the closeness centrality is defined as the inverse of the sum of the distance of a vertex to the remaining nodes of the graph, i.e., $CC(u) = \frac{n-1}{\sum_{v=1}^{n} d(u,v)}$ where $d(u,v)$ denotes the shortest path distance between nodes u and v. The betweenness centrality [7] is a measure of the extent to which a given vertex lies on the paths between the remaining vertices, where the path may be either that of shortest length or a random walk between the nodes. If $sp(v_1, v_2)$ denotes the number of shortest paths from node v_1 to node v_2, and $sp(v_1, u, v_2)$ denotes the number of shortest paths from v_1 to v_2 that pass through node u, the betweenness centrality of u is $BC(u) = \sum_{v_1=1}^{n} \sum_{v_2=1}^{n} \frac{sp(v_1,u,v_2)}{sp(v_1,v_2)}$. Note that this definition assumes that the communication takes place along the shortest path between two vertices. A number of measures have been introduced to account for alternative scenarios in which the information is allowed to flow through different paths [1, 6–8].

Recently, there has also been a surge of interest in using quantum walks as a primitive for designing novel quantum algorithms on graph structures [11]. Quantum walks on graphs represent the quantum mechanical analogue of the classical random walk on a graph. Despite being similar in their definition, the dynamics of the two walks can be remarkably different. In the classical case the evolution of the walk is governed by a double stochastic matrix, while in the quantum case the evolution is governed by a unitary matrix, thus rendering the walk reversible and non-ergodic. Moreover, the state vector of the classical random walk is real-valued, while in the quantum case the state vector is complex-valued. As there is no constraint on the sign and phase of the amplitudes, different paths are allowed to interfere with each other in both constructive and destructive ways. This in turn gives rise to faster hitting times and reduces the problems of tottering observed in classical random walks [11].

In this paper, we propose to measure the centrality of a vertex using a continuous-time quantum walk. More specifically, we relate the importance of a vertex to the influence that its initial phase has on the evolution of a suitably defined quantum walk. To this end, we make use of the quantum Jensen-Shannon divergence, a recently introduced generalisation of the classical Jensen-Shannon divergence to quantum states [12]. Just as the classical Jensen-Shannon divergence [13], the quantum Jensen-Shannon divergence is symmetric, bounded and always defined. From a physical perspective, the QJSD is computed from density matrices, whose entries are observables. As a consequence, it should be possible, at least in theory, to design a quantum algorithm to compute the QJSD centrality that could benefit from the power of quantum computers. However, the design of such an algorithm is beyond the scope of this paper.

The remainder of this paper is organised as follows: Section 2 provides an essential introduction to the basic terminology required for understanding the proposed quantum mechanical framework. With these notions to hand, we introduce our centrality measure in Section 3 and we study its properties. In Section 4 we apply the proposed measure to the analysis of two commonly used network models, while the conclusions are presented in Section 5.

2 Quantum Mechanical Background

The continuous-time quantum walk [14] is a natural quantum analogue of the classical random walk. Given a graph $G = (V, E)$, classical random walks model a diffusion process over the node set V, and have proven to be a useful tool in the analysis of its structure. Similarly, the continuous-time quantum walk is defined as a dynamical process over the vertices of the graph. By contrast to the classical case, where the state vector is constrained to lie in a probability space, in the quantum case the state of the system is defined through a vector of complex amplitudes over the node set V whose squared norm sums to unity over the nodes of the graph, with no restriction on their sign or complex phase. These phase differences allow interference effects to take place. Moreover, in the quantum case the evolution of the state vector of the walker is governed by a complex valued unitary matrix, whereas the dynamics of the classical random walk is governed by a stochastic matrix. Hence the evolution of the quantum walk is reversible, implying that quantum walks are non-ergodic and do not possess a limiting distribution. As a result, the behaviour of classical and quantum walks differs significantly, and quantum walks possess a number of interesting properties not exhibited by classical random walks.

More formally, using the Dirac notation, we denote the basis state corresponding to the walk being at vertex $u \in V$ as $|u\rangle$. A general state of the walk is a complex linear combination of the basis states, such that the state of the walk at time t is defined as

$$|\psi_t\rangle = \sum_{u \in V} \alpha_u(t) \, |u\rangle \tag{1}$$

where the amplitude $\alpha_u(t) \in \mathbb{C}$ and $|\psi_t\rangle \in \mathbb{C}^{|V|}$ are both complex.

At each instant in time the probability of the walker being at a particular vertex of the graph is given by the square of the norm of the amplitude of the relative state. Let X^t be a random variable giving the location of the walker at time t. Then the probability of the walker being at the vertex u at time t is given by $\Pr(X^t = u) = \alpha_u(t)\alpha_u^*(t)$, where $\alpha_u^*(t)$ is the complex conjugate of $\alpha_u(t)$. Moreover $\sum_{u \in V} \alpha_u(t)\alpha_u^*(t) = 1$ and $\alpha_u(t)\alpha_u^*(t) \in [0, 1]$, for all $u \in V$, $t \in \mathbb{R}^+$.

The evolution of the walk is then given by the Schrödinger equation, where we take the time-independent Hamiltonian of the system to be the graph Laplacian, yielding

$$\frac{\partial}{\partial t} |\psi_t\rangle = -iL \, |\psi_t\rangle \ . \tag{2}$$

Given an initial state $|\psi_0\rangle$, we can solve Eq. (2) to determine the state vector at time t

$$|\psi_t\rangle = e^{-iLt} \, |\psi_0\rangle \ . \tag{3}$$

Note that generally one may use any Hermitian operator as the Hamiltonian. Common choices are the graph adjacency matrix, the normalised Laplacian and the signless Laplacian.

Finally, we can compute the spectral decomposition of the graph Laplacian $L = \Phi\Lambda\Phi^\top$, where Φ is the $n \times n$ matrix $\Phi = (\phi_1|\phi_2|...|\phi_j|...|\phi_n)$ with the ordered eigenvectors ϕ_js of L as columns and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_j, ..., \lambda_n)$ is the $n \times n$ diagonal matrix with the ordered eigenvalues λ_j of L as elements, such that $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$. Using the spectral decomposition of the graph Laplacian and the fact that $\exp[-iLt] = \Phi\exp[-i\Lambda t]\Phi^\top$ we can then write

$$|\psi_t\rangle = \Phi e^{-i\Lambda t}\Phi^\top |\psi_0\rangle \ . \tag{4}$$

2.1 Quantum Jensen-Shannon Divergence

The *density operator* (or *density matrix*) is introduced in quantum mechanics to describe a system whose state is an ensemble of pure quantum states $|\psi_i\rangle$, each with probability p_i. The density operator of such a system is defined as

$$\rho = \sum_i p_i \, |\psi_i\rangle \, \langle\psi_i| \ . \tag{5}$$

The von Neumann entropy [15] H_N of a density operator ρ is defined as

$$H_N = -\operatorname{tr}(\rho \log \rho) = -\sum_i \xi_i \ln \xi_i \tag{6}$$

where $\xi_1, \ldots, \xi_n$ are the eigenvalues of ρ. If $\langle\psi_i| \, \rho \, |\psi_i\rangle = 1$, i.e., the quantum system is a pure state $|\psi_i\rangle$ with probability $p_i = 1$, then the Von Neumann entropy $H_N(\rho) = -\operatorname{tr}(\rho \log \rho)$ is zero. On other hand, for a mixed state described by the density operator σ we have a non zero Von Neumann entropy associated with it.

With the Von Neumann entropy to hand, the quantum Jensen-Shannon divergence between two density operators ρ and σ is defined as

$$D_{JS}(\rho, \sigma) = H_N\Big(\frac{\rho + \sigma}{2}\Big) - \frac{1}{2}H_N(\rho) - \frac{1}{2}H_N(\sigma) \tag{7}$$

This quantity is always well defined, symmetric and positive definite. Finally, it can also be shown that $D_{JS}(\rho, \sigma)$ is bounded, i.e., $0 \leq D_{JS}(\rho, \sigma) \leq 1$.

3 QJSD Centrality

In order to measure the centrality of vertex v, we define two quantum walks where v is initially set to be in phase and in antiphase with the respect to the remaining nodes. Let the normalised graph Laplacian be the Hamiltonian of our system, and let $\left|\psi_0^{v-}\right\rangle = \sum_{u \in V} \alpha_u^{v-}(0) \, |u\rangle$ and $\left|\psi_0^{v+}\right\rangle = \sum_{u \in V} \alpha_u^{v+}(0) \, |u\rangle$ denote the quantum walks on G with initial amplitudes

$$\alpha_j^{v-}(0) = \begin{cases} -\frac{\sqrt{d}_j}{C} & \text{if } j = v \\ +\frac{\sqrt{d}_j}{C} & \text{otherwise} \end{cases} \qquad \alpha_j^{v+}(0) = \Big\{ +\frac{\sqrt{d}_j}{C} \ \ \forall j \tag{8}$$

where C is the normalisation constant such that probabilities sum to 1. In other words, we define the initial amplitude to be proportional to the square root of the node degrees. Finally, let ρ_{v^+} and ρ_{v^-} be the density operators which describe the ensembles of quantum states $\left|\psi_t^{v-}\right\rangle$ and $\left|\psi_t^{v+}\right\rangle$ respectively, i.e.,

$$\rho_{v^-} = \lim_{T\to\infty} \frac{1}{T} \int_0^T \left|\psi_t^{v-}\right\rangle \left\langle\psi_t^{v-}\right| \, \mathrm{d}t \qquad \rho_{v^+} = \lim_{T\to\infty} \frac{1}{T} \int_0^T \left|\psi_t^{v+}\right\rangle \left\langle\psi_t^{v+}\right| \, \mathrm{d}t \quad (9)$$

Given this setting, we can measure how the initial phase of the vertex v affects the evolution of the quantum walks by computing the distance between the quantum states defined by ρ_{v^-} and ρ_{v^+}. That is, we define the quantum Jensen-Shannon divergence (QJSD) centrality of a vertex v as

$$C_{QJSD}(v) = D_{JS}(\rho_{v^-}, \rho_{v^+}) \quad (10)$$

Note that the computational complexity of the QJSD centrality is bounded by that of computing the eigendecomposition of the graph laplacian, i.e., $O(n^3)$. Let $\Phi\Lambda\Phi^\top$ be the spectral decomposition of the graph normalised Laplacian and let $P_\lambda = \sum_{k=1}^{\mu(\lambda)} \phi_{\lambda,k}\phi_{\lambda,k}^\top$ be the projection operator on the subspace spanned by the $\mu(\lambda)$ eigenvectors $\phi_{\lambda,k}$ associated with the eigenvalue λ of the graph normalised Laplacian. Rossi et al. [16] have shown that $\rho_\infty = \sum_{\lambda=1}^{m} P_\lambda \rho_0 P_\lambda^\top$, where m denotes the number of unique eigenvalues of the graph normalised Laplacian. Note that as a consequence of Eq. 9 we have that ρ_{v^-} and ρ_{v^+} are simultaneously diagonalisable. That is, there exist a single invertible matrix M such that $M^{-1}\rho_{v^-}M$ and $M^{-1}\rho_{v^+}M$ are diagonal. More precisely, here $M = \Phi$, the $n \times n$ matrix $\Phi = (\phi_1|\phi_2|...|\phi_j|...|\phi_n)$ with the ordered eigenvectors ϕ_js of the Hamiltonian as columns.

3.1 Relation with Degree Centrality

We are now interested in studying the relation between the QJSD centrality and the degree centrality. It has been shown, for example, that the degree and the betweenness centrality are highly correlated [17]. This should not come as a surprise, as we expect high degree vertices to be more often included in the shortest path along a pairs of vertices.

Let the initial states of the walks be defined as in Eq. 8 and let the normalised Laplacian be the Hamiltonian of our system. We start by observing that $\left|\psi_0^{v+}\right\rangle = \sum_{u\in V} \alpha_u^{v+}(0)\left|u\right\rangle$ corresponds to the eigenvector ϕ_0 associated with the zero eigenvalue of the Hamiltonian, and as a consequence $\left|\psi_0^{v+}\right\rangle$ will remain constant over time. In other words, we have that $\rho_{v^+} = \left|\psi_0^{v+}\right\rangle \left\langle\psi_0^{v+}\right|$. Note that the spectrum of ρ_{v^+} is composed of a single eigenvector ϕ_0 with eigenvalue equal to 1. Moreover, recall from Eq.9 that ρ_{v^-} and ρ_{v^+} are co-diagonalisable matrices. As a result, each eigenvalue of $\rho_{v^-} + \rho_{v^+}$ is a sum of eigenvalues of ρ_{v^-} and ρ_{v^+}. More precisely, when the two walks are initialised as in Eq. 8, all the eigenvalues μ_i of $\frac{\rho_{v^-}+\rho_{v^+}}{2}$ will be equal to the eigenvalues of ρ_{v^-}, except for the eigenvalue $\mu_0 + 1$ which is associated to the common eigenvector ϕ_0.

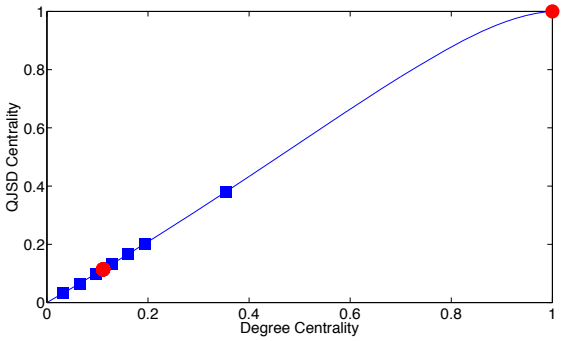


Fig. 1. The correlation between degree and QJSD centrality, for a star graph (red dots) and a scale-free graph (blue squares). The blue line shows the predicted dependency between the two centrality indices.

We now show that, as a consequence of this, the QJSD centrality is proportional to the degree centrality. Note that since ρ_{v^+} has a single non-zero eigenvalue which is equal to 1, we have that $H_N(\rho_{v^+}) = 0$. As a consequence of this and of Eq. 7, we have that

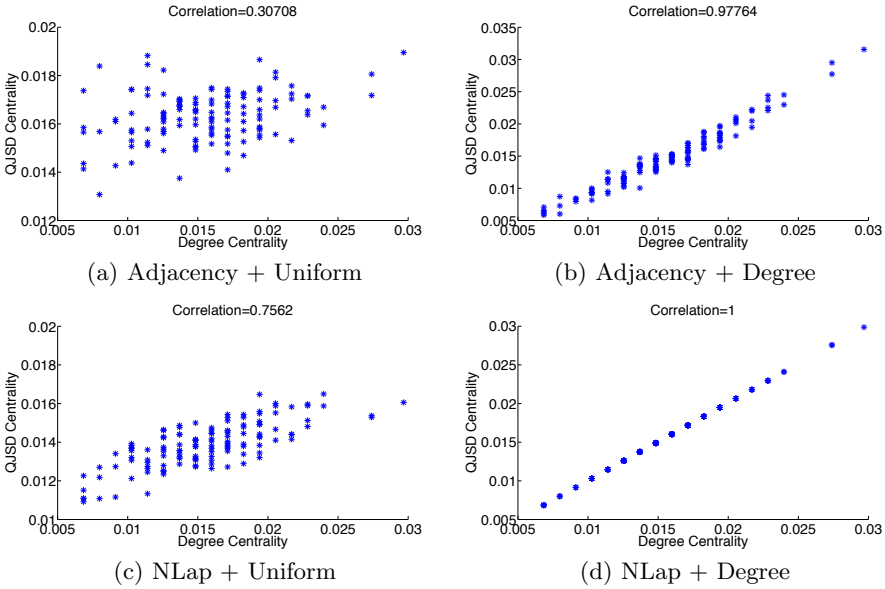
$$
\begin{aligned}
D_{JS}(\rho_{v^-}, \rho_{v^+}) &= H_N\left(\frac{\rho_{v^-} + \rho_{v^+}}{2}\right) - \frac{1}{2}H_N(\rho_{v^-}) \\
&= -\frac{\mu_0+1}{2}\log_2\frac{\mu_0+1}{2} - \sum_{i\neq 0}\frac{\mu_i}{2}\log_2\frac{\mu_i}{2} + \frac{1}{2}\sum_i \mu_i \log_2 \mu_i \\
&= \frac{\mu_0+1}{2} - \frac{\mu_0+1}{2}\log_2(\mu_0+1) + \sum_{i\neq 0}\frac{\mu_i}{2} - \frac{1}{2}\sum_{i\neq 0}\mu_i\log_2\mu_i + \frac{1}{2}\sum_i \mu_i\log_2\mu_i \\
&= 1 - \frac{1}{2}\log_2(\mu_0+1) + \frac{\mu_0}{2}\log_2\frac{\mu_0}{\mu_0+1} \qquad (11)
\end{aligned}
$$

where μ_i denotes the ith eigenvalue of ρ_{v^-} and we used the fact that $\sum_i \mu_i = 1$. We now proceed to show that μ_0 is proportional to the degree of node v, and therefore the QJSD centrality is proportional to the degree centrality. In fact, we have that

$$
\mu_0 = \langle\phi_0|\,\rho_0\,|\phi_0\rangle = \left\langle\phi_0\,\middle|\,\psi_0^{v-}\right\rangle^2 = \left(1 - \frac{d_v}{|E|}\right)^2 \qquad (12)
$$

where d_v is the degree of v and $|E|$ denotes the number of edges in the graph.

In other words, when we take the normalised Laplacian as our Hamiltonian and we initialise the walks according to Eq. 8, the QJSD centrality turns out to be quasi-linearly correlated with the degree centrality. Fig. 1 shows the correlation between the QJSD centrality and the degree centrality for a scale-free random graph and a star graph. Recall that the degree centrality is normalised between 0 and 1 by dividing it by $|V|(|V|-1)$, i.e., the maximum cardinality of the



(a) Adjacency + Uniform (b) Adjacency + Degree

(c) NLap + Uniform (d) NLap + Degree

Fig. 2. Correlation between the QJSD centrality and the degree centrality for different choices of the Hamiltonian (adjacency matrix or normalised Laplacian) and of the initial state (normalised uniform distribution or normalised degree distribution)

edge set. Note that the non-linearity of the correlation becomes evident only for those nodes with degree close to $|E|$, for which we have that $\frac{d_v}{|E|} \approx 0$ and thus $\mu_0 \approx 1 + \frac{d_v}{|E|}^2$.

So far we assumed that the Hamiltonian of the quantum walk is the graph normalised Laplacian. However, any Hermitian operator encoding the structure of the graph can be chosen as an alternative. Similarly, there is no constraint on the initial state of the walk, as long as it is a valid amplitude vector. Fig. 2 shows the correlation between the QJSD centrality and the degree centrality computed on a stochastic Kronecker graph for different choices of the initial state and the Hamiltonian. More specifically, we let the Hamiltonian be either the adjacency matrix or the normalised Laplacian of the graph, while the initial state is either proportional to the node degree as in Eq. 8 or uniformly equal to $1/\sqrt{n}$, where n denotes the number of nodes in the graph. As expected, our centrality measure is strongly correlated with the degree centrality when the Hamiltonian is the graph normalised Laplacian and the initial state is proportional to the node degree (see Fig. 2(d)). In general, we see that when the starting state is proportional to the node degree, the correlation tends to be very high, while the choice of a uniform initial state leads to a value of the centrality which is less dependent on the node degree.

Hence, in an attempt to capture structural information which are not trivially revealed by examining the node degree, we explore the consequences of letting the

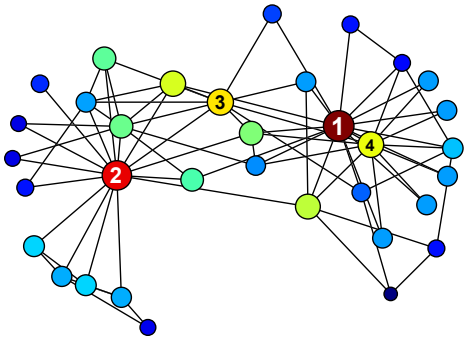


Fig. 3. Zachary's karate club network, where we have drawn each node with a diameter that is proportional to its QJSD centrality

walk start from a uniform amplitude vector and choosing the adjacency matrix as the Hamiltonian. Moreover, in order to balance the strength of the positive and negative signals, i.e., the contribution of the node amplitudes with either positive or negative phases, we let the magnitude of the initial amplitude on the node being analysed be equal to the sum of the amplitudes on the remaining nodes, which gives the initial state

$$\alpha_j^{v-}(0) = \begin{cases} -\frac{1}{\sqrt{2}} & \text{if } j = v \\ +\frac{1}{\sqrt{2(|V|-1)}} & \text{otherwise} \end{cases} \qquad \alpha_j^{v+}(0) = \begin{cases} +\frac{1}{\sqrt{2}} & \text{if } j = v \\ +\frac{1}{\sqrt{2(|V|-1)}} & \text{otherwise} \end{cases} . \tag{13}$$

4 Experimental Evaluation

We now apply the QJSD centrality to a pair of commonly used network datasets, namely Zachary's karate club [18] and Padgett's network of marriages between the 16 most eminent Florentine families in the 15th century [19]. Fig. 3 shows Zachary's karate club network, where each vertex is drawn with a diameter that is proportional to the QJSD centrality. We see that there are two main actors, node #1 and node #2, which correspond to the instructor and the administrator of the club. Note that using our measure the instructor turns out to be the node with the highest centrality, which is also the most central according to the degree centrality, while the betweeness centrality elects the administrator as the most important node. However, the betweenness centrality indicates as the second most important actor node #3, as this vertex has many contacts with both the members of the administrator cluster and the members of the instructor cluster and thus it is misunderstood as a center by the betweenness centrality. Finally, node #4 is identified as the third most important by the degree centrality, leaving node #3 at the fourth place, although the latter is more central in the sense that it shares many links with both groups.

Padgett's network of marriages is depicted in Fig. 4. In Table 1, we show the ranking of the 15 families according to their QJSD centrality. As expected, the

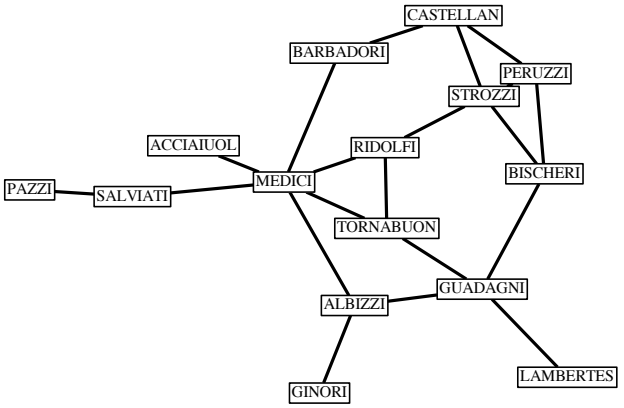


Fig. 4. Padgett's network of marriages between eminent Florentine families in the 15th century [19]. We omit the Pucci, which had no marriage ties with other families.

Table 1. The QJSD centrality of the families of Padgett's network [19]

Family	Centrality	Family	Centrality	Family	Centrality
Medici	0.4867	Castellan	0.3245	Salviati	0.2248
Ridolfi	0.4619	Barbadori	0.3205	Ginori	0.1993
Strozzi	0.4192	Albizzi	0.3172	Acciaiuol	0.1534
Tornabuon	0.4041	Guadagni	0.3091	Lambertes	0.1267
Bischeri	0.3586	Peruzzi	0.2990	Pazzi	0.1126

Medici easily outperform the Strozzi , who are their main rivals. This agrees with the historical view that Medici's supremacy was largely due to their skills in manipulating the marriage network. Interestingly the Pazzi, which is the most loosely connected family of the graph, achieve the lowest centrality. Note also that the Ridolfi family, which connect two of the most influential families at that time, the Medici and the Strozzi, is assigned a high centrality. Moreover, the Tornabuon, which form a tightly connected clique together with the Medici and the Ridolfi, is the fourth most central node of the network.

5 Conclusions

In this paper, we have proposed to measure vertex centrality using a continuous-time quantum walk. We measured the importance of a vertex as the influence that its initial phase has on the interference patterns that emerge during the quantum walk evolution. We have showed that, under particular settings, the resulting centrality measure is almost linearly correlated with degree centrality. Thus, we have proposed an alternative starting state where the contribution of the node amplitudes with positive and negative phases is equal. Finally, we have evaluated the resulting measure to two commonly used network models.

Acknowledgments. Edwin Hancock was supported by a Royal Society Wolfson Research Merit Award.

References

1. Estrada, E.: The Structure of Complex Networks. Oxford University Press (2011)
2. Jeong, H., Tombor, B., Albert, R., Oltvai, Z., Barabási, A.: The large-scale organization of metabolic networks. Nature 407, 651–654 (2000)
3. Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., Sakaki, Y.: A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proceedings of the National Academy of Sciences 98, 4569 (2001)
4. Sporns, O.: Network analysis, complexity, and brain function. Complexity 8, 56–60 (2002)
5. Newman, M.: Scientific collaboration networks. i. network construction and fundamental results. Physical Review E 64, 016131 (2001)
6. Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry, 35–41 (1977)
7. Freeman, L.C.: Centrality in social networks conceptual clarification. Social Networks 1, 215–239 (1979)
8. Newman, M.E.: A measure of betweenness centrality based on random walks. Social Networks 27, 39–54 (2005)
9. Bonacich, P.: Power and centrality: A family of measures. American Journal of Sociology, 1170–1182 (1987)
10. Stanley, W., Faust, K.: Social network analysis: methods and applications. Cambridge University, Cambridge (1994)
11. Kempe, J.: Quantum random walks: an introductory overview. Contemporary Physics 44, 307–327 (2003)
12. Lamberti, P., Majtey, A., Borras, A., Casas, M., Plastino, A.: Metric character of the quantum jensen-shannon divergence. Physical Review A 77, 052311 (2008)
13. Lin, J.: Divergence measures based on the shannon entropy. IEEE Transactions on Information Theory 37, 145–151 (1991)
14. Farhi, E., Gutmann, S.: Quantum computation and decision trees. Physical Review A 58, 915 (1998)
15. Nielsen, M.A., Chuang, I.L.: Quantum computation and quantum information. Cambridge University Press, Cambridge (2010)
16. Rossi, L., Torsello, A., Hancock, E.R., Wilson, R.C.: Characterizing graph symmetries through quantum jensen-shannon divergence. Physical Review E 88, 032806 (2013)
17. Lee, C.Y.: Correlations among centrality measures in complex networks. arXiv preprint physics/0605220 (2006)
18. Zachary, W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452–473 (1977)
19. Padgett, J.F., Ansell, C.K.: Robust action and the rise of the medici, 1400-1434. American Journal of Sociology, 1259–1319 (1993)

Max-Correlation Embedding Computation

Antonio Robles-Kelly

NICTA*, Locked Bag 8001, Canberra ACT 2601, Australia
Research School of Eng., ANU, Canberra ACT 0200, Australia

Abstract. In this paper, we present a method to compute an embedding matrix which maximises the dependence of the embedding space upon the graph-vertex coordinates and the incidence mapping of the graph. This treatment leads to a convex cost function which, by construction, attains its maximum at the leading singular value of a matrix whose columns are given by the incidence mapping and the embedded vertex coordinates. This, in turn, maximises the correlation between the spaces in which the embedding and the graph vertex coordinates are defined. It also maximises the dependence between the embedding and the incidence mapping of the graph. We illustrate the utility of the method for purposes of approximating the colour sensitivity functions of a set of over 20 commercially available digital cameras using a library of spectral reflectance measurements.

1 Introduction

In the pattern analysis community, there has recently been renewed interest 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 graph-matching problem into one of point-pattern alignment. The problem is to find matches between pairs of point sets when there is noise, geometric distortion and structural corruption. There is a considerable literature on the problem and many contrasting approaches, including relaxation [4] and optimisation [5], have been attempted. However, the main challenge in graph matching is how to deal with differences in node and edge structure. One of the most elegant recent approaches to the graph matching problem has been to use graph-spectral methods [6], and exploit information conveyed by the eigenvalues and eigenvectors of the adjacency

* 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.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 113–122, 2014.

matrix. For instance, Umeyama [7] has developed a method for finding the permutation matrix which best matches pairs of weighted graphs of the same size, by using a singular value decomposition of the adjacency matrices. Scott and Longuet-Higgins [8], on the other hand, align point-sets by performing singular value decomposition on a point association weight matrix. Shapiro and Brady [9] have reported a correspondence method which relies on measuring the similarity of the eigenvectors of a Gaussian point-proximity matrix. Kosinov and Caelli [10] have improved this method by allowing for scaling in the eigenspace. More recently, Sebastian and Kimia [11] 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 techniques 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 [12] defined an entropic graph-distance for structural graph matching. Christmas, Kittler and Petrou [4] have shown how a relaxation labeling approach can be employed to perform matching using pairwise attributes whose distribution is modelled by a Gaussian. Wilson and Hancock [13] 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 [14].

Here, we focus on the recovery of an embedding matrix based upon the graph and the embedding itself. We do this by maximising the correlation for both, the node-set for the graph and the metric space in which the embedding is defined. To this end, we depart from a cost function which aims at minimising the matrix norm between the embedding and the incidence mapping of the graph. We then rewrite the cost function so as to involve the eigenfunctions of two matrices of inner products. We show the utility of the method presented here for purposes of approximating the spectral sensitivity function of a set of over 20 digital cameras using a library of reflectances of a calibration target, *i.e.* an X-Rite ColorChecker chart.

2 Graph Theory and Spectral Geometry

As mentioned above, we aim at computing a linear mapping that can be used to embed the graph-vertices into a space of finite dimensionality based upon a known transformation to a subspace constrained by the edge space. In this manner, the embedding will reflect the structure of the edge-space of the graph while being based upon a known relationship between the graph vertex-set and the target space Ω. This has two main advantages. Firstly, the target space for the recovered mapping can be used to constrain the embedding. Secondly, note that the mapping sought here embeds the graph vertices using 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 On the Incidence Mapping of Graphs

Here, we aim at recovering a mapping $\mathcal{T}$ which is a matrix whose dimensionality is $\Omega\times \mid \mathcal{V} \mid$. In other words, we aim at recovering an operator which can embed the nodes of a graph G into a space $\Re^{\Omega}$. To commence, we require some formalism. Let $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ denote the graph with node-set $\mathcal{V}_i = \{V_1, \ldots, V_{|\mathcal{V}|}\}$, edge-set $\mathcal{E} = \{e|V_a, V_c \in \mathcal{V}\}$ and attribute-set $\mathcal{A} = \{A_1, \ldots, A_{|\mathcal{V}_i|}\}$.

Here, we view, in general, the vertex-attributes $\mathbf{A}(a)$ 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. 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 $\boldsymbol{D} = diag(deg(1), deg(2), \ldots, deg(|\mathcal{V}|))$ and $deg(c) = \sum_{a=1}^{|\mathcal{V}|} \mathcal{W}(a, c)$ is the degree of the node indexed c in the graph [6].

The use of the graph Laplacian is important, since it permits the use of the incidence mapping. Note that the incidence mapping $\mathcal{I}$ is independent of the orientation of the edges in $\mathcal{E}$. Moreover, it is an operator independent of the vertex-basis, i.e. its permutation invariant [15], which can be recovered via a Young-Householder [16] decomposition on the graph Laplacian such that $\mathcal{L} = \mathcal{I}\mathcal{I}^T$.

2.2 Embedding Computation

With these ingredients, we can formalise the problem as that of recovering the linear mapping $\mathcal{T}$ such that

$$\min_{\mathcal{T}} \left\{ f(\mathcal{T}) \right\} = \min_{\mathcal{T}} \left\{ ||\mathcal{Y} - \mathcal{T}\mathcal{I}||^2 \right\} \tag{1}$$

given the embedding $\mathcal{Y} \in \Re^{\Omega\times|\mathcal{V}|}$ of $\mathcal{V}$ in Ω and $\mathcal{I} \in \Re^{\Gamma\times|\mathcal{V}|}$, as before, is the incidence mapping of G.

It is worth noting in passing that this is akin to point pattern matching settings where the problem is that of finding a transformation which can be used to map the data points onto their counterparts in the model point-set. Nonetheless its similarities, the main difference is that, here, given the coordinates $\mathcal{I}_{v,\gamma}$ and $\mathcal{Y}_{v,\omega}$ of the embeddings and incidence mappings for the node $v \in \mathcal{V}$ in the dimensions $\gamma \in \Gamma$ and $\omega \in \Omega$, we aim at recovering the entries $\phi_{i,j}$ of the matrix $\mathcal{T}$ such that

$$\min_{\mathcal{T}} \left\{ f(\mathcal{T}) \right\} = \min_{\phi_{\omega,\gamma}\in\mathcal{T}} \left\{ \sum_{v\in\mathcal{V}} \left(||\mathcal{Y}_{v,\omega} - \sum_{\gamma\in\Gamma} \phi_{\omega,\gamma}\mathcal{I}_{v,\gamma}||^2 \right) \right\} \tag{2}$$

rather than the corresponding permutation and rotation matrices.

Indeed, the cost function above could be tackled using a least squares solution. This naturally leads to a solution akin to a linear regressor whereby $\phi_{\omega,\gamma}$ can be

viewed as the slope of the lines $\mathcal{Y}_{v,\omega} = \phi_{\omega,\gamma} \sum_{\gamma \in \Gamma} \mathcal{I}_{v,\gamma}$. This can be viewed as a minimisation on the distance about the $\mathcal{Y}_{v,\omega}$ variables [17]. Note that it would be more desirable to use the dintance, i.e. norm, spanned by both, the embedding and the incidence mapping. Thus, we rewrite the cost function above making use of the matrix $\mathbf{M}_{\omega,\gamma} = [\mathcal{Y}_{\cdot,\omega}|\mathcal{I}_{\cdot,\gamma}]$, where $\mathcal{Y}_{\cdot,\omega} = [\mathcal{Y}_{1,\omega}, \mathcal{Y}_{2,\omega}, \ldots, \mathcal{Y}_{|\mathcal{V}|,\omega}]^T$ and $\mathcal{I}_{\cdot,\gamma} = [\mathcal{I}_{1,\gamma}, \mathcal{I}_{2,\gamma}, \ldots, \mathcal{I}_{|\mathcal{V}|,\gamma}]^T$ as follows

$$\min_{\mathcal{T}} \left\{ f(\mathcal{T}) \right\} = \min_{\substack{\xi \in \Omega \\ \psi \in \Gamma}} \left\{ \sum_{v \in \mathcal{V}} \left(||\mathcal{Y}_{v,\omega} - \sum_{\gamma \in \Gamma} \frac{\xi^T \mathbf{M}_{\omega,\gamma} \psi}{\xi^T \psi} \mathcal{I}_{v,\gamma}||^2 \right) \right\} \tag{3}$$

where, by construction, ξ^T and ψ are the eigenvectors of $\mathbf{M}_{\omega,\gamma}^T \mathbf{M}_{\omega,\gamma}$ and $\mathbf{M}_{\omega,\gamma} \mathbf{M}_{\omega,\gamma}^T$, respectively [18].

The advantage of Equation 3 resides in the fact that, as we will see in the following section, the term

$$\phi_{\omega,\gamma} = \max_{\substack{\xi \in \Omega \\ \psi \in \Gamma}} \left\{ \frac{\xi^T \mathbf{M}_{\omega,\gamma} \psi}{\xi^T \psi} \right\}$$

maximises both, the correlation, in the geometric sense, of both, the pairs $\mathcal{Y}_{\cdot,\omega}$, $\mathcal{I}_{\cdot,\gamma}$ and $\mathcal{Y}_{v,\cdot}$, $\mathcal{I}_{v,\cdot}$. This is, it maximises the dependence of the recovered embedding upon the incidence mapping and that of the target space on the vertex coordinates. Moreover, as an added advantage, the computation of $\phi_{\omega,\gamma}$ can be done in a straightforward manner via the application of Singular Value Decomposition (SVD) to the matrix $\mathbf{M}_{\omega,\gamma}$ [18], i.e. $\phi_{\omega,\gamma}$ is the leading singular value of $\mathbf{M}_{\omega,\gamma}$.

2.3 Max-Correlation

Now we examine the link between the cost function above and the eigenvectors ξ and ψ. To this end, we make use of the matrix of scalar products $\mathbf{H} = \mathbf{M}_{\omega,\gamma} \mathbf{M}_{\omega,\gamma}^T$. Note that, since the developments here apply equally to the $\mathbf{M}_{\omega,\gamma}^T \mathbf{M}_{\omega,\gamma}$, we focus on $\mathbf{H}$ throughout the section.

Let ξ_l be the l^{th} eigenvector of $\mathbf{H}$ scaled so its sum of squares is equal to the corresponding eigenvalue τ_l. Since $\mathbf{H}\xi_l = \tau_l \xi_l$ and $(\mathbf{J}\mathbf{J}^T)\xi_l = \mathbf{H}\xi_l$, it follows that the squared distance between a pair of entries in the matrix $\mathbf{H}$ can be written as

$$\| \eta_i - \eta_j \|^2 = \sum_{l=1}^{N} \tau_l (\xi_l(i) - \xi_l(j))^2 = \mathbf{H}(i,i) + \mathbf{H}(j,j) - 2\mathbf{H}(i,j) \tag{4}$$

where η_i and η_j are coordinates in the embedding space such that their inner product corresponds to the entry indexed i, j of $\mathbf{H}$ and N is its rank.

With these ingredients, we can recover the variables ξ_i for the vertices in the graph such that the weighted correlations between their embedding vectors are maximum or minimum by extremasing the quantity

$$\epsilon = \sum_{i,j} \left\| \xi_i \eta_i - \xi_j \eta_j \right\|^2 \tag{5}$$

To take our analysis further, we use Equation 4 and, after some algebra, write

$$\epsilon = \sum_{i,j} \left(\xi_i^2 \mathbf{H}(i,i) + \xi_j^2 \mathbf{H}(j,j) - 2\xi_i \xi_j \mathbf{H}(i,j)\right) \tag{6}$$

Note that, Equation 6 can be divided into two sets of terms, one for the diagonal and the other for the off-diagonal elements of $\mathbf{H}$ as follows

$$\epsilon = 2M \sum_{i} \xi_i^2 \mathbf{H}(i,i) - \sum_{i,j} 2\xi_i \xi_j \mathbf{H}(i,j) \tag{7}$$

where M is the order of $\mathbf{H}$ and we have used the fact that

$$\sum_{i,j} \xi_i^2 \mathbf{H}(i,i) = N \sum_{i} \xi_i^2 \mathbf{H}(i,i) \tag{8}$$

and

$$\sum_{i,j} \xi_i^2 \mathbf{H}(i,i) = \sum_{i,j} \xi_j^2 \mathbf{H}(j,j) \tag{9}$$

Note that maximising the first term in the right-hand side of Equation 7 implies minimising the second one and vice versa. The proof of this hinges in the properties of spectral radii of symmetric matrices [19, 20]. This is also consistent with the work of Chung on isoperimetric inequalities [21]. Thus, we can focus on the term

$$\hat{\epsilon} = -\sum_{\substack{i,j \\ i \neq j}} 2\xi_i \xi_j \mathbf{H}(i,j) \tag{10}$$

Furthermore, to write Equation 10 in compact form, we can define a matrix $\hat{\mathbf{H}}$ which comprises the off-diagonal elements of $\mathbf{H}$ as follows

$$\hat{\mathbf{H}}(i,j) = \begin{cases} \mathbf{H}(i,j) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

and write

$$\hat{\epsilon} = -2\mathbf{\Pi}^T \hat{\mathbf{H}} \mathbf{\Pi} \tag{12}$$

where $\xi = [\xi_1, \xi_2, \cdots, \xi_M]^T$ is a column vector of order M whose ith element is given by ξ_i. Note that the expression above is a Rayleigh quotient. Thus, maximising ϵ is equivalent to minimising $\xi^T \hat{\mathbf{H}} \xi$, which implies that ξ is given by the eigenvector of $\hat{\mathbf{H}}$ which corresponds to the eigenvalue whose rank is the smallest. In this case, ξ is the maximiser of the correlation between the vectors η_i and η_j.

3 Recovering Camera Spectral Sensitivity Functions

In computer vision, video and graphics, we rely upon cameras and rendering contexts to capture and reproduce colour information. Moreover, the accurate

capture and reproduction of colours as acquired by digital camera sensors is an active area of research which has applications in colour correction [22–24], camera simulation [25] and sensor design [26].

To better understand the relation between the spectral radiance and the colour output of digital cameras, recall that we can express the colour output of the detector at pixel u as follows

$$I_k(u) = g(u)\mathbf{S}(u)^T diag(\mathbf{Q}_k)\mathbf{L}, \tag{13}$$

where $I_k(u)$ is the image radiance for any of the three colour channels $k = \{R, G, B\}$ at the pixel u. $\mathbf{S}(u)$ is a vector indexed to wavelength whose entries are given by the surface reflectance $S(\lambda, u)$ at the wavelength λ. $\mathbf{L}$ is the power spectrum of the light with the elements $L(\lambda)$ corresponding to the spectral power at the wavelength λ. $\mathbf{Q}_k$ is a vector whose element $Q_k(\lambda)$ corresponds to the spectral sensitivities of the k^{th} colour sensor at the wavelength λ. When dealing with flat surfaces such as colour charts, we can assume that $g(u) = 1$. This expression has been used widely in the literature [27] and is consistent with reflectance models in computer vision, such as that in [28].

By inspection, it is straightforward to note that, in Equation 13, if the object reflectance and illuminant power spectrum are known, the camera spectral sensitivity functions are, indeed, a linear mapping which "embeds" the product of the reflectance and the illuminant into the colour space. Further, we can view the product of the reflectance and the illuminant as the incidence mapping as presented previously and the ensuing colour triples as the embedding $\mathcal{Y}$. As a result, the matrices $\mathbf{M}_{\omega,\gamma}$ are defined in the colour and wavelength spaces. This is, Ω corresponds to the colour and Γ to the wavelength domain.

In the following experiments, we employ the dataset presented in [29]. This is one of the most complete studies on commercial digital camera spectral responses comprising 28 commercial models[1] Note that the dataset presented in [29] does not contain colour imagery, but rather the sensitivity functions themselves. Thus, for the dataset in [29], we have used the ground-truth power spectrum of the illuminants and the reflectance for each of the colour tiles in a semi-gloss (SG) X-Rite ColorChecker target with 140 colour patches. This is straightforward since the ColorChecker is a flat surface whose mean-scattered power can be easily computed.

3.1 Illuminants

Throughout the section, we use two standard calibrated light sources. This is in line with the standard illuminants defined by the CIE [30]. Our calibrated light sources correspond to the A and D series of illuminants. For our A series illuminant, we have used a tungsten-filament light with a correlated colour temperature (CCT) [31] of 2700^oK. Our D series light is an artificial sunlight with a CCT of 6500^oK (D65).

[1] These can be downloaded from `http://www.cis.rit.edu/jwgu/research/camspec/db.php`

It is worth noting in passing that the use of these two light sources is also aimed at spanning across a wide variety of real-world settings. This is as the A series illuminant correspond to the incandescent filament lights widely used in households and street lighting, whereas the D series illuminant accounts for outdoor environments.

3.2 Reflectance Library

Recall that we also require the spectral reflectance of the color tiles in our X-Rite target so as to compute the covariance matrices used by our method. To this end, we have acquired the reflectance of each colour tile in the X-Rite charts using a StellarNet Bluewave Spectrometer. The spectrometer delivers a spectrum of 1716 samples per tile over the visible and near infrared range. The measurements have been effected using a two-way integrating sphere and a halogen-Deuterium calibrated light source in the $[200nm - 1700nm]$ range.

Note that, for our reflectance library, we have followed the ISO standard for the visible spectrum and archived the reflectance in the range $[400nm, 780nm]$, which yields a total of 599 samples per tile over the 164 colours in the two charts. It is worth stressing in passing that we have opted for the ISO standard over the CIE since the latter is a subset of the former (the CIE standard dictates the visible range is given by the interval $[400nm, 700nm]$) [32]. As mentioned earlier, for our reflectance, we the 140 tiles of 100 different colours including the white ones. The inclusion of the white tile is important since this allows for the illuminant power spectrum computation as required in Equation 13.

3.3 Experiments

With the spectral and colour data in hand, we proceed to provide a quantitative analysis regarding the approximation yielded by our method. To this end, we have used the Euclidean deviation, in degrees, between the sensitivity functions approximated by our method and the corresponding ground truth.

We have used these two metrics so as to account for both, variations in the "shape" of the colour sensitivity functions and power spectrum of the illuminant with respect to the ground truth as well as colour variations induced by the approximation presented here. For our dataset, we have used the colours extracted from the trichromatic imagery of the X-Rite ColourChart as acquired by each camera. For both datasets, we have compared these ground truth data to the colours computed using the sensitivity functions and illuminant power spectrum approximated by our method when applied to the colour checker reflectance library.

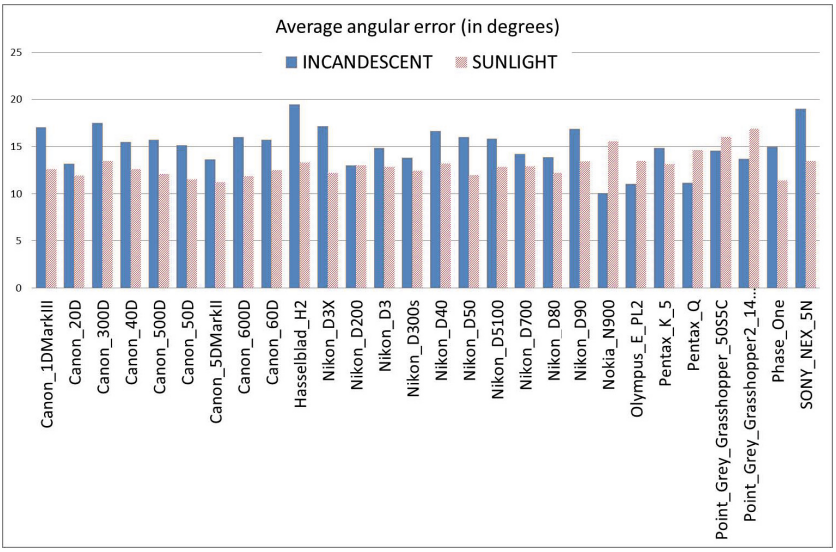
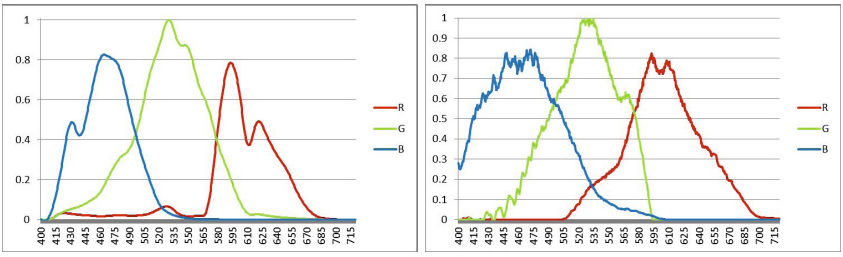


Fig. 1. The average angular error on the spectral responses across three channels for each cameras in the dataset presented in [29]



Nikon D80 (Ground truth) Nikon D80 (Approximation)

Fig. 2. Colour sensitivity functions for a sample camera in our dataset. The two panels show the ground truth and approximated sensitivity functions for the Nikon D80 camera in [29].

Figure 1 shows the Euclidean angular errors, in degrees, for the sensitivity functions corresponding to the cameras in the dataset. Note that the Euclidean angular errors are often in the order of 12 degrees for the dataset in [29].

To illustrate the quality of the approximation in a qualitative manner, in Figure 2 we show the ground truth and approximated colour sensitivity functions a sample camera in the dataset. Note the close accordance of the colour sensitivity functions approximated by our method with respect to the ground truth. Note, however, that even for 16.94 degrees average error yielded by the method presented here, the overall shape approximated by our approach is in good accordance with the ground truth (in the first panel).

4 Conclusions

In this paper, we present a method to compute an embedding which maximises the dependence of the embedding matrix upon the graph-vertex coordinates and that of the target space on the incidence mapping. This treatment leads to a convex cost function whose optimum is attained by the leading singular value of a matrix whose columns are given by the incidence mapping and the embedded vertex coordinates. We illustrate the utility of the method for purposes of approximating the colour sensitivity functions of a set of over 20 commercially available digital cameras from a single image of a colour calibration target. We do this by using a set of spectral reflectance measurements. Thus, our spectral sensitivity recovery via the computation of the corresponding embedding can be viewed as the result of maximising the relationship between the colour values yielded by the camera and the spectra in the reflectance library.

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. Neural Information Processing Systems 14, 634–640 (2002)
4. Christmas, W.J., Kittler, J., Petrou, M.: Structural matching in computer vision using probabilistic relaxation. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(8), 749–764 (1995)
5. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. PAMI 18(4), 377–388 (1996)
6. Chung, F.: Spectral Graph Theory. American Mathematical Society (1997)
7. Umeyama, S.: An eigen decomposition approach to weighted graph matching problems. PAMI 10(5), 695–703 (1988)
8. Scott, G., Longuet-Higgins, H.: An algorithm for associating the features of two images. Proceedings of the Royal Society of London 244(B), 21–26 (1991)
9. Shapiro, L., Brady, J.M.: Feature-based correspondance - an eigenvector approach. Image and Vision Computing 10, 283–288 (1992)
10. Caelli, T., Kosinov, S.: An eigenspace projection clustering method for inexact graph matching. IEEE Trans. on Pattern Analysis and Machine Intelligence 26(4), 515–519 (2004)
11. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Shock-based indexing into large shape databases. In: European Conference on Conputer Vision, vol. 3, pp. 731–746 (2002)
12. Wong, A.K.C., You, M.: Entropy and distance of random graphs with application to structural pattern recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 7, 599–609 (1985)
13. Wilson, R., Hancock, E.R.: Structural matching by discrete relaxation. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(6), 634–648 (1997)
14. Caetano, T., Cheng, L., Le, Q., Smola, A.: Learning graph matching. In: Proceedings of the 11th International Conference on Computer Vision, pp. 14–21 (2007)
15. Biggs, N.L.: Algebraic Graph Theory. Cambridge University Press (1993)

16. Young, G., Householder, A.S.: Discussion of a set of points in terms of their mutual distances. Psychometrika 3, 19–22 (1938)
17. Björck, A.: Numerical methods for least squares problems. SIAM (1996)
18. Golub, G.H., Loan, C.F.V.: Matrix Computations. The Johns Hopkins Press (1996)
19. Torgerson, W.S.: Multidimensional scaling I: Theory and method. Psychometrika 17, 401–419 (1952)
20. Varga, R.S.: Matrix Iterative Analysis, 2nd edn. Springer (2000)
21. Chung, F.: Discrete Isoperimetric Inequalities. Surveys in Differential Geometry IX (2004)
22. Wandell, B.A.: The synthesis and analysis of color images. IEEE Transactions on Pattern Analysis and Machine Intelligence 9(1), 2–13 (1987)
23. Brainard, D.H., Stockman, A.: Colorimetry. McGraw-Hill (1995)
24. Finlayson, G.D., Drew, M.S.: The maximum ignorance assumption with positivity. In: Proceedings of the IS&T/SID 4th Color Imaging Conference, pp. 202–204 (1996)
25. Longere, P., Brainard, D.H.: Simulation of digital camera images from hyperspectral input. In: van den Branden Lambrecht, C. (ed.) Vision Models and Applications to Image and Video Processing, pp. 123–150. Kluwer (2001)
26. Ejaz, T., Horiuchi, T., Ohashi, G., Shimodaira, Y.: Development of a camera system for the acquisition of high-fidelity colors. IEICE Transactions on Electronics E89–C(10), 1441–1447 (2006)
27. Kimmel, R., Elad, M., Shaked, D., Keshet, R., Sobel, I.: A variational framework for retinex. International Journal of Computer Vision 52(1), 7–23 (2003)
28. Finlayson, G.D., Schaefer, G.: Solving for colour constancy using a constrained dichromatic reflection model. International Journal of Computer Vision 42(3), 127–144 (2001)
29. Jiang, J., Liu, D., Gu, J., Süsstrunk, S.: What is the space of spectral sensitivity functions for digital color cameras? In: Workshop on Applications of Computer Vision, pp. 168–179 (2013)
30. Wyszecki, G., Stiles, W.: Color Science: Concepts and Methods, Quantitative Data and Formulae. Wiley (2000)
31. Judd, D.B., Macadam, D.L., Wyszecki, G., Budde, H.W., Condit, H.R., Henderson, S.T., Simonds, J.L.: Spectral distribution of typical daylight as a function of correlated color temperature. Journal of the Optical Society of America 54(8), 1031–1036 (1964)
32. Robles-Kelly, A., Huynh, C.P.: Imaging Spectroscopy for Scene Analysis. Springer (2013)

Fast Gradient Computation for Learning with Tensor Product Kernels and Sparse Training Labels

Tapio Pahikkala

Department of Information Technology, University of Turku
Lemminkäisenkatu 14 A, FIN-20520 Turku, Finland
tapio.pahikkala@utu.fi

Abstract. Supervised learning with pair-input data has recently become one of the most intensively studied topics in pattern recognition literature, and its applications are numerous, including, for example, collaborative filtering, information retrieval, and drug-target interaction prediction. Regularized least-squares (RLS) is a kernel-based learning algorithm that, together with tensor product kernels, is a successful tool for solving pair-input learning problems, especially the ones in which the aim is to generalize to new types of inputs not encountered in during the training phase. The training of tensor kernel RLS models for pair-input problems has been traditionally accelerated with the so-called vec-trick. We show that it can be further accelerated by taking advantage of the sparsity of the training labels. This speed improvement is demonstrated in a running time experiment and the applicability of the algorithm in a practical problem of predicting drug-target interactions.

1 Introduction

In supervised learning, such as regression, classification and ranking, one is given a training data comprised of a sequence $S = \{\mathbf{x}_h\}_{h=1}^{n}$ of inputs and a vector $\mathbf{y} \in \mathbb{R}^n$ consisting of their real-valued labels. Here, we consider learning tasks in which the inputs are paired, a property that is characterized by the following two circumstances. Firstly, the inputs can be naturally split into two parts, in this paper referred to as the data point and task parts, of which both have their own feature representations. Namely, $\mathbf{x} = (\mathbf{d}, \mathbf{t})$, where $\mathbf{d} \in \mathcal{D}$, $\mathbf{t} \in \mathcal{T}$, and $\mathcal{D}$ and $\mathcal{T}$ are sets consisting of all possible tasks and data points, respectively. Secondly, the data with known labels tends to be available in sets, in which both parts of a single input are likely to also appear as parts of other inputs, that is, the input sequence for training contains several inputs associated to the same task part and several inputs associated to the same data point part.

Typical examples of learning problems in which this type of split makes sense can be found, for example, in the fields of recommender systems, where the inputs consist of customers and products (Basilico and Hofmann, 2004), information retrieval, where they consist of queries and data to be retrieved (Liu, 2011),

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 123–132, 2014.

biochemical interaction prediction, where the inputs can be split, for instance, to drugs and targets (see e.g. Ding et al. (2013) for a recent review), prediction of game outcomes (Pahikkala et al., 2010b), and several types of multi-task learning problems involving task-specific features (see e.g. Bonilla et al. (2007); Hayashi et al. (2012)) can be considered under this framework. In these problems, both parts of the input may appear several times in the training set, for example, the same customer may have rated several products and the same product may have been rated by several customers.

Let $D \subset \mathcal{D}$ and $T \subset \mathcal{T}$ denote, respectively, the in-sample data points and in-sample tasks, that is, the sets of data points and tasks encountered in the training set. Given a new input $\mathbf{x} = (\mathbf{d}, \mathbf{t})$, whose label is to be predicted with the model learned from the training set, the above type of learning problems can be coarsely divided into four different settings of varying difficulty, shown in the following table:

$\mathbf{d} \in D$ and $\mathbf{t} \in T$	$\mathbf{d} \in D$ and $\mathbf{t} \notin T$
$\mathbf{d} \notin D$ and $\mathbf{t} \in T$	$\mathbf{d} \notin D$ and $\mathbf{t} \notin T$

Of these, learning problems corresponding to the upper left setting are often encountered in missing value estimation and link prediction problems, where a partially filled matrix needs to be completed without the need for considering new rows and columns, as in collaborative filtering. The upper right and lower left settings can be interpreted as typical multi-task or multi-label learning problems, where the tasks are fixed in advance and the aim is to learn to solve several tasks together, so that the performance in the individual learning tasks is improved compared to the approach in which the individual tasks would be learned in isolation. The lower right setting is usually the most challenging one. Neither the data point nor the task parts are in this case known during training. This paper focuses mainly on this setting, but the proposed algorithms can be straightforwardly applied for any of the above settings.

In this work we consider the setting where both the tasks and data points of interest have feature representations, possibly defined implicitly via a kernel function (Shawe-Taylor and Cristianini, 2004). Previously, learning methods based on the tensor product (of Kronecker product) kernel have been successfully applied in such settings in order to solve problems such as product recommendation (Basilico and Hofmann, 2004; Park and Chu, 2009), prediction of protein-protein interactions (Ben-Hur and Noble, 2005; Kashima et al., 2009).

The pair-input modes based on tensor product kernels can be trained very efficiently with singular value decomposition based approaches (see e.g. Martin and Van Loan (2006); Raymond and Kashima (2010); Pahikkala et al. (2010a, 2013)), if the training set is complete in the sense that it contains every possible datum-task pair with data point in D and task in T exactly once. However, if the training set is not complete, no computationally efficient closed form solutions are known, and one must resort to iterative optimization approaches, such as those based on the conjugate gradient (CG) method.

There has been several articles about accelerating the gradient computation used in these methods, all of which are based on the so-called "vec-trick", which avoids the expensive computation of the tensor product (see e.g. Kashima et al. (2009)). In this paper, we show that the gradient computation can be further accelerated by taking advantage of the sparsity of the training data, that is, only a small subset of the datum-task pairs in $D \times T$ having a known label in training time. This can not be achieved with the standard algorithms and data structures used to implement sparse matrices and computations with them, but we propose new algorithms specially tailored for solving the problem in question.

2 Training Algorithms for Pair-Input Problems

The training data consists of a sequence $S = \{\mathbf{x}_h\}_{h=1}^n \in \mathcal{X}^n$ of inputs, $\mathcal{X}$ being the set of all possible inputs, and a vector $\mathbf{y} \in \mathbb{R}^n$ of the real-valued labels of the inputs. As described above, we assume each input can be represented as a pair consisting of a data point and task $\mathbf{x} = (\mathbf{d}, \mathbf{t}) \in \mathcal{D} \times \mathcal{T}$, where $\mathcal{D}$ and $\mathcal{T}$ are the sets of all possible data points and tasks, respectively, to which we refer as the data point space and the task space. Moreover, let $D \subset \mathcal{D}$ and $T \subset \mathcal{T}$ denote the in-sample data points and in-sample tasks, that is, the sets of data points and tasks encountered in the training sequence, respectively, and let $m = |D|$ and $q = |T|$. We further define

$$\gamma : [n] \to [m] \times [q],$$

where the square bracket notation denotes the set $[n] = \{1, \ldots, n\}$, to be the function that maps the indices of the labeled inputs pairs to the index pairs corresponding to the data point and task the data consist of, that is, $\gamma(h) = (i, j)$ if $\mathbf{x}_h = (\mathbf{d}_i, \mathbf{t}_j)$. Note that γ does not necessarily have an inverse, since in some learning settings the training set may contain several data points with the same datum-task pair.

Next, unless stated otherwise, we assume that the data point space and the task space are real vector spaces, that is, $\mathcal{D} = \mathbb{R}^d$ and $\mathcal{T} = \mathbb{R}^r$, and hence both the data points and tasks have a finite dimensional feature representation. Let $\mathbf{D} \in \mathbb{R}^{m \times d}$ and $\mathbf{T} \in \mathbb{R}^{q \times r}$, respectively, contain the feature representations of the in-sample data points and tasks. Then, the joint tensor feature representation for the training data (used in several studies in the machine learning literature as discussed in Section 1) can be expressed as $\mathbf{X} = \mathbf{B}(\mathbf{T} \otimes \mathbf{D})$, where $\otimes$ denotes the tensor (or Kronecker) product of matrices and $\mathbf{B} \in \{0, 1\}^{n \times mq}$ is a bookkeeping matrix, whose rows are indexed by the n training points and columns by the mq different tensor feature vector combinations, that is, the entries of $\mathbf{B}$ are

$$\mathbf{B}_{h,k} = \begin{cases} 1 & \text{if } k = (j-1)d + i, \text{ where } (i, j) = \gamma(h) \\ 0 & \text{otherwise} \end{cases}.$$

Each row of $\mathbf{B}$ contains a single nonzero entry indicating to which training input the datum corresponds. This matrix covers both the situation in which some of

the possible paired inputs are not in the training data and the one in which there are several occurrences of the same pair. We note that there are several alternative approaches for constructing a joint feature representation for pair-input data but the tensor-based representation is the most expressive one and it enables the simultaneous generalization to both out-of-sample data points and tasks (for further analysis about the expressivity and universality of the tensor-based representation, we refer to our previous work in Waegeman et al. (2012)).

The objective function of the ridge regression problem (Hoerl and Kennard, 1970) can be expressed as

$$J(\mathbf{w}) = (\mathbf{X}\mathbf{w} - \mathbf{y})^{\mathrm{T}}(\mathbf{X}\mathbf{w} - \mathbf{y}) + \lambda \mathbf{w}^{\mathrm{T}}\mathbf{w}, \tag{1}$$

where $\lambda > 0$ is a regularization parameter controlling the trade-off between the regression error made on the training set and the complexity of the model represented by the real-valued vector $\mathbf{w}$. The minimizer of J can be found by solving the following system of linear equations:

$$\left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda \mathbf{I}\right)\mathbf{w} = \mathbf{X}^{\mathrm{T}}\mathbf{y}$$

with respect to $\mathbf{w}$. By substituting the tensor feature representations for $\mathbf{X}$, the system becomes

$$\left((\mathbf{T} \otimes \mathbf{D})^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{B}(\mathbf{T} \otimes \mathbf{D}) + \lambda \mathbf{I}\right)\mathbf{w} = (\mathbf{T} \otimes \mathbf{D})^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{y}. \tag{2}$$

One can also introduce the corresponding optimization problem known as the dual problem of (1), whose solution is obtained via solving the following system

$$\left(\mathbf{X}\mathbf{X}^{\mathrm{T}} + \lambda \mathbf{I}\right)\mathbf{a} = \mathbf{y} \tag{3}$$

with respect to the dual variables $\mathbf{a}$. According to the KKT conditions, the primal and dual solutions are connected as $\mathbf{w} = \mathbf{X}^{\mathrm{T}}\mathbf{a}$. In addition to having computational advantages under certain circumstances more elaborated below, the dual problem also makes it possible to use nonlinear kernel functions in place of the ordinary inner products between feature vectors (Shawe-Taylor and Cristianini, 2004). Note also that, when using kernels, the input space (e.g. here consisting the Cartesian product of the data point space and task space) does not have to be a finite dimensional vector space but, depending of the kernel function used, any kind of set of inputs will do. By introducing the kernel matrices $\mathbf{K} = \mathbf{D}\mathbf{D}^{\mathrm{T}}$ and $\mathbf{G} = \mathbf{T}\mathbf{T}^{\mathrm{T}}$ for the data points and tasks, respectively, (3) can be rewritten as

$$\left(\mathbf{B}(\mathbf{G} \otimes \mathbf{K})\mathbf{B}^{\mathrm{T}} + \lambda \mathbf{I}\right)\mathbf{a} = \mathbf{y}. \tag{4}$$

If one solves the primal system (2) with, for example, the conjugate gradient (CG) algorithm (see e.g. Nocedal and Wright (2000)), it is easy to conclude that the computationally most expensive operations are the following two types of matrix-vector products:

$$\mathbf{u} \leftarrow \mathbf{B}(\mathbf{T} \otimes \mathbf{D})\mathbf{v} \tag{5}$$

$$\mathbf{v} \leftarrow (\mathbf{T} \otimes \mathbf{D})^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{u} \tag{6}$$

where $\mathbf{v} \in \mathbb{R}^{dr}$ and $\mathbf{u} \in \mathbb{R}^n$. Similarly, the computationally most expensive operation involved in a CG step for solving the dual system (4) is the following matrix-vector product:

$$\mathbf{u} \leftarrow \mathbf{B}(\mathbf{G} \otimes \mathbf{K})\mathbf{B}^{\mathrm{T}}\mathbf{u} \tag{7}$$

where $\mathbf{u} \in \mathbb{R}^{mq}$.

The machine learning literature consists of several studies in which these products have been accelerated with the so-called "vec-trick", which is characterized by the following well-known results of the tensor product algebra:

Lemma 1. *Let* $\mathbf{P} \in \mathbb{R}^{a \times b}$, $\mathbf{Q} \in \mathbb{R}^{b \times c}$, *and* $\mathbf{R} \in \mathbb{R}^{c \times d}$ *be matrices. Then,*

$$(\mathbf{R}^{\mathrm{T}} \otimes \mathbf{P})\mathrm{vec}(\mathbf{Q}) = \mathrm{vec}(\mathbf{PQR}), \tag{8}$$

where vec *is the vectorization operator that stacks the columns of a matrix to a vector.*

It is obvious that the right hand size of (8) is considerably faster to compute than the left hand side, because it avoids the direct computation of the large tensor product.

Algorithm 1. Compute $\mathbf{u} \leftarrow \mathbf{B}(\mathbf{T} \otimes \mathbf{D})\mathbf{v}$

```
u ← 0 ∈ R^n
if mdr + rn < qdr + dn then
    M ← DV                                   ▷ O(mdr) time operation
    for h = 1, ..., n do
        i, j ← γ(h)
        u_h ← M_i T_{:,j}                    ▷ O(r) time operation
else
    N ← VT^T                                 ▷ O(qdr) time operation
    for h = 1, ..., n do
        i, j ← γ(h)
        u_h ← D_i N_{:,j}                    ▷ O(d) time operation
return u
```

Let us consider both (5) and (6) in detail. Let $\mathbf{V} \in \mathbb{R}^{d \times r}$ be the matrix for which $\mathbf{v} = \mathrm{vec}(\mathbf{V})$ and $\mathbf{U} \in \mathbb{R}^{m \times q}$ be the matrix for which $\mathbf{B}^{\mathrm{T}}\mathbf{u} = \mathrm{vec}(\mathbf{U})$. Then, applying (8) leads to

$$\mathbf{u} \leftarrow \mathbf{B}\mathrm{vec}(\mathbf{DVT}^{\mathrm{T}}) \tag{9}$$

$$\mathbf{v} \leftarrow \mathrm{vec}(\mathbf{D}^{\mathrm{T}}\mathbf{UT}), \text{ with } \mathbf{B}^{\mathrm{T}}\mathbf{u} = \mathrm{vec}(\mathbf{U}). \tag{10}$$

Similarly, using the vec-trick on (7) transforms it to

$$\mathbf{u} \leftarrow \mathbf{B}\mathrm{vec}(\mathbf{KUG}), \text{ with } \mathbf{B}^{\mathrm{T}}\mathbf{u} = \mathrm{vec}(\mathbf{U}). \tag{11}$$

Multiplying a vector with the matrix $\mathbf{B}$ does not increase the complexity, because it contains at most mq nonzero entries, and hence it can be performed with the standard data structures and algorithms for sparse matrix-vector products. Thus, if we restrict our consideration on only the products between the other matrices, the complexity of the vec-trick method without taking advantage of the sparsity of the label information is characterized by the following lemma:

Lemma 2. *With the vec-trick, the computational complexity of a single gradient step for solving the primal form (e.g. the computation of the right hand sides of both (9) and (10)) and the corresponding complexity for solving the dual form (e.g. the computation of the right hand side of (11)) are, respectively,*

$$O(\min(mdr + mrq, drq + mdq)) \text{ and } O(m^2q + mq^2) .$$

Proof. The complexity results directly from performing the matrix multiplications in the optimal order. □

Solving the primal problem is more cost-effective than solving the dual when the number of features is smaller than the number data points. For pair input data and tensor features, this is the case especially when both $d << m$ and $r << q$ hold simultaneously. In the opposite case, or if nonlinear kernel functions are used, it pays to solve the dual form instead.[1]

Next, we consider how the sparsity of the label information can be taken advantage of to further accelerate the gradient computations for both the primal and dual cases. With sparsity, we refer to the property that only a small portion of the datum-task pairs with the data point and task parts encountered in the training set has a known label. Formally, this means that $n << mq$.

Proposition 1. *The time complexity of computing the right hand sides of both (9) and (10), and the corresponding complexity of computing the right hand side of (11) are, respectively,*

$$O(\min(mdr + rn, drq + dn)) \text{ and } O(mn + qn) .$$

Proof. Calculating the right hand side of (9) can be started by first computing either $\mathbf{DV}$ or $\mathbf{VT}^{\mathrm{T}}$ requiring $O(mdr)$ and $O(qdr)$ time, respectively. Assume that we start with the former, and compute the matrix $\mathbf{M} \leftarrow \mathbf{DV}$. Then, each entry of $\mathbf{u}$ can be computed by taking the inner product between a row of $\mathbf{M}$ and a column of $\mathbf{T}^{\mathrm{T}}$, which are of length r. Since $\mathbf{u}$ has n entries, the overall complexity becomes $O(mdr + rn)$. If the computation is started with the latter way, each entry of $\mathbf{u}$ then requires the inner product between vectors of length d, resulting to an overall complexity $O(qdr + dn)$. This idea is summarized in Algorithm 1.

[1] The convergence properties of gradient descent methods (e.g. the number of steps required for achieving good prediction performance) may differ considerably between the primal and dual forms (Chapelle, 2007), but we leave this consideration out from this article, since it would divert the discussion too far from the scope of the paper.

The matrix $\mathbf{U}$ in (10) contains at most n nonzero entries, and hence computing either the matrix product $\mathbf{D}^{\mathrm{T}}\mathbf{U}$ or $\mathbf{UT}$ require $O(dn)$ and $O(rn)$ time, respectively. The subsequent multiplications of either with $\mathbf{T}$ from right or with $\mathbf{D}^{\mathrm{T}}$ from left, increase the overall complexities to $O(drq + dn)$ or $O(mdr + rn)$ time, respectively. This is illustrated in Algorithm 2.

Algorithm 2. Compute $\mathbf{v} \leftarrow (\mathbf{T} \otimes \mathbf{D})^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{u}$

if $mdr + rn < qdr + dn$ **then**
  $\mathbf{M} \leftarrow \mathbf{0} \in \mathbb{R}^{m \times r}$
  for $h = 1, \ldots, n$ **do**
    $i, j \leftarrow \gamma(h)$
    $\mathbf{M}_i \leftarrow \mathbf{M}_i + \mathbf{u}_h \mathbf{T}_j$ ▷ $O(r)$ time operation
  $\mathbf{v} \leftarrow \mathrm{vec}(\mathbf{D}^{\mathrm{T}}\mathbf{M})$ ▷ $O(mdr)$ time operation
else
  $\mathbf{N} \leftarrow \mathbf{0} \in \mathbb{R}^{d \times q}$
  for $h = 1, \ldots, n$ **do**
    $i, j \leftarrow \gamma(h)$
    $\mathbf{N}_{:,j} \leftarrow \mathbf{N}_{:,j} + (\mathbf{D}^{\mathrm{T}})_{:,i}\mathbf{u}_h$ ▷ $O(r)$ time operation
  $\mathbf{v} \leftarrow \mathrm{vec}(\mathbf{NT})$ ▷ $O(qdr)$ time operation
return $\mathbf{v}$

The matrix $\mathbf{U}$ in (11) has at most n nonzero entries, and hence multiplying it with $\mathbf{K}$ from left or with $\mathbf{G}$ from right take $O(mn)$ and $O(qn)$ time, respectively. The subsequent filling of the entries of $\mathbf{u}$ require n inner products between vectors of size q or m depending whether $\mathbf{U}$ was multiplied with $\mathbf{K}$ or $\mathbf{G}$, resulting in an overall time complexity of $O(mn + qn)$. This is illustrated in Algorithm 3. □

Algorithm 3. Compute $\mathbf{u} \leftarrow \mathbf{B}(\mathbf{G} \otimes \mathbf{K})\mathbf{B}^{\mathrm{T}}\mathbf{u}$

$\mathbf{M} \leftarrow \mathbf{0} \in \mathbb{R}^{m \times q}$
for $h = 1, \ldots, n$ **do**
  $i, j \leftarrow \gamma(h)$
  $\mathbf{M}_i \leftarrow \mathbf{M}_i + \mathbf{u}_h(\mathbf{G})_j$ ▷ $O(q)$ time operation
$\mathbf{u} \leftarrow \mathbf{0} \in \mathbb{R}^n$
for $h = 1, \ldots, n$ **do**
  $i, j \leftarrow \gamma(h)$
  $\mathbf{u}_h \leftarrow \mathbf{K}_i\mathbf{M}_{:,j}$ ▷ $O(m)$ time operation
return $\mathbf{u}$

3 Experiments

In the experiments, we demonstrate the use of the algorithm on a practical problem of predicting drug-target (DT) interactions, and compare the computational speed of the proposed training algorithm based on the one that employs

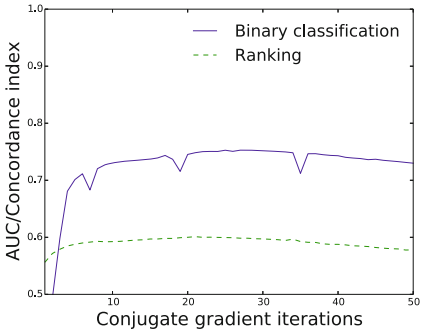


Fig. 1. Prediction performance as a function of CG iterations for both the ranking and classification tasks

the vec-trick only. The data we use for DT interaction prediction experiments consists of 1421 drug compounds, 156 protein targets, and 93356 interaction binding affinity values for DT pairs originally measured by Metz et al. (2011). That is, a bit less than half of the possible DT pairs are labeled with a known binding value. The binding values vary between 4.0 and 10.3, the larger the values the tighter binding. The features of the drugs consists of their 2D Tanimoto coefficient similarities with the other drugs, that is, the feature matrix $\mathbf{D}$ is a symmetric 1421×1421-matrix. The feature representation for the protein targets is their normalized Waterman-Smith sequence similarity with the other targets, resulting to a symmetric 156×156-dimensional feature matrix $\mathbf{T}$. We refer to Pahikkala et al. (2014) for more in depth description of the data and the similarities.[2] The implementation of the algorithm will be put online as a part of the RLScore open source machine learning library.[3]

As practical example problems, we consider the task of learning to rank the DT pairs with respect to their binding value and a binary classification problem in which a drug and target are said to interact if the binding value is larger than 7.6. In both experiments, we perform nine train-test splits of the whole data over which the performance is averaged. The splits reflect the most challenging of the four settings considered in the introduction section, that is, the one in which the model must simultaneously generalize for new drugs and targets. The performance of both learning problems is measured using the concordance index (Gönen and Heller, 2005) (C-index), also known as the pairwise ranking accuracy $\frac{1}{|\{(i,j)|y_i>y_j\}|}\sum_{y_i>y_j} H(\hat{y}_i - \hat{y}_j)$, where y_i denote the true and $\hat{y}_i$ the predicted labels, and H is the Heaviside step function. Note that this measure reduces to the area under ROC curve (AUC) in the binary classification problem. The prediction performances for the tasks are illustrated in Figure 1. The Tikhonov regularization parameter value is set to 0, and hence the only regularization mechanism is the number of CG iterations. We observe that in both tasks one requires only a few CG iterations until the performance converges, to a slightly

[2] The data is avaliable at `http://staff.cs.utu.fi/~aatapa/data/DrugTarget/`
[3] Available at `https://github.com/aatapa/RLScore`

Table 1. The time (in seconds) spent for gradient computations by the proposed accelerated method and the traditional vec-trick based approach

	Drug-target	Simulation
New method	57	0.17
Vec-trick method	67	11.43

better than random level (concordance index 0.6) for the ranking tasks but notable better classification performance (AUC 0.75). These results are in line with those published in our previous study with the data (Pahikkala et al., 2014).

We compare the running speeds of the new and the vec-trick based approach on both the DT interaction prediction problem and with a simulated experiment with randomly generated data. Table 1 presents the running time of both algorithms on 50 CG iterations for the DT problem. Since almost half of the possible DT pairs is known, the training labels are not really sparse and the difference between the running times is small. We next generated artificial data and task similarity matrices $\mathbf{D}, \mathbf{T} \in \mathbb{R}^{10000 \times 100}$ and generated a vector $\mathbf{y} \in \mathbb{R}^{10000}$ labels for inputs with random datum-task indices. For this experiment, the running time of the proposed algorithm for a single gradient iteration is almost two orders of magnitude smaller than that of the vec-trick method, demonstrating the potential of the new approach for large-scale and sparse data sets.

Acknowledgments. We would like to thank the anonymous reviewers for their insightful comments.

References

Basilico, J., Hofmann, T.: Unifying collaborative and content-based filtering. In: Brodley, C.E. (ed.) Proceedings of the Twenty-first International Conference on Machine learning (ICML 2004). ACM (2004)

Ben-Hur, A., Noble, W.: Kernel methods for predicting protein-protein interactions. Bioinformatics 21(suppl. 1), 38–46 (2005)

Bonilla, E.V., Agakov, F.V., Williams, C.K.I.: Kernel multi-task learning using task-specific features. In: Meila, M., Shen, X. (eds.) 11th International Conference on Artificial Intelligence and Statistics. JMLR Proceedings, vol. 2, pp. 43–50. JMLR.org (2007)

Chapelle, O.: Training a support vector machine in the primal. Neural Computation 19(5), 1155–1178 (2007)

Ding, H., Takigawa, I., Mamitsuka, H., Zhu, S.: Similarity-based machine learning methods for predicting drug–target interactions: a brief review. Briefings in Bioinformatics (2013)

Gönen, M., Heller, G.: Concordance probability and discriminatory power in proportional hazards regression. Biometrika 92(4), 965–970 (2005)

Hayashi, K., Takenouchi, T., Tomioka, R., Kashima, H.: Self-measuring similarity for multi-task gaussian process. In: Guyon, I., Dror, G., Lemaire, V., Taylor, G.W., Silver, D.L. (eds.) ICML Unsupervised and Transfer Learning Workshop. JMLR Proceedings, vol. 27, pp. 145–154. JMLR.org (2012)

Hoerl, A.E., Kennard, R.W.: Ridge regression: Biased estimation for nonorthogonal problems. Technometrics 12, 55–67 (1970)

Kashima, H., Kato, T., Yamanishi, Y., Sugiyama, M., Tsuda, K.: Link propagation: A fast semi-supervised learning algorithm for link prediction. In: Proceedings of the SIAM International Conference on Data Mining (SDM 2009), pp. 1099–1110. SIAM (2009)

Liu, T.Y.: Learning to Rank for Information Retrieval. Springer (2011)

Martin, C.D., Van Loan, C.F.: Shifted Kronecker product systems. SIAM Journal on Matrix Analysis and Applications 29(1), 184–198 (2006)

Metz, J.T., Johnson, E.F., Soni, N.B., Merta, P.J., Kifle, L., Hajduk, P.J.: Navigating the kinome. Nature Chemical Biology 7(4), 200–202 (2011)

Nocedal, J., Wright, S.J.: Numerical Optimization, 1st edn. Springer (2000)

Pahikkala, T., Airola, A., Pietilä, S., Shakyawar, S., Szwajda, A., Tang, J., Aittokallio, T.: Toward more realistic drug-target interaction predictions. Briefings in Bioinformatics (in press, 2014), doi:10.1093/bib/bbu010

Pahikkala, T., Airola, A., Stock, M., De Baets, B., Waegeman, W.: Efficient regularized least-squares algorithms for conditional ranking on relational data. Machine Learning 93(2-3), 321–356 (2013)

Pahikkala, T., Waegeman, W., Airola, A., Salakoski, T., De Baets, B.: Conditional ranking on relational data. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010, Part II. LNCS, vol. 6322, pp. 499–514. Springer, Heidelberg (2010)

Pahikkala, T., Waegeman, W., Tsivtsivadze, E., Salakoski, T., De Baets, B.: Learning intransitive reciprocal relations with kernel methods. European Journal of Operational Research 206(3), 676–685 (2010)

Park, S.T., Chu, W.: Pairwise preference regression for cold-start recommendation. In: Proceedings of the Third ACM Conference on Recommender Systems, pp. 21–28. ACM, New York (2009)

Raymond, R., Kashima, H.: Fast and scalable algorithms for semi-supervised link prediction on static and dynamic graphs. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010, Part III. LNCS, vol. 6323, pp. 131–147. Springer, Heidelberg (2010)

Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)

Waegeman, W., Pahikkala, T., Airola, A., Salakoski, T., Stock, M., De Baets, B.: A kernel-based framework for learning graded relations from data. IEEE Transactions on Fuzzy Systems 20(6), 1090–1101 (2012)

Nonlinear Discriminant Analysis Based on Probability Estimation by Gaussian Mixture Model

Akinori Hidaka and Takio Kurita

[1] School of Science and Engineering, Tokyo Denki University, Saitama, Japan
[2] Department of Information Engineering, Hiroshima University, Hiroshima, Japan

Abstract. The Bayesian a posterior probability is a very important element in pattern recognition. In classification problems, the posterior probabilities reflect the uncertainty of assessing an example to particular class. Such residual information will be useful for more deep understanding or analysis of examples. In this paper, we propose a nonlinear discriminant analysis based on the probabilistic estimation of the Gaussian mixture model (GMM). We use GMM to estimate the Bayesian a posterior probabilities of any classification problems. Then we use posterior probabilities estimated by GMM to construct discriminative kernel function. The performance of the proposed kernel function is confirmed by several experiments using UCI machine learning repository.

Keywords: Fisher's Linear Discriminant Analysis, Gaussian Mixture Model, Bayesian a posterior probabilities, Discriminant Kernel.

1 Introduction

The Bayesian *a posterior* probability is a very important element in pattern recognition. The task that classifies unknown example $\boldsymbol{x}$ can be interpreted as the maximization procedure to the posterior probability $P(C_k|\boldsymbol{x})$ which implies the probability that $\boldsymbol{x}$ belongs to the $k-$th class C_k. Furthermore, in classification problems, the posterior probabilities reflect the uncertainty of assessing an example $\boldsymbol{x}$ to the class C_k. Such residual information will be useful for more deep understanding or analysis of examples.

There are many ways to estimate the Bayesian *a posterior* probabilities. Naive Bayes [16] is one of the most simple probabilistic classifier. Logistic regression is a generalized linear model and it has saturated outputs which is suitable to represent probabilities [2]. Several classifiers can also perform the estimation of the posterior probability simultaneously with the classification task. Wu et al. proposed how to presume the posterior probability from the output of SVM [15]. The one of the most efficient methods to estimate the Bayesian *a posterior* probabilities $P(C_k|\boldsymbol{x})$ is to assume the probability densities of each class as multivariate Gaussian distribution. To treat multi-modal distributions, Gaussian mixture model is widely used many real application [3,17].

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 133–142, 2014.

Fisher's Linear discriminant analysis (FLDA) [4] is one of the well known methods to extract the best discriminating features for multi-class classification. FLDA is useful for linear separable cases, but for more complicated cases, it is necessary to extend it to nonlinear.

As one of the nonlinear extensions of FLDA, kernel discriminant analysis (KFDA) has been successfully applied in many applications [9,1]. The polynomial kernel, sigmoidal kernel or radial basis function (RBF) are popular and widely used. However these functions are defined a priori and selected without the clear reason. Also these functions are general and not related to probabilistic inference.

In recent years, discriminant kernel function (DKF) which is based on the Bayesian *a posterior* probability estimation is proposed [8]. This kernel is derived from the theory of optimum nonlinear discriminant analysis (ONDA) [11,12]. Since ONDA gives the optimum nonlinear mapping that maximizes the Fisher's discriminant criterion [4], the DKF derived from ONDA is also optimum in terms of the discriminant criterion. The DKF is defined by explicitly using the Bayesian *a posterior* probability $P(C_k|\boldsymbol{x})$. Similar with the Bayesian decision theory, we have to presume $P(C_k|\boldsymbol{x})$ by a certain estimation method to use DKF for real application.

In this paper, we propose a nonlinear discriminant analysis based on the probabilistic estimation of the Gaussian mixture model. We use GMM to estimate the Bayesian *a posterior* probabilities $P(C_k|\boldsymbol{x})$ of any classification problems. Then we use $P(C_k|\boldsymbol{x})$ estimated by GMM to construct discriminative kernel function which is optimal in terms of the Fisher's discriminant criterion. We call this Gaussian mixture (GM) kernel.

We investigate the performance of the proposed GM kernel by several experiments using UCI machine learning repository [5]. We compare the discriminative power of the discriminant spaces which are constructed from the proposed kernel and usual kernels. The visualization experiments for the discriminant spaces or kernel matrices show some good properties of our discriminant kernels.

The rest of this paper is organized as follows: Section 2 reviews FLDA, KFDA and discriminant kernels. Section 3 reviews Gaussian mixture model. Section 3.2 describes our proposed Gaussian mixture kernel. The experiments are described in Section 4. Finally, Section 5 concludes the paper.

2 Discriminant Analysis

2.1 Fisher's Linear Discriminant

Fisher's linear discriminant analysis (FLDA) [4] is one of the well known methods to extract the best discriminating features for multi-class classification. FLDA is formulated as a problem to find an optimum linear mapping by which the within-class scatter in the mapped discriminant feature space is made as small as possible relative to the between-class scatter.

Let an m dimensional feature vector be $\boldsymbol{x} = (x_1, \ldots, x_m)^T$. Consider K classes denoted by $\{C_1, \ldots, C_K\}$. Assume that we have n feature vectors $\{\boldsymbol{x}_i | i = 1, \ldots, n\}$ as training samples and they are labeled as one of the K classes. Then

FLDA constructs a dimension reducing linear mapping from the input feature vector $\boldsymbol{x}$ to a new feature vector $\boldsymbol{y}$ as

$$\boldsymbol{y} = A^T(\boldsymbol{x} - \bar{\boldsymbol{x}}_T) \tag{1}$$

where $A = [a_{ij}]$ is the coefficient matrix.

The discriminant criterion

$$J = \mathrm{tr}\left(\hat{\Sigma}_W^{-1}\hat{\Sigma}_B\right) \tag{2}$$

is used to evaluate the performance of the discrimination of the new feature vectors $\boldsymbol{y}$, where $\hat{\Sigma}_W$ and $\hat{\Sigma}_B$ are respectively the within-class covariance matrix and the between-class covariance matrix of the new feature vectors $\boldsymbol{y}$. The objective of FLDA is to maximize the discriminant criterion J.

The optimal coefficient matrix A is then obtained by solving the following generalized eigenvalue problem

$$\Sigma_B A = \Sigma_W A \Lambda \quad (A^T \Sigma_W A = I) \tag{3}$$

where $\Lambda = \mathrm{diag}\,(\lambda_1, \ldots, \lambda_L)$ is a diagonal matrix of eigen values and I denotes the unit matrix. The matrices Σ_W and Σ_B are respectively the within-class covariance matrix and the between-class covariance matrix of the input feature vectors $\boldsymbol{x}$, and they are computed as

$$\Sigma_W = \sum_{k=1}^{K} P(C_k)\Sigma_k \tag{4}$$

$$\Sigma_k = \frac{1}{n_k}\sum_{l_i=C_k}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_T)^k \tag{5}$$

$$\Sigma_B = \sum_{k=1}^{K} P(C_k)(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}_T)(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}_T)^T, \tag{6}$$

where n_k, $P(C_k)$, $\bar{\boldsymbol{x}}_k$ and $\bar{\boldsymbol{x}}_T$ denote the number of training samples of the class C_k, *a priori* probability of the class C_k, the mean vector of the class C_k and the total mean vector, respectively. Usually we compute the probability of the class C_k as $P(C_k) = \frac{n_k}{n}$.

The j-th column of A is the eigenvector corresponding to the j-th largest eigenvalue. Therefore, the importance of each element of the new feature vector $\boldsymbol{y}$ is evaluated by the corresponding eigenvalues. The dimension of the new feature vector $\boldsymbol{y}$ is bounded by $\min(K-1, n)$ because the rank of the matrix Σ_B is bounded by $\min(K-1, n)$.

2.2 Kernel Discriminant Analysis

FLDA is useful for linear separable cases, but for more complicated cases, it is necessary to extend it to nonlinear. Kernel discriminant analysis (KFDA) [1,9] is

one of the nonlinear extensions of FLDA and constructs a nonlinear discriminant mapping as a linear combination of kernel functions.

Consider a nonlinear mapping Φ from a input feature vector $\boldsymbol{x}$ to the new feature vector $\Phi(\boldsymbol{x})$. In KFDA the discriminant features $\boldsymbol{y}$ are constructed as a linear combinations of the new feature $\Phi(\boldsymbol{x})$.

The discriminant mapping can be given as

$$\boldsymbol{y}(\boldsymbol{x}) = U^T \Phi(\boldsymbol{x}). \tag{7}$$

Similar with the case of the kernel PCA, the coefficient matrix U can be expressed as a linear combinations of the training samples as

$$U = \sum_{j=1}^{n} \Phi(\boldsymbol{x}_j) \boldsymbol{\alpha}_j^T, \tag{8}$$

the discriminant mapping can be rewritten as

$$\boldsymbol{y}(\boldsymbol{x}) = \sum_{j=1}^{n} \boldsymbol{\alpha}_j \Phi(\boldsymbol{x}_j)^T \Phi(\boldsymbol{x}) = \sum_{j=1}^{n} \boldsymbol{\alpha}_j K(\boldsymbol{x}_j, \boldsymbol{x}) = A^T \boldsymbol{k}(\boldsymbol{x}), \tag{9}$$

where $K(\boldsymbol{x}_i, \boldsymbol{x}) = \Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x})$ and $\boldsymbol{k}(\boldsymbol{x}) = (K(\boldsymbol{x}_1, \boldsymbol{x}), \ldots, K(\boldsymbol{x}_n, \boldsymbol{x}))$ are the kernel function defined by the nonlinear mapping $\Phi(\boldsymbol{x})$ and the empirical kernel vector, respectively.

Then the discriminant criterion is given as

$$J = \mathrm{tr}\left(\hat{\Sigma}_W^{-1} \hat{\Sigma}_B \right), \tag{10}$$

where $\hat{\Sigma}_W$ and $\hat{\Sigma}_B$ are the within-class covariance matrix and the between-class covariance matrix of the new feature vectors $\boldsymbol{y}(\boldsymbol{x})$, respectively.

The polynomial functions

$$K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^T \boldsymbol{y} + 1)^q \tag{11}$$

or the Radial Basis functions

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{y}||^2}{2\sigma^2} \right) \tag{12}$$

are often used as the kernel function for KFDA.

2.3 Discriminant Kernel Functions

In the KFDA, usually the kernel functions are defined a priori and selected without the clear reason. Also such kernel functions are general and not related to the probabilistic inference.

Recently, Kurita proposed the discriminant kernel function (DKF) which is based on the Bayesian *a posterior* probability estimation [8]. This kernel function is defined as

$$K(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{K} \frac{P(C_k|\boldsymbol{x})P(C_k|\boldsymbol{y})}{P(C_k)} \tag{13}$$

where $P(C_k|\boldsymbol{x})$ is the Bayesian *a posterior* probability which is presumed by a certain estimation method, and $P(C_k)$ is the prior of the k-th class C_k.

The Eq. (13) is derived from the theory of optimum nonlinear discriminant analysis (ONDA) [11,12]. Since ONDA gives the optimum nonlinear mapping that maximizes the discriminant criterion, DKF derived from ONDA is also optimum in terms of the discriminant criterion.

As shown in Eq. (13), DKF is defined by using the Bayesian *a posterior* probability $P(C_k|\boldsymbol{x})$. Similar with the Bayesian decision theory, we have to estimate $P(C_k|\boldsymbol{x})$ by a certain estimation method to use DKF for real application. Conversely, DKF can be used as one of the optimal way to construct kernel functions maximizing the discriminant criterion from the Bayesian *a posterior* probability estimation.

There are many ways to estimate the Bayesian *a posterior* probabilities. Depending on the estimation method, we can define the corresponding discriminant kernel function. In this paper we propose discriminant kernel function based on Gaussian mixture model (GMM).

3 Gaussian Mixture Model

Multivariate Gaussian distribution is defined as

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{m/2}|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \right\} \tag{14}$$

where m is a number of variables, $\boldsymbol{\mu}$ is a mean vector and Σ is a covariance matrix.

Gaussian mixture model (GMM) is a linear combination of multiple Gaussian distributions. In GMM, each elemental Gaussian distribution is called component. GMM is formulated as

$$p(\boldsymbol{x}) = \sum_{j=1}^{J} \pi_j \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_j, \Sigma_j) \tag{15}$$

where J is a number of components, $\boldsymbol{\mu}_j$ and Σ_j is a mean vectors and a covariance matrix of the j-th component respectively, and π_j is coefficient of the linear combination.

The parameters $\boldsymbol{\mu}_j, \Sigma_j$ and π_j are usually estimated by Expectation Maximization (EM) algorithm [2].

3.1 The Bayesian *A Posterior* Probability Estimation by GMM

To estimate the Bayesian *a posterior* probability $P(C_k|\boldsymbol{x})$ by GMM, we define the probability densitiy $p(\boldsymbol{x}|C_k)$ of each class C_k as

$$p(\boldsymbol{x}|C_k) = \sum_{j=1}^{J_k} \pi_{k,j} \mathcal{N}_{k,j}(\boldsymbol{x}) \tag{16}$$

where $\mathcal{N}_{k,j}(\boldsymbol{x})$ represetns $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{k,j}, \Sigma_{k,j})$, the j-th Gaussian component for the class C_k. J_k is a number of components for the class C_k. The coefficient $\pi_{k,j}$, the mean vector $\boldsymbol{\mu}_{k,j}$ and the covariance matrix $\Sigma_{k,j}$ are estimated by using given samples $\boldsymbol{x}$ belongs to the class C_k.

Then the posterior probability can be written as

$$P(C_k|\boldsymbol{x}) = \frac{P(C_k)p(\boldsymbol{x}|C_k)}{p(\boldsymbol{x})} = \frac{P(C_k)\sum_{j=1}^{J_k} \pi_{k,j}\mathcal{N}_{k,j}(\boldsymbol{x})}{p(\boldsymbol{x})} \tag{17}$$

where

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} P(C_k)p(\boldsymbol{x}|C_k) = \sum_{k=1}^{K} P(C_k) \sum_{j=1}^{J_k} \pi_{k,j}\mathcal{N}_{k,j}(\boldsymbol{x}). \tag{18}$$

3.2 Gaussian Mixture Kernel

As described in Sec. 2.3, the estimation of the Bayesian *a posterior* probability $P(C_k|\boldsymbol{x})$ can be used to construct the kernel function which is optimum in terms of the discriminant criterion. We obtain kernel function based on Gaussian mixture model by substituting Eq. (17) for Eq. (13):

$$\begin{aligned} K_{GM}(\boldsymbol{x}, \boldsymbol{y}) &= \sum_{k=1}^{K} \frac{P(C_k|\boldsymbol{x})P(C_k|\boldsymbol{y})}{P(C_k)} = \frac{\sum_{k=1}^{K} P(C_k)p(\boldsymbol{x}|C_k)p(\boldsymbol{y}|C_k)}{p(\boldsymbol{x})p(\boldsymbol{y})} \\ &= \frac{\sum_{k=1}^{K} P(C_k) \sum_{i=1}^{J_k} \sum_{j=1}^{J_k} \pi_{k,i}\pi_{k,j}\mathcal{N}_{k,i}(\boldsymbol{x})\mathcal{N}_{k,j}(\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})} \end{aligned} \tag{19}$$

We call this the Gaussian mixture (GM) kernel.

The matrix $\hat{K}$ having the component $k_{mn} = K_{GM}(\boldsymbol{x}_m, \boldsymbol{x}_n)$ is regarded as the kernel matrix of GM kernel. We can perform a novel nonlinear discriminant analysis by applying FLDA to the matrix $\hat{K}$. We call it GMM based kernel discriminant analysis (GM KDA).

After several deformations, Eq. (19) can be rewritten as

$$K_{GM}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{k=1}^{K} P(C_k) \sum_{i=1}^{J_k} \sum_{j=1}^{J_k} \alpha_{k,i,j} \exp\left\{-\frac{M_{k,i}(\boldsymbol{x}) + M_{k,j}(\boldsymbol{y})}{2}\right\}}{p(\boldsymbol{x})p(\boldsymbol{y})}, \tag{20}$$

$$\alpha_{k,i,j} = \frac{\pi_{k,i}\pi_{k,j}}{(2\pi)^D \sqrt{|\Sigma_{k,i}||\Sigma_{k,j}|}}, \tag{21}$$

$$M_{k,i}(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{\mu}_{k,i})^T \Sigma_{k,i}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_{k,i}). \tag{22}$$

Table 1. Specifications of data sets

data set	# of classes	# of samples	# of features
heart	2	270	13
breast cancer	2	683	10
australian	2	690	14
wine	3	178	13
vehicle	4	846	18
vowel	11	990	10

$M_{k,i}(\boldsymbol{x})$ represents the Mahalanobis distance between $\boldsymbol{x}$ and $\boldsymbol{\mu}_{k,i}$, the mean vector of the i-th Gaussian component for the class C_k. Then the proposed kernel function can be interpreted as the sum of exponential of negative averaged Mahalanobis distances.

4 Experiments

The performance of our GMM based nonlinear discriminant analysis is evaluated by using six standard data sets (**heart**, **breast cancer**, **australian**, **wine**, **vehicle** and **vowel**) from UCI Machine Learning Repository [5]. Table 1 shows the statistics of these data sets.

For classification experiments, each data set is divided into a training set (2/3 of all samples) and a test set (remaining samples) at random. A training and testing task is repeated 10 times with different random seeds, and the averaged classification rate for the test sets are shown in the following sections. For all experiments, we used the class prior $P(C_k) = N_k/N$ where N_k is the number of samples in C_k.

4.1 Evaluation of the Number of Components

Gaussian mixture model has the hyper-parameter J which implies the number of Gaussian components. We confirm the relationship between the number of components and classification accuracy. In this section we express the Gaussian mixture model having J components as J-GMM.

For the dataset **heart** and **vowel**, five Gaussian mixture models (1-GMM to 5-GMM) are trained. Each model is used to make GM kernel, and these kernels are used to do the GMM based discriminant analysis.

Tab. 2 shows the training and testing accuracy. Although the performances to the training samples are improving with the number of components, the performances to the test samples are not always increasing.

To avoid the over-fitting problem, we have to reduce the unnecessary components. In this paper, we manually determine the appropriate number of components based on the preliminary experiments. For all classes C_k, we use $J_k = 1$ for **heart**, **breast cancer**, **australian**, **wine**, **vehicle** and use $J_k = 3$ for **vowel**.

Table 2. Relationship between the number of components and classification accuracy

	1-GMM	2-GMM	3-GMM	4-GMM	5-GMM
heart (train)	88.28%	90.11%	92.06%	93.50%	94.06%
heart (test)	81.56%	79.11%	77.22%	75.44%	76.00%
vowel (train)	94.11%	98.56%	99.26%	99.33%	99.38%
vowel (test)	85.67%	91.88%	94.18%	93.15%	94.03%

Table 3. Classification rates (and standard deviations) of 9-NN in discriminant spaces

	Fisher's LDA	RBF KDA	GM KDA (proposed)
heart	81.11% (1.57)	77.56% (8.84)	81.56% (3.02)
breastcancer	97.06% (1.48)	96.40% (1.79)	96.67% (1.51)
australian	85.48% (1.76)	84.87% (1.75)	85.74% (1.28)
wine	98.50% (1.46)	98.17% (2.00)	98.33% (1.76)
vehicle	76.95% (2.06)	84.49% (1.19)	82.45% (1.39)
vowel	75.52% (1.64)	97.03% (1.94)	94.18% (1.97)
Average	85.77% (1.66)	89.75% (2.92)	89.82% (1.82)

4.2 Visualization of Kernels

To compare the property of the proposed and the existing kernel functions, the feature spaces or kernel matrices of the **wine** are illustrated in Fig. 1, 2.

Fig. 1 shows the PCA space of the original features or the discriminant spaces of RBF or GM kernel. It shows a goodness of the proposed kernel. It is noticed that samples of the GM kernel are distributing only on the triangle regions. Generally, for K classes problems, the discriminant spaces of the proposed discriminant kernel forms the $K-1$ dimensional hyper-tetrahedron (simplex) which is expected to be ideal. Since the GM kernel is defined by the Bayesian *a posterior* probabilities, it easily gives a probabilistic interpretation such as how a sample is close to each class. On the other hand, samples of the original features and the RBF kernel are freely and widely distributing in the two dimensional plane.

Fig. 2 shows the visualization result for three types of kernel matrices. The first one is linear kernel; it is constructed from just a inner product of the pair of the original features. Others are the RBF or GM kernel. The color of the i-th row and the j-th column shows the similarity between sample i and j. Since the samples are sorted in order of a class label beforehand, ideally, these matrices should have a block diagonal structure. Such diagonal class structure more clearly appears in the GM kernel than the Linear or the RBF kernel.

4.3 Comparison of Classification Accuracy

We compare the performances of the proposed GMM based discriminant analysis with usual Fisher's Linear Discriminant Analysis (FLDA) and RBF Kernel Discriminant Analysis (RBF KDA). For the classification method in their discriminant spaces, k-nearest neighbor method is adopted. We use $k = 9$ for all dataset and all discriminant spaces.

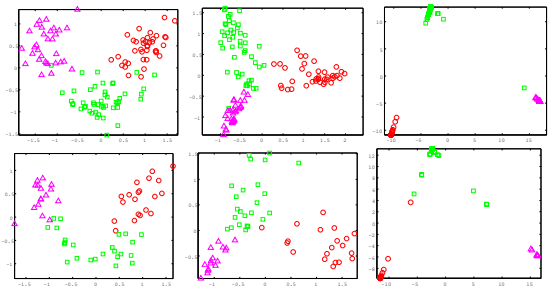


Fig. 1. Sample distributions of **wine** data. The top row and the bottom row show the training and test sets, respectively. (Left) PCA spaces of original features. (Center) Discriminant spaces obtained from RBF kernel matrices. (Right) Discriminant spaces obtained from GM kernel matrices.

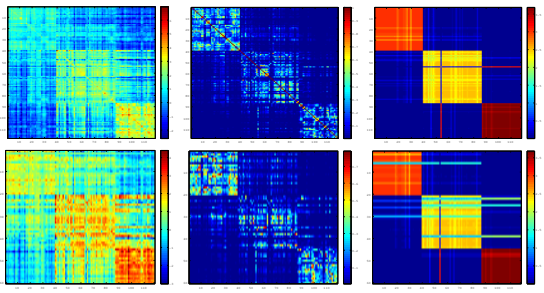


Fig. 2. Visualized kernel matrices of **wine** data. The top row and the bottom row show the results of training and test sets, respectively. (Left) Linear kernel (inner product) of original features. (Center) RBF kernel matrices. (Right) GM kernel matrices.

The parameters of RBF KDA, i.e. the coefficient σ in Eq. (12), are determined by grid search and 10-fold cross validation. We search the best σ from 31 candidates $\sigma = 2^{-15}, 2^{-14}, 2^{-13}, \cdots, 2^{+14}, 2^{+15}$.

Table 3 shows the classification rates for test samples of the proposed and existing methods. The proposed GMM based discriminant analysis has a better performance about averaged accuracy for six datasets. RBF KDA and GM KDA have almost comparable accuracy, but GM KDA shows good (smaller) averaged variance.

5 Conclusion

In this paper we propose GMM based nonlinear discriminant analysis which is formulated by the Bayesian *a posterior* probabilities estimated by Gaussian mixture model. The GM kernel has comparable classification performance with RBF kernel while GM kernel has more good stability (smaller variance).

In the experiment, we manually determined the hyper-parameter J which is the number of individual Gaussian distributions. We shoud automatically

determine J by using cross validation or several statistical validation methods as the future work.

Acknowledgments. This work was supported by JSPS KAKENHI Grant Number 23500211.

References

1. Baudat, G., Anouar, F.: Generalized discriminant analysis using a kernel approach. Neural Computation 12(10), 2385–2404 (2000)
2. Bishop, C.M.: Pattern Recognition and Machine Learning, Springer (2006)
3. Bilmes, J.: A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden markov models. Technical Report ICSI-TR-97-02, University of Berkeley (1997)
4. Fisher, R.A.: The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics 7, 179–188 (1936)
5. Frank, A., Asuncion, A.: UCI Machine Learning Repository. University of California, School of Information and Computer Science
6. Hidaka, A., Kurita, T.: Discriminant Kernels based Support Vector Machine. In: The First Asian Conference on Pattern Recognition (ACPR 2011), Beijing, China, November 28-30, pp. 159–163 (2011)
7. Kurita, T., Watanabe, K., Otsu, N.: Logistic Discriminant Analysis. In: Proc. of 2009 IEEE International Conference on Systems, Man, and Cybernetics, San Antonio, Texas, USA, October 11-14, pp. 2236–2241 (2009)
8. Kurita, T.: Discriminant Kernels derived from the Optimum Nonlinear Discriminant Analysis. In: Proc. of 2011 International Joint Conference on Neural Networks, San Jose, California, USA, July 31-August 5 (2011)
9. Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Smola, A., Muller, K.: Fisher discriminant analysis with kernels. In: Proc. IEEE Neural Networks for Signal Processing Workshop, pp. 41–48 (1999)
10. Nishida, K., Kurita, T.: RANSAC-SVM for Large-Scale Datesets. In: Proc. of International Conference on Pattern Recognition, December 8-11. Tampa Convention Center, Tampa (2008)
11. Otsu, N.: Nonlinear discriminant analysis as a natural extension of the linear case. Behavior Metrika 2, 45–59 (1975)
12. Otsu, N.: Optimal linear and nonlinear solutions for least-square discriminant feature extraction. In: Proceedings of the 6th International Conference on Pattern Recognition, pp. 557–560 (1982)
13. Scholkopf, B., Burges, C.J.C., Smola, A.J.: Advances in Kernel Methods - Support Vector Learning. MIT Press (1999)
14. Viola, P., Jones, M.: Robust real time object detection. In: IEEE ICCV Workshop on Statistical and Computational Theories of Vision (July 2001)
15. Wu, T.F., et al.: Probability Estimates for Multi-class Classification by Pairwise Coupling. Journal of MLR 5, 975–1005 (2004)
16. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29, 103–137 (1997)
17. Reynolds, D.A., Rose, R.C.: Robust text-independent speaker identification using Gaussian mixture speaker models. IEEE Trans. Speech Audio Process. 3, 72–83 (1995)

Information Theoretic Feature Selection in Multi-label Data through Composite Likelihood

Konstantinos Sechidis, Nikolaos Nikolaou, and Gavin Brown

School of Computer Science, University of Manchester, Manchester M13 9PL, UK
{sechidik,nikolaon,gavin.brown}@cs.manchester.ac.uk

Abstract. In this paper we present a framework to unify information theoretic feature selection criteria for multi-label data. Our framework combines two different ideas; expressing multi-label decomposition methods as *composite likelihoods* and then showing how feature selection criteria can be derived by maximizing these likelihood expressions. Many existing criteria, until now proposed as heuristics, can be reproduced from a single basis under the proposed framework. Furthermore we can derive new *problem-specific* criteria by making different independence assumptions over the feature and label spaces. One such derived criterion is shown experimentally to outperform other approaches proposed in the literature on real-world datasets.

1 Introduction

The problem of learning from multi-label data becomes increasingly interesting because of the large number of applications in many different areas [15]. In computer vision [2], multi-label data are used in automated image and video annotation, in situations where images can be associated with a number of semantic concepts. In bioinformatics [5], multi-label learning is used in functional genomics, where a gene or protein is associated with multiple functional labels, as an individual gene or protein usually performs a number of functions. In text mining [8], multi-label data are used in text categorization, as a news webpage can be associated with more than one category.

All of these areas have a common characteristic, a large number of features. High dimensional feature spaces are associated with a number of problems, such as over-fitting to irrelevant features and high computational complexity. The features can be divided in three categories: features that are 'relevant' to our task, features that are 'irrelevant' and features that are 'redundant' in the context of other features. The objective of feature selection is to find a minimal subset of features that provide us with maximal useful information about the data. In our work we focus on *filter methods* for feature selection, which operate under the assumption that the prediction and feature selection steps are independent [7].

More particularly the present work focuses on information theoretic feature selection techniques in multi-label datasets, a problem that has recently received

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 143–152, 2014.

a lot of attention [4,10,9]. The starting point of our work is a recently proposed framework for single label data by Brown et al. [3], which shows that many existing criteria can be seen as iterative maximizers of a common objective function: the conditional likelihood of the true label given the selected features. We extend this work by incorporating the idea of expressing multi-label decomposition methods via composite likelihood, as presented by Zhang & Schneider [16]; we show that this leads naturally to the derivation of different feature selection criteria appropriate for multi-label data. By introducing this framework we provide insights into multi-label feature selection.

There are two main contributions in our work. First, we provide a theoretical foundation that unifies various multi-label criteria proposed in the literature by maximizing full and composite likelihood expressions that describe different independence assumptions over the feature and label spaces (Sections 3-4). Second, we derive and evaluate new multi-label criteria which we compare with the state-of-the-art in real-world datasets (Section 5).

2 Reviewing Likelihood Maximization Framework

In this section we review the single-label feature selection framework presented by Brown et al. [3]. We assume that we have an underlying independent and identically distributed (i.i.d.) process $p : \mathcal{X} \to \mathcal{Y}$, and N samples of this process are observed. The observations are pairs $\{\mathbf{x}^i, y^i\}_{i=1}^N$, where the features are d-dimensional vectors $\mathbf{x}^i = [x_1^i ... x_d^i]$. The features are drawn from the random variables $X_1, ..., X_d$, with their joint distribution being $X = X_1 X_2 ... X_d$ and the labels are drawn from the random variable Y. Following Brown et al. [3], in the feature selection procedure we define θ to be a d-dimensional binary vector, where the elements have a value of 1 if the feature is selected and 0 otherwise. Furthermore $\mathbf{x}_\theta$ is the vector of the chosen and $\mathbf{x}_{\tilde{\theta}}$ the vector of the unchosen features. We assume that the process p can be defined by a subset of features and so for an optimal vector θ^* we have $p(y|\mathbf{x}) = p(y|\mathbf{x}_{\theta^*})$, in other words the unselected features $\mathbf{x}_{\tilde{\theta}^*}$ are irrelevant or redundant given the selected ones. We approximate the process p using a hypothetical predictive model f. This model has two layers of parameters: θ, corresponding to the selected features, and τ, corresponding to the parameters used in the learning procedure in order to predict y. So the problem can be defined as searching for a minimal subset of features, whilst maximizing the conditional likelihood of the training labels. For single-label data the conditional likelihood ($\mathcal{L}$) and the *log*-likelihood (ℓ) have the form:

$$\mathcal{L}(\theta, \tau; y|\mathbf{x}) = \prod_{i=1}^{N} f(y^i|\mathbf{x}_\theta^i, \tau) \Leftrightarrow \ell(\theta, \tau; y|\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} \log f(y^i|\mathbf{x}_\theta^i, \tau).$$

Brown et al. [3] showed that this likelihood decomposes as so:

$$\lim_{N\to\infty} -\ell = E_{XY}\left\{\log \frac{p(y|\mathbf{x}_\theta)}{f(y|\mathbf{x}_\theta, \tau)}\right\} + I(X_{\tilde{\theta}}; Y|X_\theta) + H(Y|X).$$

From the above three terms, the first describes how well the model f approximates p given the selected features, the second term depends on the choice of the selected features, while the third term is an irreducible constant which forms a bound on the Bayes error rate. More details regarding this decomposition can be found in Brown et al. [3]. The main assumption of filter methods is that the classification and the feature selection steps are independent [7]. Under this assumption and ignoring the constant term, the value of θ that maximizes the conditional likelihood is the same as the value of θ that minimizes the conditional mutual information

$$\arg\max_{\theta} \mathcal{L}(\theta; y|\mathbf{x}) = \arg\min_{\theta} I(X_{\tilde{\theta}}; Y|X_{\theta}). \tag{1}$$

As we see in Brown et al. [3] a greedy optimization process to minimize the conditional mutual information in eq. (1) will select a feature X_k that maximizes the following scoring function

$$J_{CMI}(X_k) = I(X_k; Y|X_{\theta}) \quad \text{with} \quad X_k \in X_{\tilde{\theta}}, \tag{2}$$

where the subscript *CMI* stands for *Conditional Mutual Information*.

Since X_{θ} is high-dimensional, the estimates of the mutual information become less reliable as we increase the number of selected features; this can lead to poorly selected subsets. For that reason there have been proposed in the literature low-dimensional approximations of this conditional mutual information, such as the *Mutual Information Maximization (MIM)* [1] and the *Joint Mutual Information maximization (JMI)* [12]. The respective criteria are given by

$$J_{MIM}(X_k) = I(X_k; Y), \qquad J_{JMI}(X_k) = \sum_{j=1}^{|X_{\theta}|} I(X_{\theta_j} X_k; Y),$$

where we used the notation $X_{\theta_j}\ \forall\ j \in \{0, ..., |X_{\theta}|\}$ to represent the j^{th} feature already selected, while $|X_{\theta}|$ is the number of selected features so far. As we can see the *MIM* criterion selects the features independently and so it has the ability to observe relevant features, but not to detect redundant ones. On the other hand the *JMI* also controls the redundancy of the selected features, as it examines the joint random variable $X_{\theta_j} X_k$. Brown et al. [3] present the assumptions made by each approximation, and derive these criteria from first principles by incorporating the assumptions in eq. (2). Furthermore they show in a large empirical study that assuming independence in the feature space (i.e. with *JMI*/*MIM*) has major benefits over the full dependence case of *CMI*. In the following section we will extend the above framework to multi-label data, exploring independence assumptions in the *label* space.

3 Extending the Framework to Multi-label Data

The key difference between single and multi-label classification is that in binary single-label classification, for example, the label space $\mathcal{Y}$ is $\{0, 1\}$, while in multi-label classification the space $\mathcal{Y}$ is $\{0, 1\}^q$ where q represents the number of labels.

The labeling of the i-th instance is a q-dimensional binary vector $\mathbf{y}^i = [y_1^i \dots y_q^i]$, with $y_l^i = 1$ if the example i is positive to the label l and $y_l^i = 0$ if it is negative. The labels are drawn from the random variables $Y_1, \dots, Y_q$ with their joint distribution denoted $Y_{1:q}$.

3.1 Label-Powerset Transformation

When learning from multi-label data, the most general approach is to not assume any label independencies [15]. This transforms the multi-label problem into a multi-class single label one by combining each different label set into a different "meta-class". This approach is known as the *Label Powerset (LP)* transformation, and the maximum number of classes is 2^q. Figure 1 represents the probabilistic graphical model for *LP* transformation, according to the framework presented in Section 2.

The framework presented in Section 2 can be extended to multi-label data just by substituting the single label output variable Y with the multi-label joint random variable $Y_{1:q}$. By making this substitution we arrive at the following multi-label filter:

$$J_{CMI}^{LP}(X_k) = I(X_k; Y_{1:q}|X_\theta). \quad (3)$$

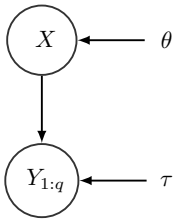


Fig. 1. Label-powerset transformation

The superscript *LP* denotes the assumption over the label space (i.e. none) and the subscript *CMI* stands for the assumptions over the feature space (i.e. also in this case, none). Using the chain rule of mutual information $I(X_k X_\theta; Y_{1:q}) = I(X_\theta; Y_{1:q}) + I(X_k; Y_{1:q}|X_\theta)$ we rewrite the *CMI* criterion as

$$X_k = \arg\max_{X_k \in X_{\bar{\theta}}} I(X_k; Y_{1:q}|X_\theta) = \arg\max_{X_k \in X_{\bar{\theta}}} I(X_k X_\theta; Y_{1:q}),$$

which is *exactly* the multi-label criterion heuristically proposed by Doquire & Verleysen [4]. In our work we derived this criterion by maximising an explicit objective function: the conditional likelihood of the training labels under the probabilistic model presented in Figure 1.

3.2 Binary-Relevance Transformation

The number of distinct label combinations is 2^q, increasing exponentially with the number of labels. Thus we need a large amount of data to have reliable estimates for the probabilities under the *LP* transformation. There have been proposed various transformation approaches to deal with this problem, a detailed exposition of these can be found in Zhang & Zhou [15]. The simplest transformation is to ignore any dependencies between the labels and predict each label independently, this method is known as *Binary Relevance (BR)* or one-vs-all transformation. The graphical model for the *BR* transformation can be seen in

Figure 2. The conditional likelihood for this model, which has been given in Zhang & Schneider [16] in the context of *composite likelihood*, has the form

$$\mathcal{L}_{BR}(\theta, \tau; \mathbf{y}|\mathbf{x}) = \prod_{i=1}^{N} \prod_{l=1}^{q} f(y_l^i | \mathbf{x}_\theta^i, \tau_l).$$

By maximising this likelihood and following the same procedure as in Section 2 we can derive the following multi-label criterion

$$J_{CMI}^{BR}(X_k) = \sum_{l=1}^{q} I(X_k; Y_l | X_\theta). \quad (4)$$

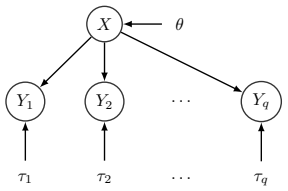


Fig. 2. Binary-relevance transformation

Again, the superscript BR represents the assumption of the conditionally independent labels, while the subscript represents the assumptions made in the feature space. This can be seen as the BR version of eq. (3); to the best of our knowledge this is the first time this has been proposed in the literature.

At this point, we can observe two types of assumption behind different criteria – in the feature space, and in the label space. Table 1 represents the different types of assumption we explore in this work. From now on we will follow this notation to describe the criteria.

Table 1. Choices in the design of multi-label feature selection criteria

		Feature space independence assumptions		
		CMI (none)	JMI (partial)	MIM (full)
Label space independence assumptions	Label Powerset (none)	$J^{\text{Y:none}}_{\text{X:none}}$	$J^{\text{Y:none}}_{\text{X:partial}}$	$J^{\text{Y:none}}_{\text{X:full}}$
	Binary Relevance (full)	$J^{\text{Y:full}}_{\text{X:none}}$	$J^{\text{Y:full}}_{\text{X:partial}}$	$J^{\text{Y:full}}_{\text{X:full}}$

In the following section we will make no assumptions on feature space, and explore the effect of label space assumptions. We will thus compare the criteria described by eqs. (3) and (4), in our new notation $J^{\text{Y:none}}_{\text{X:none}}$ and $J^{\text{Y:full}}_{\text{X:none}}$ respectively.

3.3 Empirical Comparison of the Assumptions in the Label Space

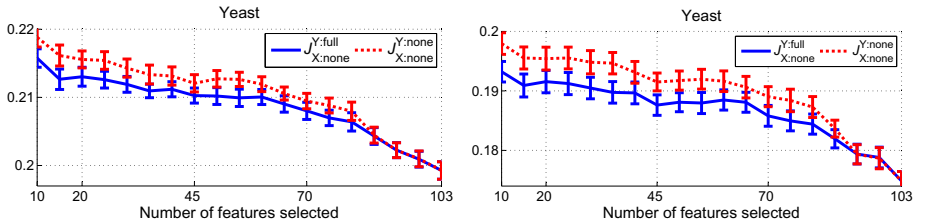
The experiments are performed on real-world multi-label datasets — *yeast* [5] and *scene* [2], taken from two characteristic applications for multi-label data: biology and computer vision, respectively. These two datasets are used by both Doquire & Verleysen [4] and Lee & Kim [9] to evaluate their criteria, with which we compare our own in Section 5. Table 2 summarises some characteristics of these datasets. In order to compare the different feature selection techniques we use a nearest neighbor multi-label classifier, ML-kNN with $k = 7$ as suggested

Table 2. Characteristics of the datasets

Name	Application	Examples	Features	Labels	Distinct labelsets
Scene	Computer Vision	2407	294	6	15
Yeast	Bioinformatics	2417	103	14	198

in Zhang & Zhou [14]. We chose a k-nearest neighbor classifier, since it makes few assumptions and it does not perform implicitly any sort of feature selection, as all the features have the same weight. We evaluate our techniques using two different loss functions: hamming loss and ranking loss [15]. Since in multi-label classification the evaluation is a complex task, we chose these two representative measures. We perform 30 random splits of the data into 50% training and 50% testing, reporting averages and 95% confidence intervals. The training data was used for selecting features and training the ML-kNN classifier, while the testing for measuring the performance of the different approaches. To estimate the mutual information we use maximum likelihood estimates, discretising continuous features into 5 bins using an equal width strategy.

In Figure 3 we compare criteria derived from different label space assumptions, making no assumptions in the feature space. The goal is to investigate the effect that the independence assumptions made on the label space have on the quality of the selected feature subset. As we can see the BR version $\left(J^{\text{Y:full}}_{\text{X:none}}\right)$ very marginally outperforms the LP version $\left(J^{\text{Y:none}}_{\text{X:none}}\right)$ for yeast dataset. This reflects that for yeast, a dataset with large number of distinct labelsets, the benefits of the conditional independence assumption regarding the labels (better probability estimates) outweigh its drawbacks (ignoring inter-label interactions). The effect is slightly more pronounced in the case of ranking loss. Naturally, the difference in performance decreases as we increase the number of selected features. We omit the figures for the scene dataset since both approaches have similar performance, and there is no statistically significant difference between the two criteria. Thus in both datasets, *the quality of the selected feature subset is not significantly affected by the different independence assumptions in the label space.* Since by increasing the number of selected features X_θ the estimates of the conditional mutual information in eq. (3) and (4) degrade, it will be interesting to explore how feature space independence assumptions help the situation.



(a) Hamming loss (b) Ranking loss

Fig. 3. Comparing criteria derived from different label space assumptions and making no feature space assumption. Y:none indicates the LP transformation, while Y:full indicates the BR transformation.

4 Criteria under Different Feature Space Assumptions

The previous section investigated independence assumptions in the label space. In this section we will explore assumptions on the feature space. While in Section 2 we reviewed the *CMI*, *JMI* and *MIM* criteria in the context of single label data, in the current section we will present how the approximate criteria *MIM* and *JMI* are converted in the multi-label context.

4.1 *MIM* and *JMI* Criteria under *LP* Transformation

Under the *LP* transformation, and following a similar procedure to that of Section 2, eq. (3) can be approximated by the lower-order criteria

$$J^{\text{Y:none}}_{\text{X:full}}(X_k) = I(X_k; Y_{1:q}), \quad (5) \qquad J^{\text{Y:none}}_{\text{X:partial}}(X_k) = \sum_{j=1}^{|X_\theta|} I\left(X_k X_{\theta_j}; Y_{1:q}\right). \quad (6)$$

The $J^{\text{Y:none}}_{\text{X:full}}$ criterion has been proposed heuristically by Spolaôr et al. [10].

4.2 *MIM* and *JMI* Criteria under *BR* Transformation

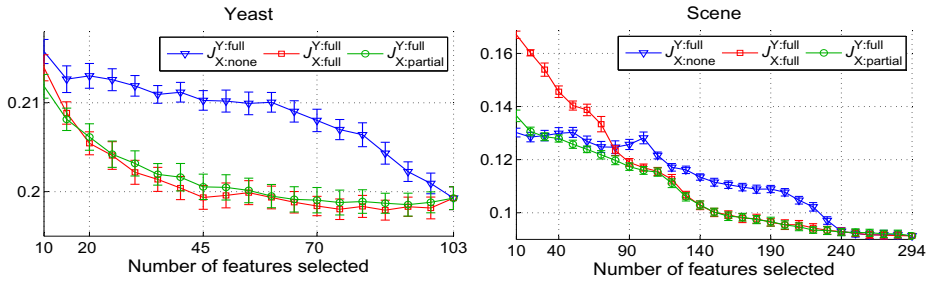
Under the *BR* transformation and following the framework presented in Section 2, eq. (4) can be approximated by the lower-order criteria

$$J^{\text{Y:full}}_{\text{X:full}}(X_k) = \sum_{l=1}^{q} I(X_k; Y_l), \quad (7) \qquad J^{\text{Y:full}}_{\text{X:partial}}(X_k) = \sum_{j=1}^{|X_\theta|} \sum_{l=1}^{q} I\left(X_k X_{\theta_j}; Y_l\right). \quad (8)$$

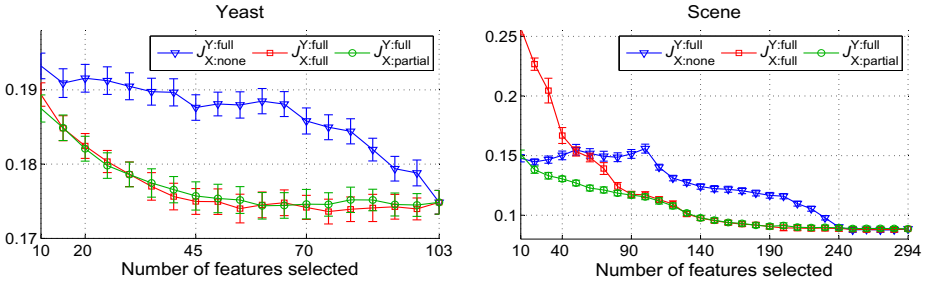
Clearly $J^{\text{Y:full}}_{\text{X:full}}$, makes the most strict assumptions, full independence of both features and labels. As such, this has been suggested heuristically in numerous works [10,11,13]. Here we have shown that this can be derived as an approximate maximizer of the composite likelihood of the model in Figure 2. However this is the first time that *JMI* criteria, such as $J^{\text{Y:none}}_{\text{X:partial}}$ and $J^{\text{Y:full}}_{\text{X:partial}}$, are introduced in the multi-label setting.

4.3 Empirical Comparison of the Assumptions in the Feature Space

Figure 4 compares criteria derived from different feature space assumptions under the same experimental setup we used in Section 3. This comparison was performed under the *BR* transformation but the results under the *LP* transformation are similar. The goal now is to investigate which independence assumption on the feature space gives the best feature selection results. We see that the JMI criterion $\left(J^{\text{Y:full}}_{\text{X:partial}}\right)$ outperforms the other two approaches as it consistently achieves good performance for both datasets. On the yeast dataset the JMI and MIM perform similarly, and we can draw the same conclusion as in Doquire & Verleysen [4], i.e. that the relevant features are non-redundant in this data. On the scene dataset JMI outperforms the other criteria for almost any number of selected features in the cases of hamming loss, and for any number of selected features for ranking loss.



(a) Hamming loss



(b) Ranking loss

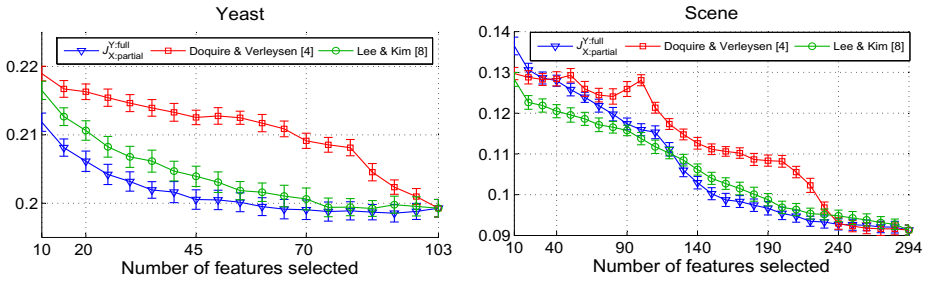
Fig. 4. Comparing criteria derived from different feature space assumptions and assuming full independence in the label space. X:none indicates the CMI criterion, X:partial the JMI and X:full the MIM.

5 Summary and Connections to Literature

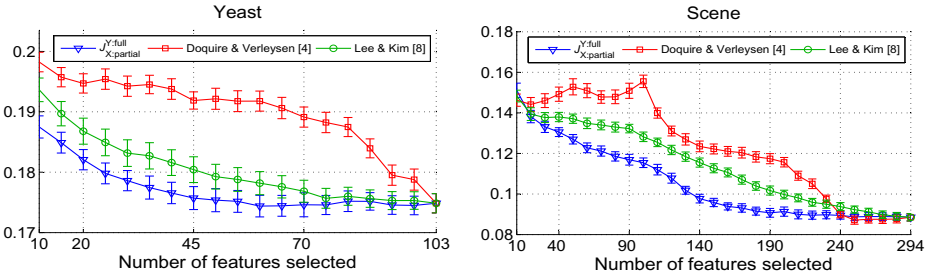
In Section 3.3 we examined the effect of the label space assumptions on the feature selection process and found that *BR* has a marginal advantage over *LP*. In Section 4.3 we investigated the effect of the feature space assumptions and observed an advantage of *JMI* over *CMI* and to a lesser extent over *MIM*. In this section we connect our work with the literature, and we compare the criterion with the best performance under our analysis, with the state-of-the-art in multi-label feature selection.

5.1 Connections with the Literature

Yang & Pedersen [13] introduced the first multi-label feature selection criteria which can be classified as the $J^{\text{Y:full}}_{\text{X:full}}$ of our analysis. Trohidis et al. [11] present a comparison between the $J^{\text{Y:full}}_{\text{X:full}}$ and $J^{\text{Y:none}}_{\text{X:full}}$ criteria, but using χ^2-statistic instead of mutual information, while recently these criteria were re-introduced under the problem transformation approach [10]. Doquire & Verleysen [4] proposed $J^{\text{Y:none}}_{\text{X:none}}$. In order to produce better estimates they use a nearest neighbor mutual information estimator and they apply the pruned problem transformation technique, under which the rare label combinations are discarded, and as a consequence this leads to some loss of information. Finally, Lee & Kim [9] propose the use of multivariate mutual information for selecting features in a criterion without applying



(a) Hamming loss



(b) Ranking loss

Fig. 5. Comparing the $J^{\text{Y:full}}_{\text{X:partial}}$ criterion with criteria proposed in the literature

any transformation, but since this method is computationally inefficient they propose an approximate solution which involves only three variables.

5.2 Comparison to the State-of-the-Art

We compare $J^{\text{Y:full}}_{\text{X:partial}}$, the criterion with the best performance under our analysis, with two different criteria proposed recently in the literature: the pruned transformation criterion proposed by Doquire & Verleysen [4] (we prune rare examples using thresholds given in that work) and the multi-variate mutual information criterion proposed by Lee & Kim [9]. As we can see in Figure 5 the proposed criterion $J^{\text{Y:full}}_{\text{X:partial}}$ consistently performs well across the different number of selected features and the different datasets. On the yeast dataset it has the best performance for both loss functions and all numbers of selected features. On the scene dataset it outperforms the other techniques in all areas apart from 10-130 selected features under hamming loss. However, in terms of ranking loss it continuously outperforms the other criteria.

6 Conclusions

We have provided a theoretical justification for multi-label feature selection criteria. Our framework introduces the idea of *maximizing the conditional composite likelihood expression for multi-label decompositions*. Different assumptions lead

naturally to different filters, some of which have been heuristically proposed in the literature, while others are novel. In our experiments we explored how different assumptions of feature/label space compare. The best trade-off appears to be assuming partial independence in feature space, and full independence in label space. Our observation regarding the label space assumptions agrees with recent empirical results in the context of wrapper feature selection [6]. The corresponding filter we propose is shown to outperform the state-of-the-art approaches on real-world datasets. Finally, under this framework we can incorporate assumptions that explicitly encode domain knowledge, leading to filters specialised for particular problems.

Acknowledgments. This work was supported by EPSRC grant [EP/I028099/1]. Sechidis gratefully acknowledges the support of the Propondis Foundation.

References

1. Battiti, R.: Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on Neural Networks 5(4), 537–550 (Jul 1994)
2. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. Pattern Recognition 37(9), 1757 – 1771 (2004)
3. Brown, G., Pocock, A., Zhao, M., Lujan, M.: Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. Journal of Machine Learning Research (JMLR) 13, 27–66 (2012)
4. Doquire, G., Verleysen, M.: Mutual information-based feature selection for multilabel classification. Neurocomputing 122, 148 – 155 (2013)
5. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Advances in Neural Inf. Processing Systems (NIPS) 14. pp. 681–687 (2001)
6. Gharroudi, O., Elghazel, H., Aussem, A.: A comparison of multi-label feature selection methods using the random forest paradigm. In: Adv. in Artificial Intelligence, Lecture Notes in Computer Science, vol. 8436, pp. 95–106. Springer (2014)
7. Guyon, I.M., Gunn, S.R., Nikravesh, M., Zadeh, L. (eds.): Feature Extraction: Foundations and Applications. Springer, 1st edn. (2006)
8. Katakis, I., Tsoumakas, G., Vlahavas, I.: Multilabel text classification for automated tag suggestion. In: ECML/PKDD Workshop on Discovery Challenge (2008)
9. Lee, J., Kim, D.W.: Feature selection for multi-label classification using multivariate mutual information. Pattern Recognition Letters 34(3), 349 – 357 (2013)
10. Spolaôr, N., Cherman, E.A., Monard, M.C., Lee, H.D.: A comparison of multi-label feature selection methods using the problem transformation approach. Electronic Notes in Theoretical Computer Science 292, 135 – 151 (2013)
11. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multilabel classification of music into emotions. In: 9th Int. Conf. on Music Inf. Retrieval (ISMIR) (2008)
12. Yang, H.H., Moody, J.: Data visualization and feature selection: New algorithms for nongaussian data. In: Advances in Neural Inf. Processing Systems (NIPS) (1999)
13. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: 14th Int. Conference on Machine Learning (ICML) (1997)
14. Zhang, M.L., Zhou, Z.H.: A k-nearest neighbor based algorithm for multi-label classification. In: IEEE International Conference on Granular Computing (2005)
15. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE Transactions on Knowledge and Data Engineering (in press) (2013)
16. Zhang, Y., Schneider, J.: A composite likelihood view for multi-label classification. In: 15th Int. Conference on Artificial Intelligence and Statistics (AISTATS) (2012)

Majority Vote of Diverse Classifiers for Late Fusion

Emilie Morvant[1], Amaury Habrard[2], and Stéphane Ayache[3]

[1] Institute of Science and Technology (IST) Austria, A-3400 Klosterneuburg, Austria
[2] Université de Saint-Etienne, CNRS, LaHC, UMR 5516, F-42000 St-Etienne, France
[3] Aix-Marseille Université, CNRS, LIF UMR 7279, F-13000, Marseille, France

Abstract. In the past few years, a lot of attention has been devoted to multimedia indexing by fusing multimodal informations. Two kinds of fusion schemes are generally considered: The *early fusion* and the *late fusion*. We focus on late classifier fusion, where one combines the scores of each modality at the decision level. To tackle this problem, we investigate a recent and elegant well-founded quadratic program named MinCq coming from the machine learning PAC-Bayesian theory. MinCq looks for the weighted combination, over a set of real-valued functions seen as voters, leading to the lowest misclassification rate, while maximizing the voters' diversity. We propose an extension of MinCq tailored to multimedia indexing. Our method is based on an order-preserving pairwise loss adapted to ranking that allows us to improve Mean Averaged Precision measure while taking into account the diversity of the voters that we want to fuse. We provide evidence that this method is naturally adapted to late fusion procedures and confirm the good behavior of our approach on the challenging PASCAL VOC'07 benchmark.

Keywords: Multimedia analysis, Classifier fusion, Majority vote, Ranking.

1 Introduction

Combining multimodal information is an important issue in pattern recognition. Indeed, the fusion of multimodal inputs can bring complementary information from various sources, useful for improving the quality of the final decision. In this paper, we focus on multimodal fusion for image analysis in multimedia systems (see [1] for a survey). The different modalities correspond generally to a relevant set of features that can be grouped into views. Once these features have been extracted, another step consists in using machine learning methods in order to build voters—or classifiers—able to discriminate a given concept. In this context, two main schemes are generally considered [17]. On the one hand, in the *early fusion* approach, all the available features are merged into one feature vector before the learning and classification steps. This can be seen as a unimodal classification. However, this kind of approach has to deal with many heterogeneous features which are sometimes hard to combine. On the other hand, the *late fusion*[1] works at the decision level by combining the prediction scores available for each modality (see Fig. 1). Even if late fusion may not always outperform early fusion[2], it tends to give better results in multimedia analysis [17]. Several methods based on a fixed

[1] The late fusion is sometimes called multimodal classification or classifier fusion.
[2] For example, when one modality provides significantly better results than others.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 153–162, 2014.

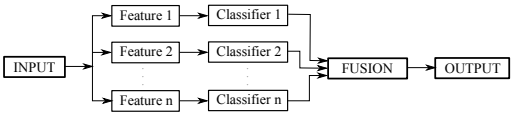


Fig. 1. Classical late classifier fusion scheme

decision rule have been proposed for combining classifiers such as `max`, `min`, `sum`, etc [9]. Other approaches, often referred to as *stacking* [20], need of an extra learning step.

In this paper, we address the problem of *late fusion* with stacking. Let h_i be the function that gives the score associated with the i^{th} modality for any instance $\mathbf{x}$. A classical method consists in looking for a weighted linear combination of the scores seen as a majority vote and defined by: $H(\mathbf{x})=\sum_{i=1}^{n} q_i h_i(\mathbf{x})$, where q_i is the weight associated with h_i. This approach is widely used because of its robustness, simplicity and scalability due to small computational costs [1]. It is also more appropriate when there exist dependencies between the views through the classifiers [21,14]. The objective is then to find an optimal combination of the classifiers' scores by taking into account these dependencies. One solution is to use machine learning methods to assess automatically the weights [10,4,16,18]. Indeed, from a theoretical machine learning standpoint: considering a set of classifiers with a high diversity is a desirable property [4]. One illustration is given by the algorithm AdaBoost [7] that weights *weak classifiers* according to different distributions of the training data, introducing some diversity. However, AdaBoost degrades the fusion performance when combining strong classifiers [19].

To tackle the late fusion by taking into account the diversity between score functions of strong classifiers, we propose a new framework based on a recent machine learning algorithm called MinCq [12]. MinCq is expressed as a quadratic program for learning a weighted majority vote over real-valued functions called voters (such as score functions of classifiers). The algorithm is based on the minimization of a generalization bound that takes into account both the risk of committing an error and the diversity of the voters, offering strong theoretical guarantees on the learned majority vote. In this article, our aim is to show the usefulness of MinCq-based methods for classifier fusion. We provide evidence that they are able to find good linear weightings, and also performing non-linear combination with an extra kernel layer over the scores. Moreover, since in multimedia retrieval, the performance measure is related to the rank of positive examples, we extend MinCq to optimize the Mean Average Precision. We base this extension on an additional order-preserving loss for verifying ranking pairwise constraints.

The paper is organized as follows. The framework of MinCq is introduced in Section 2. Our extension for late classifier fusion is presented in Section 3 and it is evaluated on an image annotation task in Section 4. We conclude in Section 5.

2 MinCq: A Quadratic Program for Majority Votes

We start from the presentation of MinCq [12], a quadratic program for learning a weighted majority vote of real-valued functions for binary classification. Note that this method is based on the machine learning PAC-Bayesian theory, first introduced in [15].

We consider binary classification tasks over a *feature space* $X \subseteq \mathbb{R}^d$ of dimension d. The *label space* (or output space) is $Y = \{-1, 1\}$. The training sample of size m is $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ where each example $(\mathbf{x}_i, y_i)$ is drawn *i.i.d.* from a fixed—but unknown—probability distribution $\mathcal{D}$ defined over $X \times Y$. We consider a set of n real-valued voters $\mathcal{H}$, such that: $\forall h_i \in \mathcal{H},\ h_i : X \mapsto \mathbb{R}$. Given a voter $h_i \in \mathcal{H}$, the predicted label of $\mathbf{x} \in X$ is given by $\text{sign}[h_i(\mathbf{x})]$, where $\text{sign}[a] = 1$ if $a \geq 0$ and -1 otherwise. Then, the learner aims at choosing the weights q_i, leading to the *$\mathcal{Q}$-weighted majority vote* $B_\mathcal{Q}$ with the lowest risk. In the specific setting of MinCq[3], $B_\mathcal{Q}$ is defined by,

$$B_\mathcal{Q}(\mathbf{x}) = \text{sign}\left[H_\mathcal{Q}(\mathbf{x})\right], \text{ with } H_\mathcal{Q}(\mathbf{x}) = \sum_{i=1}^{n} q_i h_i(\mathbf{x}),$$

where $\forall i \in \{1, \ldots, n\},\ \sum_{i=1}^n |q_i| = 1$ and $-1 \leq q_i \leq 1$. Its true risk $R_\mathcal{D}(B_\mathcal{Q})$ is defined as the probability that $B_\mathcal{Q}$ misclassifies an example drawn according to $\mathcal{D}$,

$$R_\mathcal{D}(B_\mathcal{Q}) = \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\left(B_\mathcal{Q}(\mathbf{x}) \neq y\right).$$

The core of MinCq is the minimization of the empirical version of a bound—the C-Bound [11,12]—over $R_\mathcal{D}(B_\mathcal{Q})$. The C-Bound is based on the notion of $\mathcal{Q}$-margin, which is defined for every example $(\mathbf{x}, y) \sim \mathcal{D}$ by: $yH_\mathcal{Q}(\mathbf{x})$, and models the confidence on its label. Before expounding the C-Bound, we introduce the following notations respectively for the first moment $\mathcal{M}_\mathcal{Q}^\mathcal{D}$ and for the second moment $\mathcal{M}_{\mathcal{Q}^2}^\mathcal{D}$ of the $\mathcal{Q}$-margin,

$$\begin{aligned}
\mathcal{M}_\mathcal{Q}^\mathcal{D} &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D} yH_\mathcal{Q}(\mathbf{x}) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D} \sum_{i=1}^{n} y q_i h_i(\mathbf{x}), \\
\mathcal{M}_{\mathcal{Q}^2}^\mathcal{D} &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \left(yH_\mathcal{Q}(\mathbf{x})\right)^2 = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \sum_{i=1}^{n}\sum_{i'=1}^{n} q_i q_{i'} h_i(\mathbf{x}) h_{i'}(\mathbf{x}). \qquad (1)
\end{aligned}$$

By definition, $B_\mathcal{Q}$ correctly classifies an example $\mathbf{x}$ if the $\mathcal{Q}$-margin is strictly positive. Thus, under the convention that if $y\mathbb{E}_{h\sim\mathcal{Q}}\, h(\mathbf{x}) = 0$, then $B_\mathcal{Q}$ errs on $(\mathbf{x}, y)$, we have:

$$\forall D \text{ over } X \times Y,\ R_\mathcal{D}(B_\mathcal{Q}) = \mathop{\mathbf{Pr}}_{(\mathbf{x},y)\sim D}\Big(\mathop{\mathbb{E}}_{h\sim\mathcal{Q}} yh(\mathbf{x}) \leq 0\Big).$$

Knowing this, the authors of [11,12] have proven the following C-bound over $R_D(B_\mathcal{Q})$ by making use of the Cantelli-Chebitchev inequality.

Theorem 1 (The C-bound). *Given $\mathcal{H}$ a class of n functions, for any weights $\{q_i\}_{i=1}^n$, and any distribution $\mathcal{D}$ over $X \times Y$, if $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} H_\mathcal{Q}(\mathbf{x}) > 0$ then $R_\mathcal{D}(B_\mathcal{Q}) \leq C_\mathcal{Q}^\mathcal{D}$ where,*

$$C_\mathcal{Q}^\mathcal{D} = \frac{\mathbf{Var}_{(\mathbf{x},y)\sim\mathcal{D}}(yH_\mathcal{Q}(\mathbf{x}))}{\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left(yH_\mathcal{Q}(\mathbf{x})\right)^2} = 1 - \frac{(\mathcal{M}_\mathcal{Q}^\mathcal{D})^2}{\mathcal{M}_{\mathcal{Q}^2}^\mathcal{D}}.$$

In the supervised binary classification setting, [12] have then proposed to minimize the empirical counterpart of the C-bound for learning a good majority vote over $\mathcal{H}$, justified by an elegant PAC-Bayesian generalization bound. Following this principle the authors have derived the following quadratic program called MinCq.

[3] In PAC-Bayes these weights are modeled by a distribution $\mathcal{Q}$ over $\mathcal{H}$ s.t. $\forall h_i \in \mathcal{H},\ q_i = \mathcal{Q}(h_i)$.

$$\text{argmin}_{\mathbf{Q}} \quad \mathbf{Q}_S^t \mathbf{M}_S \mathbf{Q} - \mathbf{A}_S^t \mathbf{Q}, \tag{2}$$

$$\text{s.t. } \mathbf{m}_S^t \mathbf{Q} = \frac{\mu}{2} + \frac{1}{nm} \sum_{j=1}^{m} \sum_{i=1}^{n} y_j h_i(\mathbf{x}_j), \tag{3}$$

$$\text{and } \forall i \in \{1, \ldots, n\},\ 0 \leq q'_i \leq \tfrac{1}{n}, \tag{4}$$

$$(MinCq)$$

where ^{t} is the transposed function, $\mathbf{Q} = (q'_1, \ldots, q'_n)^t$ is the vector of the first n weights q_i, $\mathbf{M}_S$ is the $n \times n$ matrix formed by $\frac{1}{m} \sum_{j=1}^{m} h_i(\mathbf{x}_j) h_{i'}(\mathbf{x}_j)$ for (i, i') in $\{1, \ldots, n\}^2$, $\mathbf{A}_S = \left(\frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} h_1(\mathbf{x}_j) h_i(\mathbf{x}_j), \ldots, \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} h_n(\mathbf{x}_j) h_i(\mathbf{x}_j) \right)^t$, and, $\mathbf{m}_S = \left(\frac{1}{m} \sum_{j=1}^{m} y_j h_1(\mathbf{x}_j), \ldots, \frac{1}{m} \sum_{j=1}^{m} y_j h_n(\mathbf{x}_j) \right)^t$.

Finally, the majority vote learned by MinCq is $B_{\mathcal{Q}}(\mathbf{x}) = \text{sign}[H_{\mathcal{Q}}(\mathbf{x})]$, with,

$$H_{\mathcal{Q}}(\mathbf{x}) = \sum_{i=1}^{n} \underbrace{\left(2q'_i - \tfrac{1}{n}\right)}_{q_i} h_i(\mathbf{x}).$$

Concretely, MinCq minimizes the denominator of the C-bound (Eq. (2)), given a fixed numerator, *i.e.* a fixed $\mathcal{Q}$-margin (Eq. (3)), under a particular regularization (Eq. (4))[4]. Note that, MinCq has showed good performances for binary classification.

3 A New Framework for Classifier Late Fusion

MinCq stands in the particular context of machine learning binary classification. In this section, we propose to extend it for designing a new framework for multimedia late fusion. We actually consider two extensions for dealing with ranking, one with pairwise preferences and a second based on a relaxation of these pairwise preferences to lighten the process. First of all, we discuss in the next section the usefulness of MinCq in the context of multimedia late fusion.

3.1 Justification of MinCq as a Classifier Late Fusion Algorithm

It is well known that diversity is a key element in the success of classifier combination [1,10,4,6]. From a multimedia indexing standpoint, fuzing diverse voters is thus necessary for leading to good performances. We quickly justify that this is exactly what MinCq does by favoring majority votes with maximally uncorrelated voters.

In the literature, a general definition of diversity does not exist. However, there are popular diversity metrics based on pairwise difference on every pair of individual classifiers, such as Q-statistics, correlation coefficient, disagreement measure, *etc.* [10,13] We consider the following diversity measure assessing the disagreement between the predictions of a pair of voters according to the distribution $\mathcal{D}$,

$$\text{diff}_{\mathcal{D}}(h_i, h_{i'}) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} h_i(\mathbf{x}) h_{i'}(\mathbf{x}).$$

[4] For more technical details on MinCq please see [12].

We then can rewrite the second moment of the $\mathcal{Q}$-margin (see Eq.(1)),

$$\mathcal{M}^{\mathcal{D}}_{\mathcal{Q}^2} = \sum_{i=1}^{n}\sum_{i'=1}^{n} q_i q_{i'}\, \mathrm{diff}_{\mathcal{D}}(h_i, h_{i'}). \tag{5}$$

The first objective of MinCq is to reduce this second moment, implying a direct optimization of Eq. (5). This implies a maximization of the diversity between voters: MinCq favors maximally uncorrelated voters and appears to be a natural way for late fusion to combine the predictions of classifiers separately trained from various modalities.

3.2 MinCq for Ranking

In many applications, such as information retrieval, it is well known that ranking documents is a key point to help users browsing results. The traditional measures to evaluate the ranking ability of algorithms are related to precision and recall. Since a low-error vote is not necessarily a good ranker, we propose in this section an adaptation of MinCq to allow optimization of the Mean Averaged Precision (MAP) measure.

Concretely, given a training sample of size $2m$ we split it randomly into two subsets S' and $S=\{(\mathbf{x}_j, y_j)\}_{j=1}^{m}$ of the same size. Let n be the number of modalities. For each modality i, we train a classifier h_i from S'. Let $\mathcal{H} = \{h_1, \ldots, h_n\}$ be the set of the n associated prediction functions and their opposite. Now at this step, the fusion is achieved by MinCq: We learn from S the weighted majority vote over $\mathcal{H}$ with the lowest risk.

We now recall the definition of the MAP measured on S for a given real-valued function h. Let $S^+ = \{(\mathbf{x}_j, y_j) : (\mathbf{x}_j, y_j) \in S \wedge y_j = 1\} = \{(\mathbf{x}_{j^+}, 1)\}_{j^+=1}^{m^+}$ be the set of the m^+ positive examples from S and $S^- = \{(\mathbf{x}_j, y_j) : (\mathbf{x}_j, y_j) \in S \wedge y_j = -1\} = \{(\mathbf{x}_{j^-}, -1)\}_{j^-=1}^{m^-}$ the set of the m^- negative examples from S ($m^+ + m^- = m$). For evaluating the MAP, one ranks the examples in descending order of the scores. The MAP of h over S is,

$$MAP_S(h) = \frac{1}{|m^+|}\sum_{j:y_j=1} Prec@j,$$

where $Prec@j$ is the percentage of positive examples in the top j. The intuition is that we prefer positive examples with a score higher than negative ones.

MinCq with Pairwise Preference. To achieve this goal, we propose to make use of *pairwise preferences* [8] on pairs of positive-negative instances. Indeed, pairwise methods are known to be a good compromise between accuracy and more complex performance measure like MAP. Especially, the notion of order-preserving pairwise loss was introduced in [23] in the context of multiclass classification. Following this idea, [22] have proposed a SVM-based method with a hinge-loss relaxation of a MAP-loss. In our specific case of MinCq for late fusion, we design an order-preserving pairwise loss for correctly ranking the positive examples. For each pair $(\mathbf{x}_{j^+}, \mathbf{x}_{j^-}) \in S^+ \times S^-$, we want,

$$H_{\mathcal{Q}}(\mathbf{x}_{j^+}) > H_{\mathcal{Q}}(\mathbf{x}_{j^-}) \Leftrightarrow H_{\mathcal{Q}}(\mathbf{x}_{j^-}) - H_{\mathcal{Q}}(\mathbf{x}_{j^+}) < 0.$$

This can be forced by minimizing (according to the weights) the following hinge-loss relaxation of the previous equation (where $[a]_+ = \max(a, 0)$ is the hinge-loss),

$$\frac{1}{m^+ m^-}\sum_{j^+=1}^{m^+}\sum_{j^-=1}^{m^-}\Big[\sum_{i=1}^{n}\underbrace{\left(2q_i - \tfrac{1}{n}\right)\left(h_i(\mathbf{x}_{j^-}) - h_i(\mathbf{x}_{j^+})\right)}_{H_{\mathcal{Q}}(\mathbf{x}_{j^-}) - H_{\mathcal{Q}}(\mathbf{x}_{j^+})}\Big]_+. \tag{6}$$

To deal with the hinge-loss of (6), we consider $m^+ \times m^-$ additional *slack variables* $\boldsymbol{\xi}_{S^+\times S^-} = (\xi_{j^+j^-})_{1\leq j^+\leq m^+, 1\leq j^-\leq m^-}$ weighted by a parameter $\beta > 0$. We make a little abuse of notation to highlight the difference with ($MinCq$): Since $\boldsymbol{\xi}_{S^+\times S^-}$ appear only in the linear term, we obtain the following quadratic program ($MinCq_{PW}$),

$$
\begin{aligned}
&\operatorname{argmin}_{\mathbf{Q},\boldsymbol{\xi}_{S^+\times S^-}} \quad \mathbf{Q}_S^t \mathbf{M}_S \mathbf{Q} - \mathbf{A}_S^t \mathbf{Q} + \beta\, \mathbf{Id}^t \boldsymbol{\xi}_{S^+\times S^-},\\
&\text{s.t. } \mathbf{m}_S^t \mathbf{Q} = \frac{\mu}{2} + \frac{1}{nm}\sum_{j=1}^{m}\sum_{i=1}^{n} y_j h_i(\mathbf{x}_j),\\
&\forall (j^+, j^-) \in \{1,..,m^+\}\times\{1,..,m^-\},\ \xi_{j^+j^-} \geq 0,\ \xi_{j^+j^-} \geq \frac{1}{m^+m^-}\sum_{i=1}^{n}\left(2q_i' - \tfrac{1}{n}\right)\left(h_i(\mathbf{x}_{j^-}) - h_i(\mathbf{x}_{j^+})\right),\\
&\text{and } \forall i \in \{1,\ldots,n\},\ 0 \leq q_i' \leq \tfrac{1}{n}, \qquad\qquad (MinCq_{PW})
\end{aligned}
$$

where $\mathbf{Id} = (1,\ldots,1)$ of size $(m^+ \times m^-)$. However, one drawback of this method is the incorporation of a quadratic number of additive variables $(m^+ \times m^-)$ which makes the problem harder to solve. To overcome this problem, we relax this approach as follows.

MinCq with Average Pairwise Preference. We relax the constraints by considering the average score over the negative examples: we force the positive ones to be higher than the average negative scores. This leads us to the following alternative problem ($MinCq_{PWav}$) with only m^+ additional variables.

$$
\begin{aligned}
&\operatorname{argmin}_{\mathbf{Q},\boldsymbol{\xi}_{S^+}} \quad \mathbf{Q}_S^t \mathbf{M}_S \mathbf{Q} - \mathbf{A}_S^t \mathbf{Q} + \beta\, \mathbf{Id}^t \boldsymbol{\xi}_{S^+},\\
&\text{s.t. } \mathbf{m}_S^t \mathbf{Q} = \frac{\mu}{2} + \frac{1}{nm}\sum_{j=1}^{m}\sum_{i=1}^{n} y_j h_i(\mathbf{x}_j),\\
&\quad \forall j^+ \in \{1,\ldots,m^+\},\ \xi_{j^+} \geq 0,\ \xi_{j^+} \geq \frac{1}{m^+m^-}\sum_{j^-=1}^{m^-}\sum_{i=1}^{n}\left(2q_i' - \tfrac{1}{n}\right)\left(h_i(\mathbf{x}_{j^-}) - h_i(\mathbf{x}_{j^+})\right),\\
&\text{and } \forall i \in \{1,\ldots,n\},\ 0 \leq q_i' \leq \tfrac{1}{n}, \qquad\qquad (MinCq_{PWav})
\end{aligned}
$$

where $\mathbf{Id} = (1,\ldots,1)$ of size m^+.

Note that the two approaches stand in the original framework of MinCq. In fact, we regularize the search of the weights for majority vote leading to an higher MAP. To conclude, our extension of MinCq aims at favoring $\mathcal{Q}$-majority vote implying a good trade-off between classifiers maximally uncorrelated and leading to a relevant ranking.

4 Experiments on PascalVOC'07 Benchmark

Protocol. In this section, we show empirically the usefulness of late fusion MinCq-based methods with stacking. We experiment these approaches on the PascalVOC'07 benchmark [5], where the objective is to perform the classification for 20 concepts. The corpus is constituted of $10,000$ images split into train, validation and test sets. For most of concepts, the ratio between positive and negative examples is less than 10%, which leads to unbalanced dataset and requires to carefully train each classifier. For simplicity reasons, we generate a training set constituted of all the training positive examples and negative examples independently drawn such that the positive ratio is $1/3$. We keep the original test set. Indeed, our objective is not to provide the best results on this benchmark but rather to evaluate if the MinCq-based methods could be helpful for the late fusion step in multimedia indexing. We consider 9 different visual features, that are

Table 1. MAP obtained on the PascalVOC'07 test sample

concept	$MinCq_{PWav}$	$MinCq_{PW}$	$MinCq$	Σ	Σ_{MAP}	$best$	h_{best}
aeroplane	0.487	0.486	**0.526**	0.460	0.241	0.287	0.382
bicycle	0.195	0.204	**0.221**	0.077	0.086	0.051	0.121
bird	0.169	0.137	**0.204**	0.110	0.093	0.113	0.123
boat	0.159	0.154	0.159	0.206	0.132	0.079	**0.258**
bottle	0.112	**0.126**	0.118	0.023	0.025	0.017	0.066
bus	0.167	0.166	**0.168**	0.161	0.098	0.089	0.116
car	**0.521**	0.465	0.495	0.227	0.161	0.208	0.214
cat	**0.230**	0.219	0.220	0.074	0.075	0.065	0.116
chair	**0.257**	0.193	0.230	0.242	0.129	0.178	0.227
cow	0.102	0.101	**0.118**	0.078	0.068	0.06	0.101
diningtable	0.118	0.131	0.149	**0.153**	0.091	0.093	0.124
dog	**0.260**	0.259	0.253	0.004	0.064	0.028	0.126
horse	0.301	0.259	0.303	**0.364**	0.195	0.141	0.221
motorbike	0.141	0.113	0.162	**0.193**	0.115	0.076	0.130
person	**0.624**	0.617	0.604	0.001	0.053	0.037	0.246
pottedplant	**0.067**	0.061	0.061	0.057	0.04	0.046	0.073
sheep	0.067	0.096	0.0695	**0.128**	0.062	0.064	0.083
sofa	0.204	**0.208**	0.201	0.137	0.087	0.108	0.147
train	0.331	0.332	**0.335**	0.314	0.164	0.197	0.248
tvmonitor	**0.281**	**0.281**	0.256	0.015	0.102	0.069	0.171
Average	0.240	0.231	**0.243**	0.151	0.104	0.100	0.165

SIFT, Local Binary Pattern (LBP), Percepts, 2 Histograms Of Gradient (HOG), 2 Local Color Histograms (LCH) and 2 Color Moments (CM):

- LCH are $3\times3\times3$ histogram on a grid of 8×6 or 4×3 blocs. Color Moments are represented by the two first moments on a grid of 8×6 or 4×3 blocs.
- HOG is computed on a grid of 4×3 blocs. Each bin is defined as the sum of the magnitude gradients from 50 orientations. Thus, overall EDH feature has 600 dimensions. HOG feature is known to be invariant to scale and translation.
- LBP is computed on grid of 2×2 blocs, leading to a $1,024$ dimensional vector. The LBP operator labels the pixels of an image by thresholding the 3×3-neighborhood of each pixel with the center value and considering the result as a decimal number. LBP is known to be invariant to any monotonic change in gray level.
- Percept features are similar to SIFT codebook where visual words are related to semantic classes at local level. There are 15 semantic classes such as 'sky', 'skin', 'greenery', 'rock', etc. We also considered SIFT features from a dense grid, then map it on a codebook of 1000 visual words generated with Kmeans.

We train a SVM-classifier for each feature with the LibSVM library [2] and a RBF kernel with parameters tuned by cross-validation. The set $\mathcal{H}$ is then constituted by the 9 score functions associated with the SVM-classifiers.

In a first series of experiments, the set of voters $\mathcal{H}$ is constituted by the 9 SVM-classifiers. We compare our 3 MinCq-based methods to the following 4 baselines:

- The best classifier of $\mathcal{H}$: $h_{best} = \text{argmax}_{h_i\in\mathcal{H}}\, MAP_S(h_i)$.
- The one with the highest confidence: $best(\mathbf{x}) = \text{argmax}_{h_i\in\mathcal{H}}\, |h_i(\mathbf{x})|$.
- The sum of the classifiers (unweighted vote): $\Sigma(\mathbf{x}) = \sum_{h_i\in\mathcal{H}} h_i(\mathbf{x})$.

Table 2. MAP obtained on the PascalVOC'07 test sample with a RBF kernel layer

concept	$MinCq^{rbf}_{PWav}$	$MinCq^{rbf}$	SVM^{rbf}
aeroplane	**0.513**	**0.513**	0.497
bicycle	**0.273**	0.219	0.232
bird	**0.266**	0.264	0.196
boat	**0.267**	0.242	0.240
bottle	**0.103**	0.099	0.042
bus	**0.261**	**0.261**	0.212
car	**0.530**	**0.530**	0.399
cat	**0.253**	0.245	0.160
chair	**0.397**	**0.397**	0.312
cow	0.158	**0.177**	0.117
diningtable	**0.263**	0.227	0.245
dog	**0.261**	0.179	0.152
horse	**0.495**	0.450	0.437
motorbike	**0.295**	0.284	0.207
person	**0.630**	0.614	0.237
pottedplant	0.102	**0.116**	0.065
sheep	**0.184**	0.175	0.144
sofa	**0.246**	0.211	0.162
train	**0.399**	0.385	0.397
tvmonitor	**0.272**	0.257	0.230
Average	**0.301**	0.292	0.234

- The MAP-weighted vote: $\Sigma_{MAP}(\mathbf{x}) = \sum_{h_i \in \mathcal{H}} \frac{MAP_S(h_i)}{\sum_{h_{i'} \in \mathcal{H}} MAP_S(h_{i'})} h_i(\mathbf{x})$.

In a second series, we propose to introduce non-linear information with a RBF kernel layer for increasing the diversity over the set $\mathcal{H}$. We consider a larger $\mathcal{H}$ as follows. Each example is represented by the vector of its scores with the 9 SVM-classifiers and $\mathcal{H}$ is now the set of kernels over the sample S: Each $\mathbf{x} \in S$ is seen as a voter $k(\cdot, \mathbf{x})$. We compare this approach to classical stacking with SVM.

Finally, for tuning the hyperparameters we use a 5-folds cross-validation process, where instead of selecting the parameters leading to the lowest risk, we select the ones leading to the best MAP. The MAP-performances are reported on Tab. 1 for the first series and on Tab. 2 for the second series.

Results. Firstly, the performance of Σ_{MAP} fusion is lower than Σ, which means that the performance of single classifiers is not correlated linearly with its importance on the fusion step. On Tab. 1, for the first experiments, we clearly see that the linear MinCq-based algorithms outperform on average the linear baselines. MinCq-based method produces the highest MAP for 16 out of 20 concepts. Using a Student paired t-test, this result is statistically confirmed with a p-value < 0.001 in comparison with Σ_{MAP}, $best$ and h_{best}. In comparison of Σ, the p-values respectively associated with ($MinCq_{PWav}$), ($MinCq_{PW}$) and ($MinCq_{PW}$) are 0.0139, 0.0232 and 0.0088. We can remark that ($MinCq_{PW}$) implies lower performances than its relaxation ($MinCq_{PWav}$). A Student test leads to a p-value of 0.223, which statistically means that the two approaches produce similar results. Thus, when our objective is to rank the positive examples before the negative examples, the average constraints appear to be a good solution. However, we note that the order-preserving hinge-loss is not really helpful: The classical ($MinCq$) shows the best MAP results (with a p-value of 0.2574). Indeed, the trade-off between diversity and ranking is difficult to apply here since the 9 voters are probably not enough

expressive. On the one hand, the preference constraints appear hard to satisfy, on the other hand, the voters' diversity do not really vary.

The addition of a kernel layer allows us to increase this expressivity. Indeed, Tab. 2 shows that the MinCq-based methods achieve the highest MAP for every concept in comparison with SVM classifier. This confirms that the diversity between voters is well modeled by MinCq algorithm. Especially, $MinCq^{rbf}_{PWav}$ with the averaged pairwise preference is significantly the best: a Student paired test implies a p-value of 0.0003 when we compare $MinCq^{rbf}_{PWav}$ to SVM, and the p-value is 0.0038 when it is compared to $MinCq^{rbf}$. Thus, the the order-preserving loss is a good compromise between improving the MAP and keeping a reasonable computational cost. Note that we do not report the results for $(MinCq_{PW})$ in this context, because the computational cost is much higher and the performance is lower. The full pairwise version implies too many variables which penalize the resolution of $(MinCq_{PW})$. Finally, it appears that at least one MinCq-based approach is the best for each concept, showing that MinCq methods outperform the other compared methods. Moreover, a Student test implies a p-value < 0.001 when we compare $MinCq^{rbf}_{PWav}$ to the approaches without kernel layer. $MinCq^{rbf}_{PWav}$ is significantly then the best approach in our experiments.

We conclude from these experiments that MinCq-based approaches are a good alternative for late classifiers fusion as it takes into account the diversity of the voters. In the context of multimedia documents retrieval, the diversity of the voters comes from either the variability of input features or by the variability of first layer classifiers.

5 Conclusion and Perspectives

In this paper, we proposed to make use of a well-founded learning quadratic program called MinCq for multimedia late fusion tasks. MinCq was originally developed for binary classification, aiming at minimizing the error rate of the weighted majority vote by considering the diversity of the voters [12]. We designed an adaptation of MinCq able to deal with ranking problems by considering pairwise preferences while taking into account the diversity of the models. In the context of multimedia indexing, this extension of MinCq appears naturally appropriate for combining the predictions of classifiers trained from various modalities in a late classifier fusion setting. Our experiments have confirmed that MinCq is a very competitive alternative for classifier fusion in the context of an image indexing task. Beyond these results, this work gives rise to many interesting remarks, among which the following ones. Taking advantage of a margin constraint for late classifier fusion may allow us to prove a new C-bound specific to ranking problems, and thus to derive other algorithms for classifier fusion by maximizing the diversity between the classifiers. This could be done by investigating some theoretical results using the Cantelli-Chebychev's inequality [3] as in [12]. Additionally, it might be interesting to study the impact of using other diversity metrics [10] on performances for image and video retrieval. Such an analysis would be useful for assessing a trade-off between the quality of the ranking results and the diversity of the inputs for information retrieval. Finally, another perspective, directly founded on the general PAC-Bayes theory [15], could be to take into account a prior belief on the classifiers of $\mathcal{H}$. Indeed, general PAC-Bayesian theory allows one to obtain theoretical guarantees on majority votes with respect to the distance between the considered vote and the prior belief

measured by the Kullback-Leibler divergence. The idea is then to take into account prior information for learning good majority votes for ranking problems.

Acknowledgments. This work was in parts funded by the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no 308036. The authors would like to thanks Thomas Peel for useful comments.

References

1. Atrey, P.K., Hossain, M.A., El-Saddik, A., Kankanhalli, M.S.: Multimodal fusion for multimedia analysis: a survey. Multimedia Syst. 16(6), 345–379 (2010)
2. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001)
3. Devroye, L., Gyorfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer (1996)
4. Dietterich, T.G.: Ensemble methods in machine learning. In: Multiple Classifier Systems, pp. 1–15 (2000)
5. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL visual object classes challenge 2007 (VOC 2007) results (2007)
6. Fakeri-Tabrizi, A., Amini, M.-R., Gallinari, P.: Multiview semi-supervised ranking for automatic image annotation. In: ACM Multimedia, pp. 513–516 (2013)
7. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Proc. of ICML, pp. 148–156 (1996)
8. Fürnkranz, J., Hüllermeier, E.: Preference Learning. Springer (2010)
9. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. TPAMI 20, 226–239 (1998)
10. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms (2004)
11. Lacasse, A., Laviolette, F., Marchand, M., Germain, P., Usunier, N.: PAC-Bayes bounds for the risk of the majority vote and the variance of the gibbs classifier. In: NIPS (2006)
12. Laviolette, F., Marchand, M., Roy, J.-F.: From PAC-Bayes bounds to quadratic programs for majority votes. In: ICML (2011)
13. Leonard, D., Lillis, D., Zhang, L., Toolan, F., Collier, R.W., Dunnion, J.: Applying machine learning diversity metrics to data fusion in information retrieval. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 695–698. Springer, Heidelberg (2011)
14. Ma, A.J., Yuen, P.C., Lai, J.-H.: Linear dependency modeling for classifier fusion and feature combination. TPAMI 35(5), 1135–1148 (2013)
15. McAllester, D.A.: PAC-bayesian model averaging. In: COLT, pp. 164–170 (1999)
16. Re, M., Valentini, G.: Ensemble methods: a review. In: Advances in machine learning and data mining for astronomy, pp. 563–582 (2012)
17. Snoek, C., Worring, M., Smeulders, A.W.M.: Early versus late fusion in semantic video analysis. In: ACM Multimedia, pp. 399–402 (2005)
18. Sun, S.: A survey of multi-view machine learning. Neural Computing and Applications 23(7-8), 2031–2038 (2013)
19. Wickramaratna, J., Holden, S., Buxton, B.F.: Performance degradation in boosting. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 11–21. Springer, Heidelberg (2001)
20. Wolpert, D.H.: Stacked generalization. Neural Networks 5(2), 241–259 (1992)
21. Wu, Y., Chang, E.Y., Chang, K.C.-C., Smith, J.R.: Optimal multimodal fusion for multimedia data analysis. In: ACM Multimedia, pp. 572–579 (2004)
22. Yue, Y., Finley, T., Radlinski, F., Joachims, T.: A support vector method for optimizing average precision. In: SIGIR, pp. 271–278 (2007)
23. Zhang, T.: Statistical analysis of some multi-category large margin classification methods. JMLR 5, 1225–1251 (2004)

Entropic Graph Embedding via Multivariate Degree Distributions

Cheng Ye, Richard C. Wilson, and Edwin R. Hancock

Department of Computer Science,
University of York,
York, YO10 5GH, UK
{cy666,richard.wilson,edwin.hancock}@york.ac.uk

Abstract. Although there are many existing alternative methods for using structural characterizations of undirected graphs for embedding, clustering and classification problems, there is relatively little literature aimed at dealing with such problems for directed graphs. In this paper we present a novel method for characterizing graph structure that can be used to embed directed graphs into a feature space. The method commences from a characterization based on the distribution of the von Neumann entropy of a directed graph with the in and out-degree configurations associated with directed edges. We start from a recently developed expression for the von Neumann entropy of a directed graph, which depends on vertex in-degree and out-degree statistics, and thus obtain a multivariate edge-based distribution of entropy. We show how this distribution can be encoded as a multi-dimensional histogram, which captures the structure of a directed graph and reflects its complexity. By performing principal components analysis on a sample of histograms, we embed populations of directed graphs into a low dimensional space. Finally, we undertake experiments on both artificial and real-world data to demonstrate that our directed graph embedding method is effective in distinguishing different types of directed graphs.

Keywords: directed graph embedding, von Neumann entropy, entropy distribution.

1 Introduction

There has been a considerable body of work aimed at extracting features from undirected graphs which reflect their structure and complexity. With such features to hand, especially multi-dimensional ones, then problems such as graph embedding, clustering and classification can be addressed using standard machine learning and pattern recognition techniques. Unfortunately, there is very little work on the corresponding problems for directed graphs. This is unfortunate since many of the most common networks structures, e.g. the World Wide Web, exist in the form of directed graphs.

Motivated by the need to fill this gap in literature, in this paper we aim to develop a method based on information theory to extract multi-dimensional

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 163–172, 2014.

features that can be used to characterize the structure of directed graphs, and hence render them amenable to embedding, clustering and classification. The starting point is a recent result where we have shown how to compute the von Neumann entropy for a directed graph using the configurations of in and out-degrees on directed edges.

1.1 Related Literature

Quantifying the intrinsic complexity of undirected graphs is a problem of fundamental practical importance in network analysis and pattern recognition. A good recent review of the state of the art can be found in the collection of papers edited by Dehmer and Mowshowitz [1]. Moreover, the entropy measures have also been shown to be an effective tool for representing the complexity in graph structure. Han et al. [2] have shown how to approximate the calculation of von Neumann entropy in terms of simple degree statistics rather than needing to compute the normalized Laplacian spectrum.

However, while the problem of computing the entropy of undirected graphs is well studied, the literature on directed graphs is rather limited. One recent exception is the work of Berwanger et al. [3], who have proposed a new parameter for the complexity of infinite directed graphs by measuring the extent to which cycles in graphs are intertwined.

We now turn our attention to embedding methods, which has become a topic of considerable interest for characterizing patterns and graphs in recent years. Broadly speaking, with different choices of graph structure characteristics, there are many existing alternative measures for embedding undirected graphs into feature vectors. An interesting method is provided by Ren et al. [4], who have used the polynomial coefficients determined by the Ihara zeta function to construct a feature vector, which shows good performance in graph clustering. Moreover, feature vectors can also be derived by embedding graphs into a feature space based on dissimilarity embedding [5]. Unfortunately, there are relatively few corresponding methods developed for embedding directed graphs into a feature space. One exception is the work proposed by Chen et al. [6], who have suggested a directed graph embedding method by preserving the local information of vertices in a directed graph. Similarly, directed graph embedding can also be obtained by retaining the information of directionality of the graph [7].

1.2 Contribution

The motivation of this paper is to explore whether we can extract multi-dimensional structural features from directed graphs, and hence apply standard techniques from pattern recognition and machine learning to embed, cluster and classify data in the form of samples of directed graphs. One natural way of capturing the structure of a graph at the complexity level, is to use an entropic characterization. Hence we commence by computing the von Neumann entropy associated with each edge in a directed graph. An analysis, extending our own

previously published work [8] shows that the entropy depends on the configuration of in and out-degrees of the two vertices defining a directed edge. This leads us to a four-dimensional characterization of directed graph structure, which depends on the distribution of entropy with the in and out-degrees of pairs of vertices connected by a directed edge. We represent this distribution by a four-dimensional histogram, which can be encoded as a long-vector for the purposes of analysis. To curb the size of the histogram, we show how to requantize the bin-contents using quantiles of the four cumulative degree distributions.

2 Graph Embedding via Von Neumann Entropy Distribution

In this section, we start from an approximation of the von Neumann entropy of a directed graph [8], and quantify the entropy associated with each directed edge. We show that this entropy is determined by the in and out-degrees of the start and end vertices connected by a directed edge. Based on this observation we explore the multivariate distribution of directed edge entropy with the different combinations of vertex in and out-degrees that define edges in a graph. In practice this distribution can be computed by constructing a multi-dimensional histogram whose bins are indexed by the in and out-degrees of the connected vertices and whose contents accumulates the entropy contributions over the directed edges in the graph. The contents of the histogram can be represented by a multi-dimensional array whose contents can be encoded as a long-vector, which serves as a feature vector for the graph.

One of the problems that potentially limits this approach is that the vertex degree is unbounded. Hence, the size of histogram can become large. Moreover, it can become populated by a large number of empty bins. This renders the analysis of the feature vector unstable. In order to keep the vector length constant and reduce the number of empty bins, we requantize the degree bins of the histogram using quantiles of the cumulative distribution function (CDF). Specifically, we determine the m-quantiles, which divides the ordered vertex degree data into m essentially equal-sized parts. This allows us to relabel each vertex with two quantile labels $(1, 2, \ldots, m)$, one for in-degree and the second for out-degree. As a result, the length of our proposed feature vector is not affected by the variance of the degree distribution.

2.1 Edge-Based Local Entropic Measure

Suppose $G(V, E)$ is a directed graph with vertex set V and edge set $E \subseteq V \times V$, then the adjacency matrix A is defined as follows

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

The in-degree and out-degree of vertex u are

$$d_u^{in} = \sum_{v \in V} A_{vu} \quad d_u^{out} = \sum_{v \in V} A_{uv} \tag{2}$$

Recently, commencing from Passerini and Severini's work [9], Ye et al. [8] have extended the calculation of von Neumann entropy from undirected graphs to directed graphs, using Chung's definition of the normalized Laplacian of a directed graph [10], with the result that

$$H_{VN}^{D} = \frac{1}{2|V|}\left\{ \sum_{(u,v)\in E} \frac{d_u^{in}}{d_v^{in} d_u^{out^2}} + \sum_{(u,v)\in E_b} \frac{1}{d_u^{out} d_v^{out}} \right\} \tag{3}$$

where $E_b = \{(u,v)|(u,v) \in E \text{ and } (v,u) \in E\}$ is the set of bidirectional edges.

In particular, if the cardinality of E_b is very small ($|E_b| \ll |E|$), i.e. a graph is strongly directed (SD), this expression can be simplified one step further by ignoring the summation over E_b in Eq.(3),

$$H_{VN}^{SD} = \frac{1}{2|V|} \sum_{(u,v)\in E} \left\{ \frac{d_u^{in}}{d_v^{in} d_u^{out^2}} \right\} \tag{4}$$

These approximations sum the entropy contribution from each directed edge, and these are based on the in and out-degree statistics of the directed edge. In other words we can compute a normalized local entropy measure for each directed edge. Specifically, for an edge $(u,v) \in E$, we compute

$$I_{uv} = \frac{d_u^{in}}{2|E||V| d_v^{in} d_u^{out^2}} \tag{5}$$

as the entropy contribution. If this edge is bidirectional, i.e. $(u,v) \in E_b$, then we add an addition entropy contribution

$$I'_{uv} = \frac{1}{2|E_b||V| d_u^{out} d_v^{out}} \tag{6}$$

This local measure represents the entropy associated with each directed edge since for arbitrary directed graphs, we have $\sum_{(u,v)\in E} I_{uv} + \sum_{(u,v)\in E_b} I'_{uv} = H_{VN}^{D}$ and for strongly directed graphs, we also have $\sum_{(u,v)\in E} I_{uv} = H_{VN}^{SD}$. Moreover, this measure avoids the bias caused by graph size, which means that it is the edge entropy contribution determined by the in and out-degree statistics, and neither the vertex number or edge number of the graph that distinguishes a directed edge.

2.2 Feature Vector Extracted from Entropy Distribution

Our directed graph characterization is based on the statistical information converged by the distribution of directed edge entropy with the in and out-degrees of the start and end vertices. We represent this distribution of entropy using a four-dimensional histogram over the in and out-degrees of the two vertices.

As noted above, one potential problem is that the bin-contents can become sparse in a high dimensional histogram. To overcome this problem we turn to the cumulative distribution function. Suppose a directed graph $G(V,E)$ has $|V|$

vertices which have been sorted according to in-degree (or out-degree) in the sequence $d_1^{in} \leq d_2^{in} \leq \cdots \leq d_{|V|}^{in}$. Let $P(X = d_i^{in})$ be the in-degree probability distribution of the graph. The corresponding cumulative distribution function for the in-degree is given by

$$F_X(d_i^{in}) = P(X \leq d_i^{in})$$

where $i = 1, 2, \ldots, |V|$. This function describes the probability that a given in-degree X takes on a value less than or equal to d_i^{in}.

Quantiles are intervals of equal size over the cumulative distribution function. They divide the ordered data $d_1^{in}, d_2^{in}, \cdots, d_{|V|}^{in}$ into a number of equal-sized data subsets. Since vertex degree is always a non-negative integer, the quantiles can thus be viewed as new quantization of the degree based on its statistical distribution. We define our degree quantiles over the cumulative distribution of degree for the entire sample of graphs under study, and produce requantized versions of the individual entropy histograms for each individual graph. Suppose the number of quantiles in each dimension of the degree distribution is fixed to be m. Then, for example, the m-quantiles of the in-degree distribution can be obtained as follows

$$Q_j = \underset{d_i^{in}}{\operatorname{argmin}} \left\{ F_{Q_j}(d_i^{in}) - \frac{j}{m} \right\} \tag{7}$$

where $i = 1, 2, \ldots, |V|$ and $j = 1, 2, \ldots, m$. It is clear that these degree quantiles satisfy $Q_1 \leq Q_2 \leq \cdots \leq Q_m$ and in fact, $Q_m = d_{|V|}^{in}$.

With the sample degree quantiles to hand, we assign each vertex degree quantile labels. We first examine the original in-degree d_u^{in} of a vertex u, if d_u^{in} satisfies the condition that $Q_{k-1} < d_u^{in} \leq Q_k$, then its in-degree quantile is $q_u^{in} = k$. The corresponding out-degree quantile labels can also be obtained in the same manner. Since all the vertices in the graph have in-degree and out-degree quantile labels ranging from 1 to m, we can then simply construct the directed edge entropy histogram whose size in each dimension is fixed to m. The histogram is stored as a four-dimensional array.

To do this, we first construct a $m \times m \times m \times m$ array M whose elements represent the histogram bin-contents, and whose indices represent the degree quantile labels of the vertices. For instance, the element $M(1, 2, 3, 4)$ accumulates the entropy contribution for all the directed edges starting from vertices with out-degree quantile label 1 and in-degree quantile label 2, pointing to vertices with out-degree quantile label 3 and in-degree quantile label 4. We then compute the bin-contents by summing the directed edge entropy contributions over the sample graph. The histogram bins contain all directed edges having the same quantile label combinations. We store the accumulated sum in the corresponding element of array M. The elementwise accumulation is formally given as

$$M_{ijkl} = \sum_{\substack{q_u^{out}=i, q_u^{in}=j \\ q_v^{out}=k, q_v^{in}=l \\ (u,v)\in E}} \left\{ \frac{d_u^{in}}{2|E||V| d_v^{in} d_u^{out^2}} \right\} \tag{8}$$

If the graph contains bidirectional edges, we additionally accumulate the following quantity

$$M'_{ijkl} = \sum_{\substack{q_u^{out}=i,q_u^{in}=j \\ q_v^{out}=k,q_v^{in}=l \\ (u,v)\in E_b}} \left\{ \frac{1}{2|E_b||V|d_u^{out}d_v^{out}} \right\} \tag{9}$$

where $i, j, k, l = 1, 2, \ldots, m$. To extract a feature vector from M, we can simply list all the elements in the array, with the result that

$$v = (M_{1111}, M_{1112}, \cdots, M_{111m}, M_{1121}, M_{1122}, \cdots, M_{mmmm})^T \tag{10}$$

Clearly, this feature vector has length m^4.

It is worth pausing to consider the case of strongly directed graphs. For such graphs, from Eq.(4) it is clear that directed edge entropy does not depend on d_v^{out}. As a result the dimensionality of the corresponding histogram can be reduced from four to three by ignoring the third index k in M_{ijkl} (Eq.(8)). This leads to a new feature vector with length m^3. In the following discussion, to distinguish between these two kinds of feature vectors, we name the former full-form (FF) while the latter strongly-directed (SD).

When accumulated in this way we effectively count directed edges with the same configurations of degree quantile labels, and weight them according to their entropy. If the different quantile labels were independent, we would expect a uniform histogram. However, structure in the individual sample graphs due to preferred combinations of vertex in-degree and out-degree will give rise to a non-uniform distribution. To some extent, the quantization of the distribution of entropy with degree according to quantile labels, may dilute this structure due to merging adjacent degree bins. However, the directed edge entropy contribution is based on the original vertex in and out-degree statistics, and the m-quantiles play a role in diminishing the bias caused by different populations of directed graphs. Therefore our proposed representation can still be effective in capturing statistical information concerning the local structural properties in the graph. By embedding graphs into a space spanned by feature vectors, it provides a theoretically principled and efficient tool for graph characterization tasks, which captures the graph characteristics at both the statistical and structural levels.

3 Experiments and Evaluations

In this section, we aim to evaluate the experimental performance of our suggested directed graph characterization. Specifically, we first explore the graph clustering performance of our method on a set of random graphs generated from three classical random graph models. Then we apply our method to some real-world data, including the COIL object recognition data and protein database, and report the graph classification results.

3.1 Datasets

We commence by giving a brief overview of the datasets used for experiments in this paper. We use three different datasets, the first one is synthetically generated artificial networks, while the other two are extracted from real-world systems.

Artificial Data: Contains a large number of directed graphs which are randomly generated according to a) the classical Erdős-Rényi model, b) the "small-world" model, and c) the "scale-free" model. The different graphs in the database are created using a variety of model parameters, e.g. the graph size and the vertex connection probability in the Erdős-Rényi model, the edge rewiring probability in the "small-world" model and the number of added connections at each time step in the "scale-free" model.

COIL Data: Contains object recognition data collected by Nene et al. [11], in which each 3D object consists of 72 images collected from equally spaced changes in viewing direction over 360 degrees. For each image, we establish a 3-nearest neighbour graph on the extracted feature points, i.e. each feature point have three directed edges going to its nearest neighbour points, thus the graph is directed and the out-degree of all vertices is 3. There are two subsets in this database, one contains the directed graphs extracted from 4 different 3D objects while the other contains graphs from 8 objects.

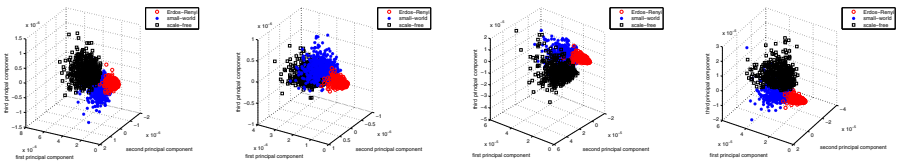
Protein Data: Is extracted from the protein database previously used by Riesen and Bunke [12]. It consists of over 200 graphs, representing proteins labelled with their corresponding enzyme class labels from the BRENDA enzyme database. The database consists of six classes (labelled EC $1, \ldots,$ EC 6), which represent proteins out of the six enzyme commission top level hierarchy (EC classes). The proteins are converted into graphs by first replacing the secondary structure elements of a protein with vertices, and then constructing a 3-nearest neighbour graph for the secondary structure elements. The graphs are thus directed.

3.2 Graph Clustering Performance

To investigate the clustering performance of our proposed directed graph characterization, we perform principle component analysis (PCA) on both FF feature vectors and SD feature vectors extracted from the randomly generated graphs in the Artificial Data. These feature vectors are long-vectors formed by concatenating the elements of the four and three-dimensional histograms respectively. Here we select different parameter settings to generate 500 normal directed graphs and 500 additional strongly directed graphs for each of the three random graph models, with graph size ranged between 100 and 150. Moreover, in all the experiments in this section, we choose the number of quantiles $m = 3$, giving all the FF feature vectors with a constant length $m^4 = 81$, while for SD feature vectors, the length is $m^3 = 27$.

Figures 1(a), (c) and (d) each show that by embedding different random graphs into a feature space spanned by the first three principal components constructed from the feature vectors, the three classes of random graphs display

some clear separation between each other. However in Fig.1(b), which is the plot of SD feature vectors extracted from normal directed graphs, the "small-world" graphs and "scale-free" graphs show some overlap. This suggests the FF feature vectors are efficient in distinguishing any normal directed graphs while the SD feature vectors are effective only for strongly directed graphs, which is an expected result. Therefore in the following experiments we use the FF feature vectors in our analysis.



(a) (b) (c) (d)

Fig. 1. Clustering performance for random graphs using PCA: a) FF feature vectors extracted from normal directed graphs; b) SD feature vectors extracted from normal directed graphs; c) FF feature vectors extracted from SD graphs; d) SD feature vectors extracted from SD graphs. Red: Erdős-Rényi graphs; blue: "small-world" graphs; black: "scale-free" graphs.

3.3 Graph Classification Results

To take this analysis one step further, we evaluate the classification performance of our method on the graphs in COIL DATA and Protein Data, using standard vector-based clustering and classification algorithms. In the following evaluation, we perform the 10-fold cross-validation using two classifiers, namely support vector machine (SVM) classifier associated with the sequential minimal optimization (SMO) [13] and the Pearson VII universal kernel (Puk), and k-nearest neighbour (kNN) classifier. All the SMO-SVM and kNN parameters are optimized for each method on a Weka platform, and all experiments are performed on an Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz processor, with 8 GB memory.

In Fig.2 we report the average classification rates of 10 runs for both SVM and kNN classifiers as a function of quantile number m on three different datasets, including the 4-object data and 8-object data in COIL Data and Protein Data. Figure 3 gives the relationship between the average runtime and the quantile number of the experiments on these datasets.

From Fig.3 we find that the experimental runtime for all three classification problems grows as the quantile number increases, which is as expected since greater quantile number leads to greater size of the feature vector, resulting in the greater computational complexity. Moreover, it is clear that our directed graph characterization is computationally tractable as the runtime does not increase rapidly even when the size of the feature vector becomes particularly large.

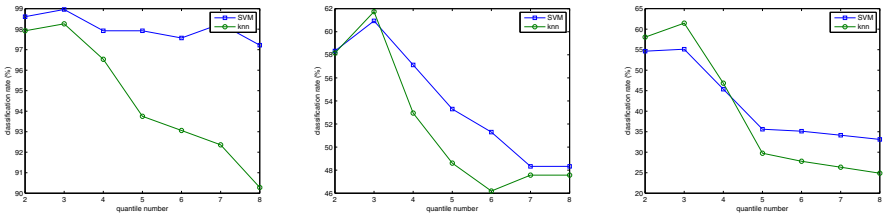


Fig. 2. Average classification rates for both SVM and kNN classifiers with different quantile numbers on datasets: a) 4-object data; b) 8-object data and c) Protein Data. Square: SVM classifier; circle: kNN classifier.

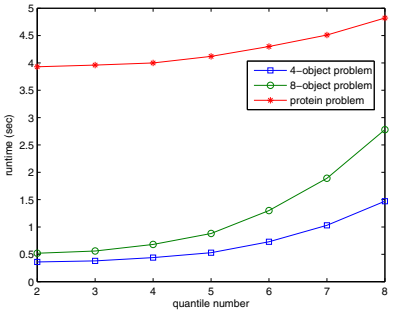


Fig. 3. Average experimental runtime with various quantile numbers for different classification problems. Square: 4-object problem; circle: 8-object problem; star: protein problem.

Turning attention to the classification results reported in Fig.2(a), (b) and (c), we find the performance is particularly good on 4-object data, with a classification accuracy over 98%, and on 8-object data and 6-class protein database, the accuracy is still acceptable (50% to 60%). Moreover, as the increase of the quantile number, the classification rates for both classifiers on all three datasets witness a slight growth, reaching a peak when the quantile number reaches 3, then they drop significantly. This is because in the graphs of these datasets, all vertices have the same out-degree 3, therefore when $m = 3$ the corresponding feature vectors can precisely preserve the information of the vertex in and out-degree statistics, which guarantees that $m = 3$ gives the best classification performance and any greater quantile number will lead to a decrease of classification accuracy. Furthermore, with this choice of quantile number, the experimental runtime is relatively low, which suggests that our method can achieve a sufficient accuracy without causing expensive computation. Overall, based on these observations we claim that that our directed graph characterization can be both accurate and computationally efficient in clustering and classifying directed graphs when the appropriate parameters are selected.

4 Conclusion

In this paper we have suggested a novel and effective method for directed graph characterization based on the multivariate distribution of local von Neumann entropy contribution with vertex in-degree and out-degree. This provides a complexity level characterization of graph structure based on the statistical information residing edge degree distribution. By representing graphs using feature vectors that encode the entropy distribution, both clustering and classification can be addressed using standard pattern recognition and machine learning techniques. We have undertaken experiments to demonstrate that our method is both accurate and computationally efficient in dealing with both artificial and real-world data. In the future, we intend to explore kernels defined over the inner products of our entropy distribution feature vectors.

References

1. Dehmer, M., Mowshowitz, A., Emmert-Streib, F.: Advances in Network Complexity. Wiley-Blackwell (2013)
2. Han, L., Escolano, F., Hancock, E., Wilson, R.: Graph characterizations from von neumann entropy. Pattern Recognition Letters 33, 1958–1967 (2012)
3. Berwanger, D., Gradel, E., Kaiser, L., Rabinovich, R.: Entanglement and the complexity of directed graphs. Theoretical Computer Science 463, 2–25 (2012)
4. Ren, P., Wilson, R., Hancock, E.: Graph characterization via ihara coefficients. IEEE Transactions on Neural Networks 22, 233–245 (2011)
5. Bunke, H., Riesen, K.: Improving vector space embedding of graphs through feature selection algorithms. Pattern Recognition 44, 1928–1940 (2010)
6. Chen, M., Yang, Q., Tang, X.: Directed graph embedding. In: IJCAI, pp. 2707–2712 (2007)
7. Perrault-Joncas, D., Meliă, M.: Directed graph embedding: an algorithm based on continuous limits of laplacian-type operators. Advances in Neural Information Processing Systems 24, 990–998 (2011)
8. Ye, C., Wilson, R.C., Comin, C.H., da F. Costa, L., Hancock, E.R.: Entropy and heterogeneity measures for directed graphs. In: Hancock, E., Pelillo, M. (eds.) SIMBAD 2013. LNCS, vol. 7953, pp. 219–234. Springer, Heidelberg (2013)
9. Passerini, F., Severini, S.: The von neumann entropy of networks. Inthernational Journal of Agent Technologies and Systems, 58–67 (2008)
10. Chung, F.: Laplacians and the cheeger inequailty for directed graphs. Annals of Combinatorics 9, 1–19 (2005)
11. Nene, A., Nayar, S., Murase, H.: Columbia object image library (coil-20). Technical Report (CUCS-005-96) (February 1996)
12. 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.) SSPR&SPR 2008. LNCS, vol. 5342, pp. 287–297. Springer, Heidelberg (2008)
13. Platt, J.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. MIT Press (1999)

On Parallel Lines in Noisy Forms

George Nagy

Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY, USA
nagy@ecse.rpi.edu

Abstract. Quantification of the rectilinear configuration of typeset rules (lines) opens the way to form classification and content extraction. Line detection on scanned forms is often accomplished with the Hough transform. Here it is followed by simultaneous extraction of the dominant perpendicular sets of extracted lines, which ensures rotation invariance. Translation and scale invariance are attained by using minimal horizontal and vertical sets of distance ratios ("rule gap ratios") instead of rule-edge locations. The ratios are logarithmically mapped to an alphabet so that the resulting symbol strings can be classified by. edit distance. Some probability distributions associated with these steps are derived. Analytical considerations and small-scale experiments on scanned forms suggest that this approach has potential merit for processing degraded forms.

Keywords: forms, tables, rules, distance ratio, rotation invariance, scale invariance, random-phase noise, edit distance.

1 Introduction

Many documents exhibit an isothetic configuration consisting of orthogonal sets of parallel components. Line segments are explicit in ruled tables and forms, and implicit in parallel rows of text and justified margins and gutters. Rectilinear structures are also common in artifacts like cultivated fields, cities, buildings and machines: in fact, their presence is one of the prime clues for distinguishing man-made from natural. Although parallel lines play a role in other image processing and computer vision tasks as well, here we address only scanned or photographed form images. Fig. 1 shows examples of forms that offer a rich line structure but may have been scanned at too low resolution or are too noisy for OCR-based classification.

The "near-horizontal" lines shown in Fig. 1 were extracted by the Hough transform in rho-theta format. The line configurations include both rectilinear *rules* (the printing and publishing term for typeset lines), and s*purious lines* induced by accidental alignments of diverse page content. The images display various rule configurations, with the members of each class sharing essentially the same rule configuration but exhibiting different spurious lines. The task at hand is classifying new images into predefined classes. Since the forms are captured by a scanner or a camera, their position, scale and skew angle within the image are unknown.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 173–182, 2014.

The ratios of the distances between pairs of rules (*rule gap ratios*) are geometrically invariant features. (Invariant features are more commonly used in scene image analysis than in document recognition.) The ordered sets of horizontal and vertical ratio values are converted to a pair of symbol strings that characterize the ruling configuration of the underlying form. The forms are then classified according to the (1,1,1) edit distance between new images and existing class representative. So we

1. Distinguish isothetic rules from spurious lines formed by accidental alignments;
2. Compute the minimum set of algebraically independent rule gap ratios;
3. Map the ordered horizontal and vertical gap ratios into symbol strings;
4. Classify the unknown images based on the edit distance between symbol strings.

In the following sections we review prior work, examine each of the above steps, and give an example of their application to a set of degraded and mutilated form images.

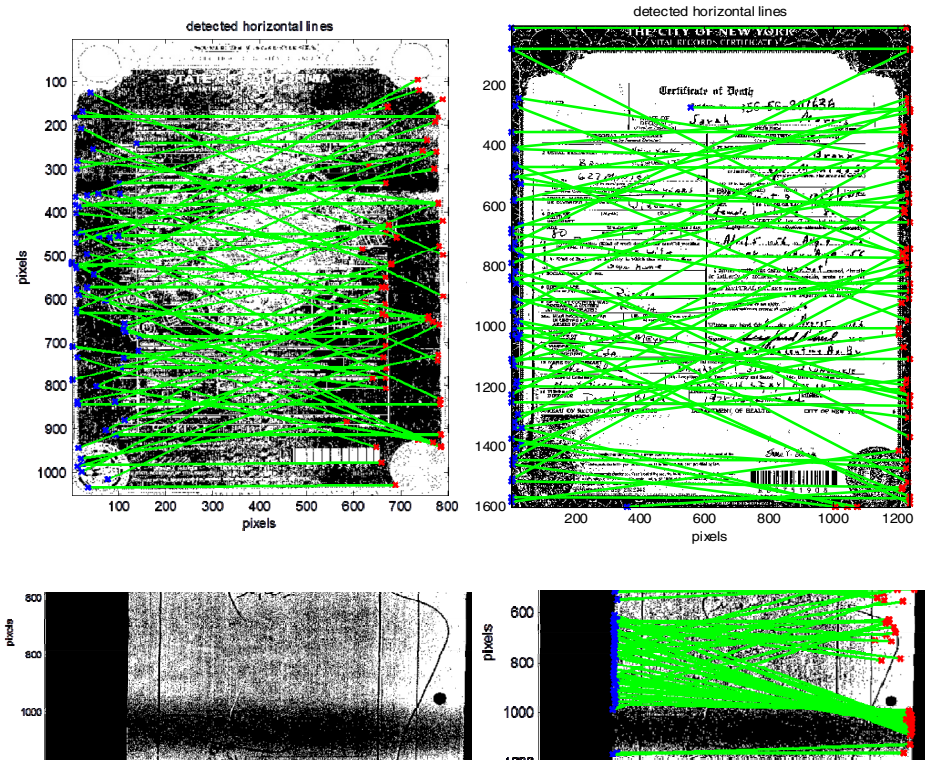


Fig. 1. Examples of forms with explicit isothetic rule structure. The two forms on top are from web archives. The partial form image on the bottom is from our classification experiment. Here only lines (shown in green) within ±30° of horizontal are extracted.

2 Prior Work

Line segment recognition has been steadily improved during the last three decades as part of table interpretation, form processing, and engineering drawing analysis. Historical form analysis became popular even as most contemporary forms migrated to the web. The Hough transform for line location has remained one of the leading methods for line and arc extraction since its rediscovery by Duda and Hart in the early seventies [1]. It does not require edge linking and is therefore often preceded only by edge extraction with the venerable Prewitt filter [2]. Other 3×3 pixel edge filters (Sobel, Roberts) yield similar results. We have found neither research addressing the extraction and quantification of rectilinear rule structures independently of other document content, nor prior application of orthogonal line filtering to Hough lines.

Our interest in spatial sampling noise was triggered by peaks in the autocorrelation function corresponding to opposite stroke edges in scanned character images [3]. The variation (noise!) due to repeated scanning was exploited by Zhou and Lopresti to decrease OCR error [4]. Random-phase sampling noise was systematically investigated in remote sensing [5,6] and in scanned documents [7], but pixel jitter is usually modeled as if it were independent random displacement of sensor elements [8]. The relationship between spatial and amplitude quantization in scanning was explored thoroughly by Barney Smith [9].

Levenshtein introduced the edit distance for error-correcting codes in 1965 [10]. The optimal Wagner-Fischer algorithm was published a decade later [11]. Many variations of the original algorithms have appeared since then [12,13,14]. The role of the edit distance in communications and text processing was augmented by its application to genome sequencing. Developments relevant to document image analysis include normalization methods [15] and kernel techniques for embedding the edit distance into a vector space [16]. The public-domain EDIT DISTANCE WEIGHTED program that we use was posted in 2010 by B. Schauerte [17].

The current study was initiated during a phase of the MADCAP project [18] concerned with categorization of a small subset of the collection of Kurdish documents recovered during the Anfal uprising [19,20]. The Hough transform parameters and preliminary results on classification of some degraded forms were presented at the 2014 SPIE Conference on Document Recognition and Retrieval [21].

3 Orthogonal Line Extraction

The accidental alignments of handwriting, stamps, binder holes, checkmarks and other non-rule pixels may give rise to far more spurious lines than the number actual rules on the page (Fig. 2). Since in contrast to the randomly distributed spurious lines all the nominally horizontal (or vertical) rules have the same angle, an obvious way to distinguish them is to histogram all the line angles. Then the lines in the most populated bin will be the rules. This stratagem fails only if too many spurious lines fall into some other bin. Below we calculate the dominant term of the probability of such an event as a proxy for the actual probability.

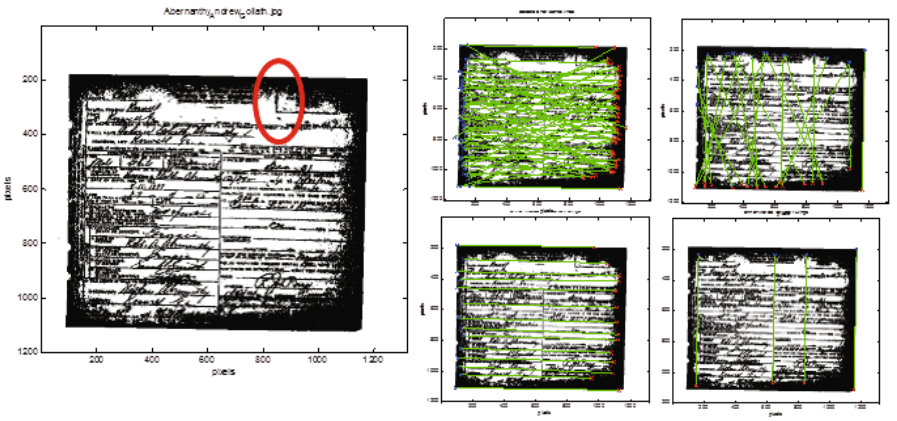


Fig. 2. A low-resolution, noisy and skewed death certificate. Near-horizontal and near vertical lines extracted by the Hough Transform and rules retained by orthogonal filtering.

Extreme skew is unlikely, therefore only lines within ±20° of the nominal and x- and y-axes need be extracted. Let there be R rules and S spurious lines on a page. Their angles are sorted into a histogram with N uniformly spaced bins ($N > R+S$). The rules are parallel and therefore fall into the same bin, but the skew detection will be incorrect if R or more of the spurious lines fall into some other bin. Under a (questionable!) i.i.d. assumption, the most probable such case is that R of the S spurious lines fall into a single bin and that each of the others occupies one bin. This can happen in as many ways as there are of picking single-occupancy bins. Therefore a lower bound on the probability that at least R of the S spurious lines fall into the same bin is:

$$P(\mathit{false}\ \max) > PeR = N\binom{S}{R,1,1,\ldots,1}\left(\frac{1}{N-1}\right)^{S}\binom{N-2}{S-R,N-2-(S-R)}$$

Table 1. Dominant term of the probability of false maxima in the angle histogram

N	R	S	PeR %		N	R	S	PeR %
20	3	3	0.27701		40	3	3	0.065746
20	3	6	3.95461		40	3	6	1.122005
20	3	9	6.61081		40	3	9	3.119688
20	3	12	3.33205		40	3	12	4.099149
20	6	6	0.00004		40	6	6	1.11E-06
20	6	9	0.00242		40	6	9	7.94E-05
20	6	12	0.01060		40	6	12	0.000579

The shaded cells of Table 1 show that while the probability of a false maximum for 3 rules and 6 spurious lines is at least an appreciable 3.9%, doubling the number of lines reduces the dominant term to 0.01%. This can be achieved by adding 90° to the theta coordinate of every line within 20° of the vertical axis and histogramming all the line angles together. In the image of Fig. 2, every visible vertical rule is found,

including the edge of the box at the top right of the form marked with a red oval, with no false positives. Simultaneous identification of orthogonal lines pays off.

4 Rule Gap Ratios

No further use of the theta coordinates is made. The computation of the rule gap ratios requires only sorting the Hough rho coordinates of each set of extracted and ortho-filtered parallel lines and subtracting them pairwise to find the successive horizontal and vertical edge-to-edge rule gaps. Given N parallel rules, there are $O(N^2)$ pairs of rules and $O(N^4)$ possible ratios. It is clear, however, that there cannot be more than $N-2$ algebraically independent ratios from which the value of all the others can be calculated. We choose as *basis ratios* the ratios of consecutive gaps, defined for horizontal or vertical lines located at $x_1, x_2, \ldots, x_i, \ldots, x_N$ (w.r.t. an arbitrary origin) as:

$$R_i = \left(x_{i+1} - x_i\right) / \left(x_{i+2} - x_{i+1}\right)$$

There are $N-2$ such ratios, and any other ratio of line segments can be recovered from them. The proof is conceptually simple but notationally tedious, so we give an example instead. Let the three distances between four lines be a, b, and c (Fig. 3).

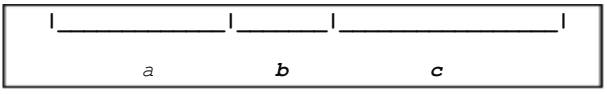


Fig. 3. Ratios of rule gaps

The two basis ratios are $R_1 = a/b$, and $R_2 = b/c$. An arbitrary ratio such as $(b+c)/(a+b)$ can be expressed in terms of the basis ratios as:

$$\frac{b+c}{a+b} = \frac{1}{(a/b)+1} + \frac{1}{(b/c)(a/b+1)} = \frac{1}{R_1+1} + \frac{1}{R_2(R_1+1)}$$

The general formula that proves the sufficiency of the basis ratios is:

$$\frac{x_s - x_t}{x_u - x_v} = \frac{\left(R_1 \times R_2 \times \ldots \times R_{t-1}\right)\left(1 + R_t\left[1 + R_{t+1}\left[\ldots\left[1 + R_{s-1}\right]\right]\right]\right)}{\left(R_1 \times R_2 \times \ldots \times R_{v-1}\right)\left(1 + R_v\left[1 + R_{v+1}\left[\ldots\left[1 + R_{u-1}\right]\right]\right]\right)}$$

The rule configuration of a page is preserved by the two sets of translation, scale and rotation invariant basis ratios. Lines are considered to be of infinite extent. If end-point information is required, it is kept separately. The accuracy of the rule gap ratios is affected by edge location variability and by random-phase sampling noise.

4.1 Edge Location Variability

Some applications must cope with forms reprinted at different times and by different printers. Even if the variability of the line and line-edge locations as a fraction of page

size is small, it may have a significant effect on the gap ratios. Each gap ratio is a function of the position of three (parallel) rules. What is the probability density function (pdf) of the ratio as a function of the variability of the edges?

The only line-segment ratio we found discussed in the literature is that resulting from of splitting a unit-length line segment by a uniformly distributed point L, which results in ratio $W = L/(1-L)$ [22]. The probability density of W,

$$f(w) = 1/(1+w)^2,$$

is skewed because its range is zero to infinity but its mean must be 0.5.

We extended the calculation of the pdf of $W = L/(1-L)$ to two *independent* (non-adjacent) gaps of lengths L_1 and L_2 distributed uniformly: $L_1 \in x_0 \pm a$ and $L_2 \in y_0 \pm a$. The resulting piecewise rational polynomial functions provide further insight:

$$f(w) = 0 \qquad \text{if } w \le \frac{x_0 - a}{y_0 + a} \text{ or } \frac{x_0 + a}{y_0 - a} < w;$$

$$f(w) = \frac{1}{8a^2}\left[(y_0 + a)^2 - \left(\frac{x_0 - a}{w}\right)^2\right] \qquad \text{if } \frac{x_0 - a}{y_0 + a} < w \le \frac{x_0 - a}{y_0 - a};$$

$$f(w) = \frac{y_0}{2a} \qquad \text{if } \frac{x_0 - a}{y_0 - a} < w \le \frac{x_0 + a}{y_0 + a};$$

$$f(w) = \frac{1}{8a^2}\left[\left(\frac{(x_0 + a)}{w}\right)^2 - (y_0 - a)^2\right] \qquad \text{if } \frac{x_0 + a}{y_0 + a} < w \le \frac{x_0 + a}{y_0 - a}$$

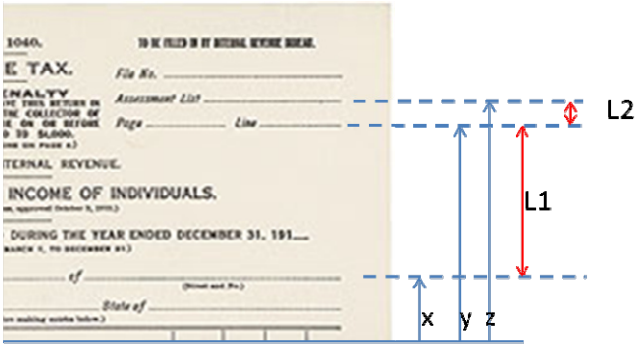


Fig. 4. The rule edges x, y, and z are uniformly distributed over a range of 2a. What is the pdf of the gap ratio $L_1/L2_?$?

What we must consider, however, is the more difficult three-variable case of a basis ratio formed by three *adjacent* edges located at *x, y,* and *z,* where x is uniformly and independently distributed over $x_0 \pm a$, *y* over $y_0 \pm a$, and *z over* $z_0 \pm a$ (Fig. 4). The gaps are $L_1 = y\text{-}x$ and $L_2, = z\text{-}y$. The basis ratio is $W = L_1/L_2$, as in Fig. 3,

The gap lengths L_1 and L_2 are the difference of uniformly and independently distributed variables and therefore have a simple triangular distribution centered on the mean difference. But analytical formulation of the joint pdf of L_1 and L_2 is complicated by the statistical dependence induced by the shared edge y. After deriving the lengthy formula we must still resort to simulation to compute the pdf of the ratio W.

The effect on the ratio of edge variability is illustrated in Fig. 5 for $x_0 = 1$, $y_0 = 4$, $z_0 = 10$, and three values of a. Large values of a correspond to high rule edge variability. W ranges from $(y_0\text{-}x_0\text{-}2a)/(z_0\text{-}y_0\text{+}2a)$ to $(y_0\text{-}x_0\text{+}2a)/(z_0\text{-}y_0\text{-}2a)$. As a approaches zero, the distribution converges to a delta function located at the nominal value of the ratio.

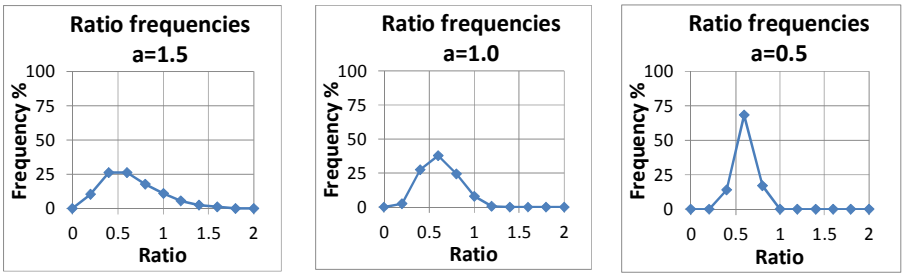


Fig. 5. Frequency distribution of gap ratio between variable edge locations

4.2 Random-Phase Sampling Noise

The precise quantification of gap ratios, like that of all image features, is also hampered by the random-phase noise induced by the arbitrary placement of any document with respect to the scanner's or camera's sensor array. This noise can be reduced, but not eliminated, by increasing the spatial sampling rate.

The distances between rule edges are quantized to integer values by scanning. As a one-dimensional analogy, consider rule gaps of length L_1 and L_2 sampled at δ-length intervals (Fig. 6). After sampling, L_1 will be of length $\lfloor L_1/\delta \rfloor$ or $\lfloor L_1/\delta \rfloor - 1$, and L_2 will be $\lfloor L_2/\delta \rfloor$ or $\lfloor L_2/\delta \rfloor - 1$. (Gap length is the number of background pixels minus1.) The ratio can take only one of three values: $(\lfloor L_1/\delta \rfloor - 1)/(\lfloor L_2/\delta \rfloor)$, $(\lfloor L_1/\delta \rfloor - 1)/(\lfloor L_2/\delta \rfloor - 1)$, and $(\lfloor L_1/\delta \rfloor)/(\lfloor L_2/\delta \rfloor - 1)$. In the worst case, when, $L_i = \lfloor L_i \rfloor + \delta/4$,: the three possible values occur with probabilities of 0.25, 0.50, 0.25. If random-phase sampling noise changes the mapping of any ratio to a symbol (cf. §5), then identical rule configurations will result in different symbol strings and therefore in non-zero edit distance between them.

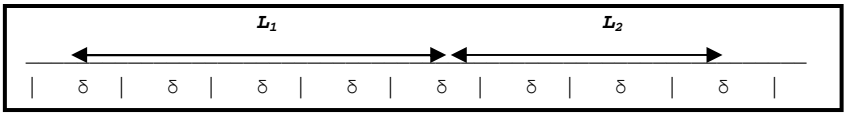


Fig. 6. Random-phase noise. Here $L_1 = 4.2\delta$. After spatial sampling L_1 will be either 3 or 4 pixels long, depending on its position relative to the sampling grid.

5 Ratio Quantization and Edit Distance

The smallest gaps in a document typically correspond to the space required to print or write a word or a number. Even densely printed documents have no more than 60 lines of text; most forms have fewer than 20. The smallest gaps are likely to be those from double rule dashes. The largest gap can be no larger than page height. Gap ratios typically range from 0.1 to 10, and the smallest significant difference is about 30%.

Uniform quantization of the ratios – for edit distance computation – would map the prevalent near-unity ratios into very few symbols. Logarithmic mapping of gap ratios to string symbols flattens the resulting symbol probability distribution. Therefore gap ratio R is mapped into bin k, where k ranges from 1 to N:

$$k = F(R;K,N) = \min\left(\max\left(\left\lceil \frac{(\log_{10} R + K)(N-2)}{2K} \right\rceil + 1, \quad 1 \right), \quad N \right)$$

The parameters N and K govern the logarithmic bin size. The domain of the mapping includes two semi-open intervals for very small and very large ratios (for $|\log R| > K$). Setting N =24 and K=1.3 yields 22 finite bins increasing by 30% from R=0.05 to R = 20. The resulting symbol alphabet is {'1', '2', …, '24'}.

The metric used for classification is the Levenshtein edit distance. Schauerte's open-source program accepts arbitrary weights for the cost of the insertions, deletions and substitutions necessary to convert one string into another, but lacking enough training data to estimate the optimal weights we set them all equal. With more data, substitutions could be also weighted according to the size difference of the gap ratios.

The edit distance computation could take into account missing or spurious rules. When a symbol does not match, the algorithm can check whether combining adjacent gaps would reduce the edit distance. (A rule missed in one document is equivalent to a spurious rule in the other and can be treated analogously.) This check can be extended, at exponentially growing cost, to several consecutive gaps.

6 Plausible Applications

Deteriorated and poorly-scanned forms abound in historical census, military and municipal records. Some of the recent interest in such documents is due to genealogical research (including its medical implications). Even contemporary forms may be degraded by repeated photocopying, reduced resolution for web display, or batch scanning with a page-feed scanner without adequate skew and binarization control.

Modern form identification is generally based on a barcode or some Form Identification Number (FIN) prominently printed at the top or near one of the corners. In their absence, OCR'd forms can be identified using preprinted text specific to each type of form. Both the FIN and the preprinted labels usually exhibit enough redundancy to tolerate OCR errors. The ruling-based classification discussed here is appropriate only for forms that cannot be OCR'd and have an isothetic rule structure without too many other aligned edges. In principle the method could be applied hierarchically, possibly via the quad tree [23], to forms with highly localized rules.

The rule detection, logarithmic gap ratio quantization and string matching were applied as part of the MADCAT project to a set of 158 extremely noisy scanned forms of 15 types (Fig. 7). These filled-out forms contain personnel information collected by Iraqi government agencies and regrettably only redacted or partial images can be presented. The forms were classified by a Nearest Neighbor classifier with the edit distance function. The resulting error rate was 11% (17 errors). Ten errors are due to groups 3 and 12. One error is unavoidable because Group 13, with only one member, has no reference pattern for Nearest Neighbor. There are 6 confusions between groups 2 and 3 that that differ only by a single ruling. The Matlab program runs in 1 second per form on a 2 GHz laptop, with 83% of the time taken by the Hough transform.

		Assigned																
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	ERROR	TOTAL
	1	3															0	3
	2		23	2		1											3	24
	3		4	0		2											6	6
	4				37												0	37
	5					10											0	10
	6						6										0	8
True	**7**							4									0	4
	8								5								0	5
	9									5							0	5
	10		1			1					11						2	13
	11				1							3					1	4
	12					2	1	1					8				4	12
	13									1				0			1	1
	14														6		0	6
	15															20	0	20
		0	5	2	1	6	1	1	0	1	0	0	0	0	0	0	17	158

Fig. 7. Results from leave-one-out edit-distance based classification of 158 MADCAT forms

7 Envoy

In the expectation of future large-scale endeavors on degraded but rule-rich corpora, we examined some benefits and drawbacks of three related ideas:

- Simultaneous orthogonal filtering of Hough lines to eliminate of spurious lines.
- Extracting gap ratios of parallel rules for geometric invariance.
- Classifying the ratios by edit distance, bridging statistical and structural methods.

Acknowledgment. The author thanks Prof. Daniel Lopresti (Lehigh University) for access to the MADCAT data and for suggestions on edit distance computation. He is also grateful for the close reading of the manuscript and recommendations of Dr. Prateek Sarkar (Google, Inc.) and of one of the referees.

References

1. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. Wiley (1973)
2. Prewitt, J.M.S.: Object Enhancement and Extraction. In: Lipkin, B.S., Rosenfeld, A. (eds.) Picture Processing and Psychopictorics. Academic Press (1970)

3. Nagy, G.: On the Spatial Autocorrelation Function of Noise in Sampled Typewritten Characters. In: 1968 IEEE Region III Convention Record, New Orleans, United States, pp. 7.6.1–7.6.5 (1968)
4. Zhou, J., Lopresti, D.: Repeated Sampling to Improve Classifier Accuracy. In: Proc. IAPR Workshop Machine Vision Applications, Kawasaki, Japan, pp. 346–351 (1994)
5. Havelock, D.I.: Geometric Precision in Noise-Free Digital Images. IEEE Trans. Pattern Analysis and Machine Intelligence 11(10), 1,065–1,075 (1989)
6. Havelock, D.I.: The Topology of Locales and Its Effect on Position Uncertainty. IEEE Trans. Pattern Analysis and Machine Intelligence 13(4), 380–386 (1991)
7. Sarkar, P., Lopresti, D., Zhou, J., Nagy, G.: Spatial Sampling of Printed Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(3), 344–351 (1998)
8. Baird, H.S.: The State of the Art in Document Image Degradation Modeling. In: Chaudhuri, B.B. (ed.) Digital Document Processing, pp. 261–279. Springer (2007)
9. Barney Smith, E.: Characterization of image degradation caused by scanning. Pattern Recognition Letters 19(13), 1191–1197 (1998)
10. Levenshtein, V.I.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Doklady Akademii Nauk SSSR 163(4), 845–848 (1965)
11. Wagner, R.A., Fischer, M.J.: The String-to-String Correction Problem. Journal of the ACM 21(1), 168–173 (1974)
12. Hall, P.A.V., Dowling, G.R.: Approximate String Matching. ACM Computing Surveys 2(4), 381–402 (1980)
13. Sankoff, D., Kruskal, J.B.: Time warps, string edits, and macromolecules: The theory and practice of sequence comparison. Addison Wesley (1983)
14. Navarro, G.: A guided tour to approximate string matching. ACM Computing Surveys 33(1), 31–88 (2001)
15. Marzal, A., Vidal, E.: Computation of Normalized Edit Distance and Applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(9) (September 1993)
16. Neuhaus, M., Bunke, H.: Edit distance-based kernel functions for structural pattern classification. Pattern Recognition 39, 1852–1863 (2006)
17. Schauerte, B., Fink, G.A.: Focusing Computational Visual Attention in Multi-Modal Human-Robot Interaction. In: Proc. ICMI (2010)
18. Multilingual Automatic Document Classification and Translation Evaluation (MADCAT), `http://www.nist.gov/itl/iad/mig/madcat.cfm` (accessed January 8, 2014)
19. Montgomery, B.P.: The Iraqi Secret Police Files: A Documentary Record of the Anfal Genocide. Archivaria 52, 81–82 (2001)
20. Montgomery, B.P.: Returning Evidence to the Scene of the Crime: Why the Anfal Files Should be Repatriated to Iraqi Kurdistan. Archivaria 69, 143–171 (2010)
21. Nagy, G., Lopresti, D.: Form similarity via Levenshtein distance between ortho-filtered logarithmic ruling-gap ratios. In: SPIE/IST Document Recognition and Retrieval (February 2014)
22. Pickover, C.A.: The Problem of the Bones. In: The Mathematics of Oz: Mental Gymnastics from Beyond the Edge, ch. 8. Cambridge University Press, New York (2002)
23. Samet, H.: The Design and Analysis of Spatial Data Structures. Addison-Wesley, Reading (1990)

Metric Learning in Dissimilarity Space for Improved Nearest Neighbor Performance

Robert P.W. Duin[1], Manuele Bicego[2], Mauricio Orozco-Alzate[3], Sang-Woon Kim[4], and Marco Loog[1]

[1] PRLab, Delft University of Technology, The Netherlands
{r.p.w.duin,m.loog}@tudelft.nl
[2] Department of Computer Science,
University of Verona, 37134, Verona, Italy
manuele.bicego@univr.it
[3] Departamento de Informática y Computación,
Universidad Nacional de Colombia, Sede Manizales, Colombia
morozcoa@unal.edu.co
[4] Dept. of Computer Science and Engineering,
Myongji University, Yongin,
449-728 South Korea
kimsw@mju.ac.kr

Abstract. Showing the nearest neighbor is a useful explanation for the result of an automatic classification. Given, expert defined, distance measures may be improved on the basis of a training set. We study several proposals to optimize such measures for nearest neighbor classification, explicitly including non-Euclidean measures. Some of them may directly improve the distance measure, others may construct a dissimilarity space for which the Euclidean distances show significantly better performances. Results are application dependent and raise the question what characteristics of the original distance measures influence the possibilities of metric learning.

1 Introduction

The Nearest Neighbor (NN) rule is a classical and very natural classifier. It does not need density estimation or function optimization as it entirely relies on the user defined distance measure. An important advantage is that it gives an intuitive motivation of the assigned class label by showing the nearest neighbor(s) to the user. A second advantage is that the distance measure fully determines the classification performance as there is no learning involved. All is based on the collection of training examples.

The second advantage is also a disadvantage as it shows that there is room for improvement by using a training set. In case the original objects are represented in a vector space, e.g. by features, the performance may be improved by selecting or rescaling features. Such methods can also be considered as procedures for metric learning. In general, metric learning aims to find a better distance measure between objects on the basis of a training set.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 183–192, 2014.

Studies on metric learning either focus on adaptations of the vector space, preserving the original Euclidean distance, or optimize the metric, preserving the given vector representation, or combine a set of given distance measures. Examples are the Large Margin NN Classifier [11] and the Direct Minimization of the NN Error [3].

We will primarily deal with given, possible non-Euclidean, dissimilarities. New dissimilarity measures defined on the given ones will be proposed and evaluated. This may also yield a non-Euclidean result. We will use the word dissimilarity to emphasize that we allow ill-defined measures that even may violate the triangle inequality. This is in line with many applications based on images, shapes or sequences. It will not harm the use of the NN rule as long as there is a monotonic relation between measured dissimilarities and object differences.

An important possibility that we include in our considerations is that dissimilarities may be used to define a dissimilarity space [4],[6] and that in this space a distance measure is defined that combines the dissimilarities to the objects in the representation set that constitutes the dissimilarity space.

The vector space defined by the dissimilarity representation differs from the feature representation by the mentioned monotonic relation, as well as by the natural correlations arising from using similar objects for representation. Three proposals using these characteristics will be evaluated for some public domain real-world datasets. For evaluation, the performance of the NN rule will be used.

In Section 2 the three proposals will be presented. They are evaluated with the direct NN performance on the given distances as well as with the NN performance in the dissimilarity space. In Section 3 the datasets and some of their properties are reported. Results are presented in Section 4 and conclusions are summarized in the final section.

2 Methods

Let X be a set of labeled training objects $X = \{x_i, i = 1, ..., n\}$ and let x be an arbitrary object inside or outside X. The objects are initially only represented by their dissimilarities $\mathbf{d}(x) = [d(x, x_i), i = 1, ..., n]$. These dissimilarities are defined by some expert (e.g. as function of raw measurements on x and x_i) in such a way that if $d(x, x_1) < d(x, x_2)$ it is more likely that x belongs to the same class as x_1 than that it belongs to the class of x_2. For that reason the NN rule using $\mathbf{d}(x)$ is an appropriate classifier.

We are searching for a modified dissimilarity measure $d_{mod}(x, x_i)$ being a function of all distances to the training set $\mathbf{d}(x)$ such that the performance of the NN rule improves. Any such procedure can be used directly by classifying new objects on the basis of their modified dissimilarities. Below we discuss one existing and three new procedures that will be evaluated in Section 4.

The training set used for metric learning is a square dissimilarity matrix

$$D = [\mathbf{d}(x_1), \mathbf{d}(x_2), ..., \mathbf{d}(x_n)] \tag{1}$$

It is not always symmetric and some procedures allow even non-zero diagonals. When needed we make it symmetric by averaging and force a zero-diagonal.

Such a matrix can be embedded in a $(n-1)$-dimensional pseudo-Euclidean space (PE-Space) [6] that consists of two Euclidean subspaces. These are built by an eigenvalue decomposition of a Gram matrix derived from (1). The eigenvectors corresponding to the positive eigenvalues constitute the positive space, the other ones constitute the negative space. For Euclidean dissimilarity matrices the dimensionality of the latter is zero as in that case all eigenvalues are positive. In this paper the PE-Space will only be used to characterize the dissimilarities.

2.1 Dissimilarity Space, DS

A straightforward way to derive new dissimilarities to a given set of representative objects (the representation set) by combining the available ones is the dissimilarity space, [6]. This is the vector space constructed by the vector of distances as mentioned in the previous subsection: $\mathbf{d}(x) = [d(x, x_i), i = 1, n]$. Here we will use the training set for representation as well. If we use Euclidean distances in the dissimilarity space the modified dissimilarity can be written as:

$$d_{DS}(x, x_i) = \|\mathbf{d}(x) - \mathbf{d}(x_i)\|$$

It has been found in the past [6] that the NN performance may improve as well as deteriorate by this modification. It is still an open issue to find the conditions when one or the other may happen.

2.2 Locally Adaptive Nearest Neighbor Distances, LANN

The locally adaptive distance measure was originally proposed by Wang et al. [10], claiming that it significantly improves the performance of the kNN rule when used with a metric distance measure. The rationale behind their local adaptation approach is simple and elegant: dividing a conventional distance measure —the authors restricted themselves to the Euclidean and Manhattan metrics for five feature-based data sets— by the smallest distances from the corresponding training examples to training examples of different classes. We study the application of the procedure, referred as LANN, to given and unconstrained dissimilarity measures. More formally, LANN can be described as follows.

Let d be a dissimilarity measure and x and x_i be a test object and a training object, respectively. Let r_i be the radius of the largest topological ball[1] around x_i that excludes —in the corresponding PE-space— all training objects from other classes. This radius is given by

$$r_i = \min_{j:\theta_j \neq \theta_i} d(x_i, x_j)$$

where θ_i is the class label associated to the i-th training object.

The locally adaptive dissimilarity measure $d_{LANN}(x, x_i)$ is then defined as:

$$d_{LANN}(x, x_i) = \frac{d(x, x_i)}{r_i} \tag{2}$$

[1] Notice that depending on the dissimilarity measure, the neighborhoods defined by objects with dissimilarity to x_i less than r_i may not be a proper ball.

LANN can be understood as a columnwise scaling of the test dissimilarity matrix, where the scaling factors correspond to the radii associated to the training objects. Dissimilarities to training objects with large radii are diminished/rewarded since they are considered more trustable (a large neighborhood of the same class); conversely, dissimilarities to objects with small radii are, comparatively, emphasized/penalized (less trustable due to a small neighborhood of the same class). Two potential drawbacks associated to LANN are noise sensitivity and dependency on the sample size: notice that (i) outliers, even though not trustable, are associated to large radii and (ii) small training sample sizes will produce large but empty neighborhoods where unseen objects of different classes might lie in.

2.3 Non-linear Scaling of Dissimilarities

Here we explore the possibility of transforming the input dissimilarities by employing a non linear function: in particular we explore the effect of applying the power transformation to each pairwise dissimilarity:

$$d_{NLScale}(x, x_i) = d(x, x_i)^{\rho} \qquad \rho > 0 \tag{3}$$

Clearly, this operation does not have an impact on the NN rule based on the original dissimilarities[1], since a monotonic transformation does not change the ordering of objects. On the contrary, this operation may change the behavior of the NN rule in the dissimilarity space, as it represents a *non-linear scaling* of it.

In general, scaling feature spaces is often very useful, especially for classifiers based on the Euclidean distance or inner products (like NN or SVM). The typical choice in this context is to perform a *linear scaling*, like the well known z-score standardization (every feature is centered and divided by the standard deviation). Nevertheless, there can be situations where the linearity assumption is too restrictive, and a benefit may be obtained from a non-linear scaling, which acts in different ways in different parts of the feature space. One clear example of non-linear transformation, which has nevertheless scarcely applied in the classification context, is the well known Box-Cox transformation [1], [8], introduced in the 60's, representing a parametric way to non linearly transform a set of points in order to make their distribution approximately Gaussian. More recent approaches, explicitly devoted to the classification case, appeared in [2], where kernels for HMM-based generative embeddings were successfully augmented via a non-linear transformation of the space.

Here we propose to use this non-linear scaling to enhance the performances of the NN rule in the dissimilarity space. Dissimilarities appear to be an optimal context where to apply this non-linear mapping, for different reasons: i) the power mapping does not change the rankings of the objects, so the original information on which the space is built is preserved; ii) all the directions of the dissimilarity space share the same nature (they are all dissimilarities), therefore

[1] Even if useless in the NN case, this operation can be beneficial for other classification techniques, especially if they rely on the Euclideaness of the space: actually for $\rho < 1$ the Euclideaness of the dissimilarity matrix is increased by this non linear mapping.

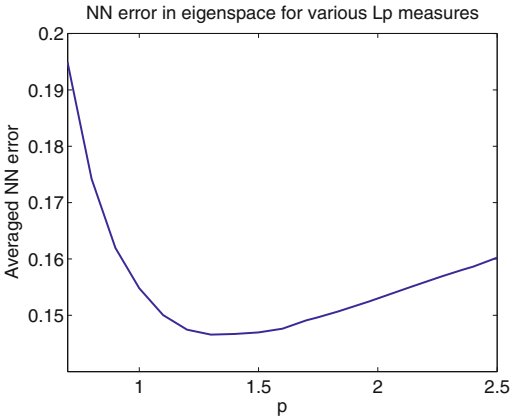


Fig. 1. The NN error in a set of 2-fold cross validation experiments (repeated 10 times) averaged over the 44 Chickenpieces datasets

it may be simpler to find a common good parameter for all the directions; iii) all directions are positives, avoiding strange effects for negative values.

In our implementation, the scaling factor ρ is optimized by a grid search between 0.03 and 30 by a leave-one-out cross-validation. This is still fast up to a few thousand objects (the dissimilarity matrix should fit in fast memory).

2.4 Distance in Eigenspace, ESL1.5

For various applications like histograms and images, other distance measures may be more appropriate than the Euclidean distance based on bins or pixels. In [5] it was suggested to use the L^1 metric. As other metrics than the Euclidean one (L^2) are rotation sensitive, it was suggested in that study to perform an eigenspace rotation first, thereby removing all correlation. (It is admitted that this part of the procedure uses L^2).

Dissimilarity spaces suffer, like pixel based representation, heavily from correlations. We wondered whether other distance measures than L^2 would make sense in the dissimilarity space.

The distance transformation can thereby be written as follows. First the total training set is considered in the dissimilarity space derived from the dissimilarity matrix (1). It coincides with the training set represented in the dissimilarity space. We compute the set of eigenvectors E, so $ED = \Lambda D$ with Λ a diagonal matrix. A vector $\mathbf{d}$(x) in the dissimilarity space is transformed to the eigenspace by

$$\mathbf{e}(x) = E\mathbf{d}(x)$$

The L^p distance in this space of an object x and a training object x_i is:

$$d_{ESLp}(x, x_i) = (\sum_j |e_j(x) - e_j(x_i)|^p)^{1/p} \tag{4}$$

in which $e_j(x)$ is the j-th component of $\mathbf{e}(x)$. Fig. 1 shows a preliminary experiment based on the Chickenpieces dissimilarity dataset, see Section 3. The

NN performances in a 2-fold cross validation experiment averaged of all Chickenpieces datasets are shown for L^p as a function of p. It shows that there is a significant minimum between $p = 1$ and $p = 2$. This appeared to be true in other experiments as well. In a more extensive study p might be optimized for every application. Here we decided to use always $p = 1.5$, avoiding additional cross-validation loops, and named the procedure ESL1.5.

3 Datasets

We use a set of public domain datasets, see Table 1. More information on the datasets themselves can be found on the internet[1]. Most datasets are obtained from real objects (images, text, protein sequences). PolyDisH57 and PolyDisM57 are the only two artificial datasets, obtained by the (modified) Hausdorff distance on randomly generated pentagons and heptagons. The Chickenpieces dataset consists out of 44 dissimilarity matrices. In the table, the average characteristics are shown. The Pendigits dataset is much larger. To make our experiments feasible we used a randomly selected subset of 4000 objects.

Here are short definitions of the properties used in Table 1, see also [4].

- *size*: the total number of objects in the dataset.
- *class*: the number of classes.
- *ID*: an estimate of the the intrinsic dimensionality.
- *LOO*: the leave-one-out NN error.
- *NEF*: the negative eigenfraction, a measure for the Euclideaness.
- *NMF*: the non-metricity fraction of triplets violating the triangle inequality.
- *SignP*: the number of positive eigenvalues in pseudo-Euclidean embedding[2].
- *SignN*: the number of negative eigenvalues in pseudo-Euclidean embedding.
- *Asym*: the averaged deviation of symmetric dissimilarity measure.

4 Evaluation

The procedures described in Section 2 are applied to all datasets mentioned in Section 3. A two-fold cross-validation is repeated 25 times. The errors found by the NN rule are averaged. The mean errors and the standard deviation of the means are listed in Table 2. Results that are significantly better than those obtained for the original dissimilarities are printed in bold. (We judge a difference in means as significant if the intervals defined by the two standard deviations do not overlap). In order to save space, the errors over the Chickenpieces datasets are averaged. Below they will be summarized in some figures.

Table 2 shows the results found by a direct use of the (modified) dissimilarities in the left of every column and the results of the corresponding dissimilarity space in the right. The two procedures LANN and ESL1.5 show many significant

[1] `http://37steps.com/prdisdata`

[2] The two numbers [SignP SignN] are called the *signature* of the embedding.

Table 1. Dataset properties

Dataset	*size*	*class*	*ID*	*LOO*	*NEF*	*NMF*	*SignP*	*SignN*	*Asym*
CatCortex	65	4	18	0.12	0.208	0.002	41	23	0.000
Chickenpieces	446	5	3	0.13	0.273	0.000	242	203	0.051
CoilDelftDiff	288	4	22	0.47	0.128	0.000	163	124	0.000
CoilDelftSame	288	4	13	0.65	0.027	0.000	249	38	0.000
CoilYork	288	4	4	0.23	0.258	0.000	169	118	0.009
DelftGestures	1500	20	6	0.04	0.308	0.000	765	734	0.000
FlowCyto	612	3	2	0.38	0.230	0.004	330	281	0.000
NewsGroups	600	4	83	0.25	0.202	0.000	153	387	0.000
Pendigits	4000	10	4	0.01	0.348	0.002	1944	2055	0.000
PolyDisH57	4000	2	9	0.03	0.415	0.000	2054	1945	0.000
PolyDisM57	4000	2	11	0.02	0.356	0.000	1819	2180	0.000
ProDom	2604	4	17	0.00	0.043	0.000	1502	680	0.000
Protein	213	4	14	0.02	0.001	0.000	205	4	0.000
WoodyPlants50	791	14	5	0.10	0.229	0.000	395	395	0.000
Zongker	2000	10	14	0.44	0.419	0.002	1038	961	0.000

improvements on the original dissimilarities. Note however that the ESL1.5 procedure itself already computes distances (using the L1.5 norm) in dissimilarity space. NLScale transforms the given dissimilarities by a monotonic transformation, the same for all dissimilarities. This does not influence the NN assignments as explained in Section 2.3. Its results on the given dissimilarities are thereby identical to the original ones. The results for its dissimilarity space (right column) show many significant results. In general, it is shown that metric learning may be useful for these datasets.

All Chickenpieces datasets refer to the same set of silhouettes. Bunke and Spillmann [9] just used different parameters in the weighted edit distance measure. They constitute thereby an interesting set of slightly changing dissimilarities. All results for these datasets are summarized in Fig. 2, clearly showing the improvements that are obtained by the various methods.

Since the errors associated to the studied methods correspond to coordinates in the vertical axis, dots below the line indicate that the modified dissimilarity measures are better than their original counterparts (since the lower the error, the better the performance). The further a dot is from the line, the greater the margin of improvement.

Below the individual procedures proposed in Section 2 are discussed separately.

The *dissimilarity space*, Section 2.1 (the right part of each of the columns in Table 2) is a general procedure to combine given dissimilarities into new ones by treating them as vectors. It is not focussed on improvement, but it puts pairwise dissimilarities in the context of all other objects. Sometimes the NN rule on the distances obtained from the dissimilarity space shows an improvement, sometimes it does not. It is an open issue to get a better understanding when this happens.

Table 2. Averaged two-fold cross validation results (error × 1000) for the NN-rule based on 25 repetitions. In every column on the left the NN errors on the dissimilarities, on the right the NN error in the corresponding dissimilarity space. In between brackets the standard deviation of the estimated mean errors. In bold the results that significantly improve the original dissimilarities.

Dataset	*Original*		*LANN*		*NLScale*		*ESL1.5*	
CatCortex	138(10)	**96**(7)	**96**(11)	126(11)	138(10)	**95**(8)	**88**(8)	**106**(8)
Chickenpieces	161(3)	**150**(2)	**123**(3)	156(2)	161(3)	**122**(2)	**144**(2)	216(3)
CoilDelftDiff	513(6)	**464**(7)	**465**(7)	**464**(6)	513(6)	**456**(7)	**450**(7)	531(9)
CoilDelftSame	656(6)	**410**(8)	**540**(8)	**423**(8)	656(6)	**425**(9)	**416**(8)	**517**(10)
CoilYork	319(5)	396(7)	333(5)	411(8)	319(5)	331(7)	392(8)	546(9)
DelftGestures	50(1)	95(1)	66(2)	97(1)	50(1)	54(2)	83(2)	187(2)
FlowCytoDis	403(4)	408(5)	**338**(4)	417(5)	403(4)	404(5)	403(4)	426(6)
NewsGroups	291(5)	293(6)	**269**(4)	332(6)	291(5)	293(5)	295(6)	341(7)
Pendigits	15(1)	23(1)	17(1)	30(1)	15(1)	16(1)	18(1)	61(1)
PolyDisH57	40(1)	**31**(1)	**22**(1)	**30**(1)	40(1)	**20**(1)	**30**(1)	84(1)
PolyDisM57	23(1)	**15**(1)	**12**(0)	**16**(0)	23(1)	**17**(1)	**16**(1)	22(1)
ProDom	9(1)	19(1)	**5**(1)	20(1)	9(1)	8(1)	13(1)	143(3)
Protein	37(5)	**6**(2)	**14**(3)	**4**(1)	37(5)	**8**(2)	**5**(1)	**17**(3)
WoodyPlants50	127(3)	165(3)	**119**(3)	204(3)	127(3)	121(3)	154(3)	263(3)
Zongker	358(25)	**53**(1)	**196**(21)	**130**(7)	358(25)	**40**(2)	**50**(1)	**114**(2)

Metric learning based on the *local adaptive NN procedure*, LANN, Section 2.2 performs remarkably well. It always shows improvements except for the three cases mentioned above. We were afraid that this procedure is very noise sensitive, but apparently the noise introduced by the arbitrary distances to the nearest neighbor does not harm. It is a simple, effective procedure that does not require any optimization.

Let us try to understand the behavior of the *non-linear scaling procedure*, NLScale, Section 2.3, concentrating on the case of $\rho < 1$ (for which we almost always got the best results). When using $\rho < 1$ lower dissimilarities are raised, whereas large ones are reduced. This operation has three effects:

- points tend to have the same distance from all the other points (since the dissimilarities tend to be all equal): this potentially augments the intrinsic dimensionality of the dataset (i.e. the dimensionality of the manifold where the objects lie). The larger this dimensionality, the more Euclidean (flat) the space: techniques relying on Euclidean assumptions (as the NN in the dissimilarity space) can benefit from this. Clearly, such correction can also destroy the information contained in the dissimilarities, as shown in [7].
- the contribution to the dissimilarity space of possible outliers is possibly reduced, since high distances – namely distances from very far points, i.e. outliers – are shrinked.
- the neighborhood of every point is enlarged: small distances, i.e. distances between near points, are emphasized, therefore augmenting the importance in the dissimilarity space of nearest points.

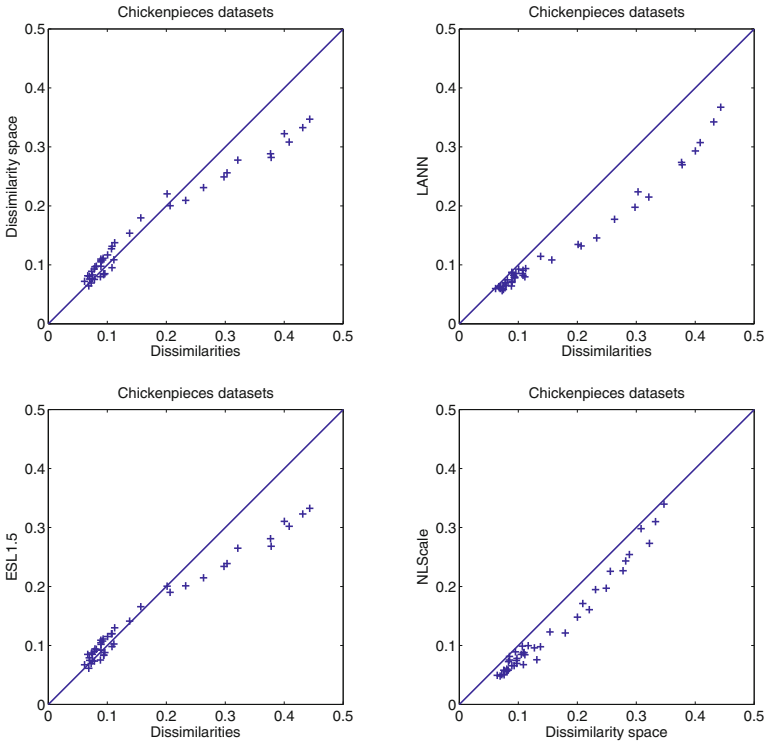


Fig. 2. Results for the 44 Chickenpieces datasets

The eigenspace procedure, ESL1.5, Section 2.4, effectively operates in the original dissimilarity space. For consistency we have printed in bold the significant differences with the original dissimilarity results themselves. Improvements in comparison with the dissimilarity space are less striking, but almost always shown. We conclude from this that the idea of using a non-Euclidean measure in the dissimilarity space (which is almost always used as an Euclidean space [6]) is effective.

5 Conclusion

This study is based on "given dissimilarities": dissimilarity datasets arising from applications, external to our study. In such applications the dissimilarity measure may have been optimized for the given objects. Thereby we might have sometimes made a second attempt to improve this measure by learning from a training set that has already been taken into account. We admit that thereby overtraining may be introduced by squeezing the data further. Nevertheless it is interesting that for 12 of the 15 datasets, one or even several significant improvements could be found. Systematic procedures for metric learning apparently make sense for NN classification.

The datasets have very diverse backgrounds and are based on entirely different dissimilarity measures. One may wonder whether from the dataset characteristics listed in Table 1 can be predicted which procedure for which dataset is promising (meta-learning). At this moment we cannot answer this in a positive way. It is, however, interesting that the datasets that could not be improved (CoilYork, DelftGestures and Pendigits) belong to the most non-Euclidean ones according to the NEF measure. PolyDisH57 and PolyDisM57 have a high NEF value as well, but their distance measures have not been optimized for the application. The ones that could not be improved are the result of studies in which the researchers tried to obtain an optimal result. This might explain both, their strong non-Euclidean behavior as well as the difficulty to improve the metric.

In conclusion, it has been shown that metric learning for a large variation of given, non-Euclidean dissimilarities is well possible and may yield significant improvements.

Acknowledgments. Support from Dirección de Investigación - Sede Manizales (DIMA), Universidad Nacional de Colombia, is acknowledged as well as the Cooperint program from University of Verona.

References

1. Box, G., Cox, D.: An analysis of transformations. Journal of the Royal Statistical Society: Series B (Methodological) 26(2), 211–252 (1964)
2. Carli, A., Bicego, M., Baldo, S., Murino, V.: Nonlinear mappings for generative kernels on latent variable models. In: ICPR, pp. 2134–2137 (2010)
3. Chernoff, K., Loog, M., Nielsen, M.: Metric learning by directly minimizing the k-NN training error. In: ICPR, pp. 1265–1268. IEEE (2012)
4. Duin, R., Pękalska, E., Loog, M.: Non-Euclidean dissimilarities: Causes, embedding and informativeness. In: Pelillo, M. (ed.) Similarity-Based Pattern Analysis and Recognition. Advances in Computer Vision and Pattern Recognition, pp. 13–44. Springer, London (2013)
5. Kim, S.-W., Duin, R.P.W.: Dissimilarity-based classifications in eigenspaces. In: San Martin, C., Kim, S.-W. (eds.) CIARP 2011. LNCS, vol. 7042, pp. 425–432. Springer, Heidelberg (2011)
6. Pękalska, E., Duin, R.: The Dissimilarity Representation for Pattern Recognition. Foundations and Applications. World Scientific, Singapore (2005)
7. Plasencia-Calaña, Y., Cheplygina, V., Duin, R.P.W., García-Reyes, E.B., Orozco-Alzate, M., Tax, D.M.J., Loog, M.: On the informativeness of asymmetric dissimilarities. In: Hancock, E., Pelillo, M. (eds.) SIMBAD 2013. LNCS, vol. 7953, pp. 75–89. Springer, Heidelberg (2013)
8. Sakia, R.: The Box-Cox transformation technique: a review. The Statistician 41, 169–178 (1992)
9. Spillmann, B.: Description of the distance matrices. Tech. rep. (2004), `http://www.iam.unibe.ch/fki/databases/string-edit-distance-matrices/dmdocu.pdf`
10. Wang, J., Neskovic, P., Cooper, L.N.: Improving nearest neighbor rule with a simple adaptive distance measure. Pattern Recognition Letters 28(2), 207–213 (2007)
11. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research 10, 207–244 (2009)

Matching Similarity for Keyword-Based Clustering

Mohammad Rezaei and Pasi Fränti

University of Eastern Finland
{rezaei,franti}@cs.uef.fi

Abstract. Semantic clustering of objects such as documents, web sites and movies based on their keywords is a challenging problem. This requires a similarity measure between two sets of keywords. We present a new measure based on matching the words of two groups assuming that a similarity measure between two individual words is available. The proposed matching similarity measure avoids the problems of traditional measures including minimum, maximum and average similarities. We demonstrate that it provides better clustering than other measures in a location-based service application.

Keywords: clustering, keyword, semantic, hierarchical.

1 Introduction

Clustering has been extensively studied for text mining. Applications include customer segmentation, classification, collaborative filtering, visualization, document organization and indexing. Traditional clustering methods consider numerical and categorical data [1], but recent approaches consider also different text objects such as documents, short texts (e.g. topics and queries), phrases and terms.

Keyword-based clustering aims at grouping objects that are described by a set of *keywords* or *tags*. These include movies, services, web sites and text documents in general. We assume here that the only information available about each data object is its keywords. The keywords can be assigned manually or extracted automatically. Fig. 1 shows an example of services in a location-based application where the objects are defined by a set of keywords. For presenting an overview of available services to a user in a given area, clustering is needed.

Several methods have been proposed for the problem [2, 3, 4, 5] mostly by agglomerative clustering based on single, compete or average links. The problem is closely related to *word clustering* [6, 7, 8] but instead of single words, we have a set of words to be clustered. Both problems are based on measuring similarity between words as the basic component.

To solve clustering, we need to define a similarity (or distance) between the objects. In agglomerative methods such as *single link* and *complete link*, similarity between individual objects is sufficient, but in partitional clustering such as *k-means* and *k-medoids* cluster representative is also required to measure object-to-cluster similarity. Using semantic content, however, defining the representative of a cluster is not trivial. Fortunately, it is still possible to apply partitional clustering even without the representatives. For example, an object can be assigned to such cluster that minimizes

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 193–202, 2014.

Fig. 1. Five examples of location-based services in Mopsi (http://www.uef.fi/mopsi): name of the service, representative image, and the keywords describing the service

(or maximizes) the cost function where only the similarities between objects are needed.

In this paper, we present a novel similarity measure between two sets of words, called *matching similarity*. We apply it to keyword-based clustering of services in a location-based application. Assuming that we have a measure for comparing semantic similarity between two words, the problem is to find a good measure to compare the sets of words. The proposed matching similarity solves the problem as follows. It iteratively pairs two most similar words between the objects and then repeats the process for the rest of the objects until one of the objects runs out of words. The remaining words are then matched just to their most similar counterpart in the other object.

The rest of the paper is organized as follows. In Section 2, we review existing methods for comparing the similarity of two words, and select the most suitable for our need. The new similarity measure is then introduced in Section 2. It is applied to agglomerative clustering in Section 3 with real data and compared against existing similarity measures in this context.

2 Semantic Similarity between Word Groups

In this section, we first review the existing methods for measuring semantic similarity between individual words, because it is the basic requirement for comparing two sets of words. We then study how they can be used for comparing two set of words, present the new measure called *matching similarity*, and demonstrate how it is applied in clustering of services in a location based application.

2.1 Similarity of Words

Measures for semantic similarity of words can be categorized to *corpus-based*, *search engine-based, knowledge-based* and *hybrid*. Corpus-based measures such as *point-wise mutual information* (PMI) [9] and *latent semantic analysis* (LSA) [9] define the similarity based on large corpora and term co-occurrence. Search engine-based measures such as *Google distance* are based on web counts and snippets from results of a search engine [8], [10, 11]. *Flickr distance* first searches two target words separately through the image tags and then uses image contents to calculate the distance between the two words [12].

Knowledge-based measures use lexical databases such as *WordNet* [13] and *CYC* [13], which can be considered as computational format of large amounts of human knowledge. The knowledge extraction process is very time consuming and the database depends on human judgment and it does not scale easily to new words, fields and languages [14, 15].

WordNet is a taxonomy that requires a procedure to derive the similarity score between words. Despite its limitations it has been successively used for clustering [16]. Fig. 2 illustrates a small part of WordNet hierarchy where mammal is the *least common subsumer* of wolf and hunting dog. *Depth* of a word is the number of links between it and the root word in WordNet. As an example, Wu and Palmer measure [17, 18] is defined as follows:

$$S(w_1, w_2) = \frac{2 \times depth(LCS(w_1, w_2))}{depth(w_1) + depth(w_2)} \tag{1}$$

where *LCS* is the least common subsumer of the words w_1 and w_2.

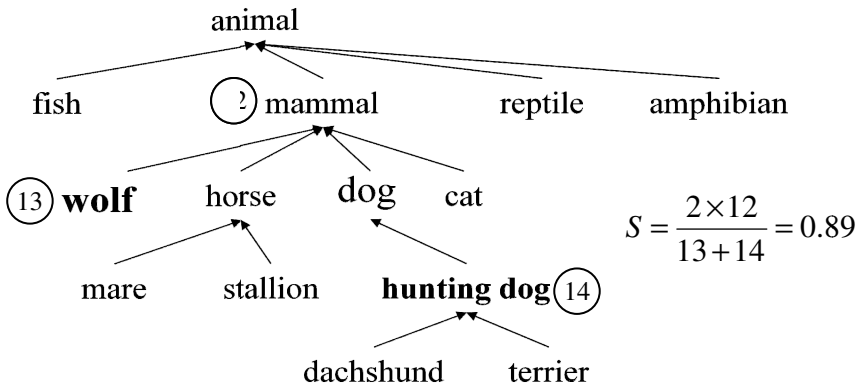


Fig. 2. Part of WordNet taxonomy; the numbers in the circles represent the depths

Jiang-Contrath [13] is a hybrid of corpus-based and knowledge-based as it extracts the information content of two words and their LCS in a corpus. Methods based on Wikipedia or similar websites are also hybrid in the sense that they use organized corpora with links between documents [19]. In the rest of the paper, we use Wu & Palmer measure due to its simplicity and reasonable results in earlier work [16].

2.2 Similarity of Word Groups

Given a measure for comparing two words, our task is to measure similarity between two sets of words. Existing measures calculate either minimum, maximum or average similarities. Minimum and maximum measures find the pair of words (one from each object) that are least (minimum) and most (maximum) similar. Average similarity considers all pairs of words and calculates their average value. Example is shown in Fig. 3, where the values are min=0.21, max=0.84, average=0.57.

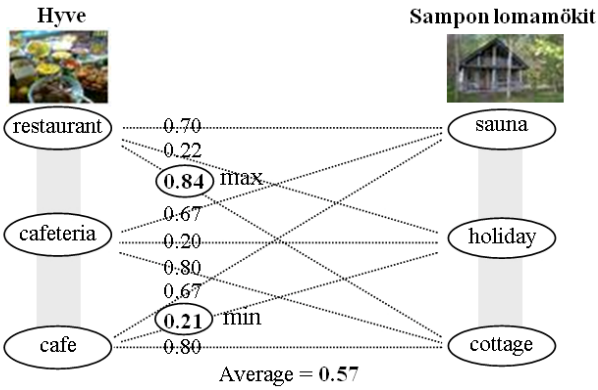


Fig. 3. Minimum and maximum similarities between two location-based services is derived by considering two keywords with minimum and maximum similarities

Now consider two objects with exactly the same keywords (100% similar) as follows:

(a) Café, lunch
(b) Café, lunch

The word similarity between Café and lunch is 0.32. The corresponding minimum, average and maximum similarity measures would result in 0.32, 0.66 and 1.00. It is therefore likely that minimum and average measures would cluster these in different groups and only maximum similarity would cluster them correctly in the same group.

Now consider the following two objects that have a common word:

(a) Book, store
(b) Cloth, store

The maximum similarity measure gives 1.00 and therefore as soon as the agglomerative algorithm processes to these objects, it clusters them in one group. However, if data contains lots of stores, they might have to be clustered differently.

The following example reveals another disadvantage of minimum similarity. These two objects should have a high similarity as their only difference is the drive-in possibility of the first service.

(a) Restaurant, lunch, pizza, kebab, café, drive-in
(b) Restaurant, lunch, pizza, kebab, café

Minimum similarity would result to S(drive-in, pizza)=0.03, and therefore, place the two services in different clusters.

2.3 Matching Similarity

The proposed *matching similarity* measure is based on a greedy pairing algorithm, which first finds two most similar words across the sets, and then iteratively matches next similar words. Finally, the remaining non-paired keywords (of the object with more keywords) are just matched with the most similar words in the other object. Fig. 4 illustrates the matching process between two sample objects.

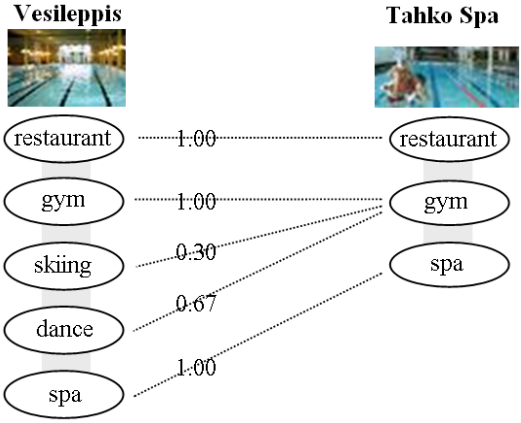


Fig. 4. Matching between the words of two objects

Consider two objects with N_1 and N_2 keywords so that $N_1 > N_2$. We define the normalized similarity between the two objects as follows:

$$S(O_1, O_2) = \frac{\sum_{i=1}^{N_1} SW(w_i^{O_1}, w_{p(i)}^{O_2})}{N_1} \quad (2)$$

where SW measures the similarity between two words, and $p(i)$ provides the index of the matched word for w_i in the other object.

The proposed measure provides more intuitive results than existing measures, and eliminates some of their disadvantages. As a straightforward property it gives the similarity 1.00 for the case of objects with same set of keywords.

3 Experiments

We study the method with Mopsi data (http://www.uef.fi/mopsi), which includes various location-tagged data such as services, photos and routes. Each service includes a set of keywords to describe what it has to offer. Both English and Finnish languages keywords have been casually used. For simplicity, we translated all Finnish words into English by Microsoft Bing translator for these experiments. Some issues raised in translation such as stop words, Finnish word converting to multiple English words, and some strange translations due to using automatic translator. We manually refined the data to remove the problematic words and the stop words.

In total, 378 services were used for evaluating the proposed measure and compare it against the following existing measures: *minimum*, *maximum* and *average similarity*. We apply complete and average link clustering algorithms as they have been widely used in different applications. Each of the clustering algorithms is performed based on three similarity measures. Here we fixed the number of clusters to 5 since our goal of clustering is to present user the main categories of services, with easy navigation to find the desired target without going through a long list. We find the natural number

of clusters using *SC* criteria introduced in [16] by finding minimum *SC* value among clusterings with different number of clusters. We then display four largest clusters and put all the rest in the fifth cluster. The data and the corresponding clustering results can be found here (http://cs.uef.fi/paikka/rezaei/keywords/).

The three similarity measures of five selected services in Table 1 are demonstrated in Table 2. The first three and the last two services should be in two different clusters according to their similarities. However, both minimum and average similarities show small differences when they compare *Parturi-kampaamo Nona* with *Parturi-kampaamo Koivunoro* and *Kahvila Pikantti*, whereas the proposed matching similarity can differentiate them much better. Despite that *Parturi-kampaamo Nona* and *Parturi-kampaamo Koivunoro* have exactly the same keywords, only the matching similarity provides value 1.00 indicating perfect match.

Table 1. Similarities between five services for the measures: minimum, average and matching

Mopsi service:	A1-Parturi-kampaamo Nona	A2-Parturi-kampaamo Platina	A3-Parturi-kampaamo Koivunoro	B1-Kielo	B2-Kahvila Pikantti
Keywords;	barber hair salon	barber hair salon	barber hair salon shop	cafe cafeteria coffe lunch	lunch restaurant

Table 2. Similarity between services described in Table 1

Services	A1	A2	A3	B1	B2
	Minimum similarity				
A1	-	0.42	0.42	0.30	0.30
A2	0.42	-	0.42	0.30	0.30
A3	0.42	0.42	-	0.30	0.30
B1	0.30	0.30	0.30	-	0.32
B2	0.30	0.30	0.30	0.32	-
	Average similarity				
A1	-	0.67	0.67	0.47	0.51
A2	0.67	-	0.67	0.47	0.51
A3	0.67	0.67	-	0.48	0.51
B1	0.47	0.47	0.48	-	0.63
B2	0.51	0.51	0.51	0.63	-
	Matching similarity				
A1	-	1.00	0.99	0.57	0.56
A2	1.00	-	0.99	0.57	0.56
A3	0.99	0.99	-	0.55	0.56
B1	0.57	0.57	0.55	-	0.90
B2	0.56	0.56	0.56	0.90	-

In general, the problems of minimum and average similarities are observable in the clustering results both for complete and average link. Several services with the same set of keywords (barber, hair, salon) are clustered together, and a service with the same keywords has its own cluster when complete link clustering is applied with minimum similarity measure. Average link method clusters the services with these keywords correctly but for services with other keywords (sauna, holiday, cottage), it clusters them in different groups even when using average similarity. This problem does not happen with matching similarity.

Another observation of minimum similarity with complete link clustering is that there appear many clusters with only one object, and a very large cluster that contains most of the other objects. Matching similarity leads to more balanced clusters with both algorithms. Interestingly, it also produces almost the same clusters with the two different clustering methods.

For more extensive objective testing, we should have a ground truth for the wanted clustering but this is not currently available as it is non-trivial to construct. We therefore make indirect comparison by using the *SC* criterion from [16]. The assumption here is that the smaller the value, the better is the clustering. Fig. 5 summarizes the SC-values for different number of clusters. The overall minima for complete link and average link are 131, 86, 146 (minimum, average and matching similarities) and 279, 96 and 140, respectively. Our method provides always the minimum *SC* value. The sizes of 4 biggest clusters in each case are listed in Table 3.

Table 3. The sizes of the four largest clusters for complete and average link clustering

Complete link				
Similarity:	Sizes of 4 biggest clusters			
Minimum	106	88	18	18
Average	44	22	20	19
Matching	27	23	19	17
Average link				
Similarity:	Sizes of 4 biggest clusters			
Minimum	22	12	10	8
Average	128	41	34	17
Matching	27	23	17	17

The effectiveness of the proposed method for displaying data with limited number of clusters still exists. The number of clusters is too large for practical use and we need to improve the clustering validity index to find larger clusters but without creating meaningless clusters. We also observed some issues in clustering that originate from the similarity measure of two words, which implies that better similarity measure would also be useful.

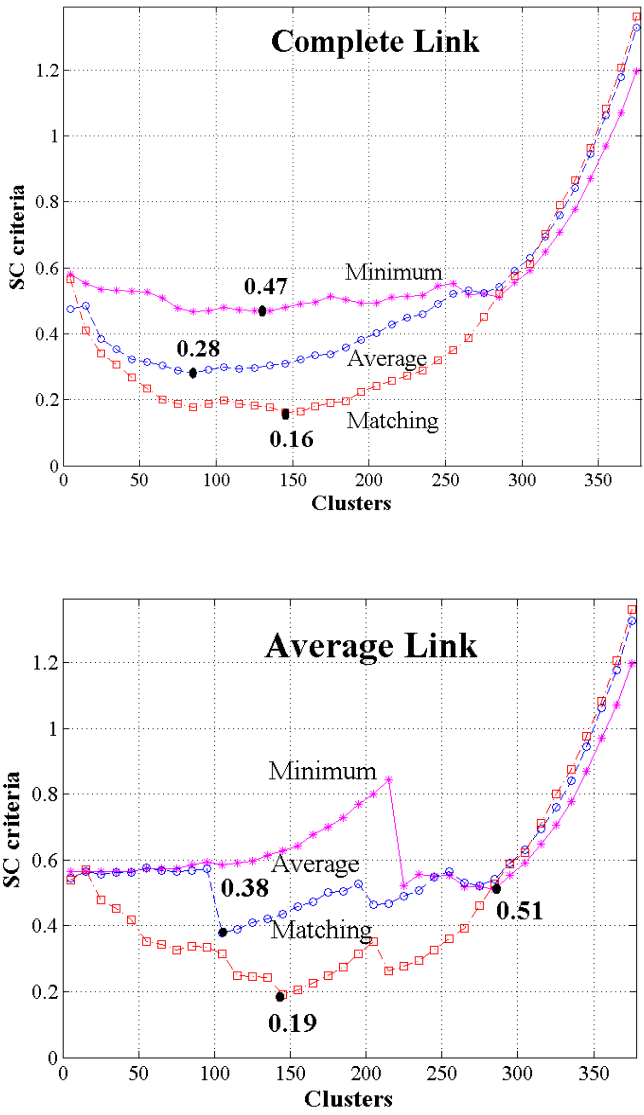


Fig. 5. Complete link and average link clustering using three similarity measures

4 Conclusion

A new measure called matching similarity was proposed for comparing two groups of words. It has simple intuitive logic and it avoids the problems of the considered minimum, maximum and average similarity measures, which fail to give proper results with rather simple cases. Comparative evaluation on a real data with SC criterion

demonstrates that the method outperforms the existing methods in all cases, and by a clear marginal. A limitation of the method is that it depends on the semantic similarity measure between two words. As future work, we plan to generalize the matching similarity to other clustering algorithms such as k-means and k-medoids.

Acknowledgements. This research has been supported by MOPIS project and partially by Nokia Foundation grant.

References

1. Aggarwal, C.C., Zhai, C.: A survey of text clustering algorithms. In: Mining Text Data, pp. 77–128. Springer US (2012)
2. Ricca, F., Pianta, E., Tonella, P., Girardi, C.: Improving Web site understanding with keyword-based clustering. Journal of Software Maintenance and Evolution: Research and Practice 20(1), 1–29 (2008)
3. Hasan, B., Korukoglu, S.: Analysis and Clustering of Movie Genres. Journal of Computing 3(10) (2011)
4. Ricca, F., Tonella, P., Girardi, C., Pianta, E.: An empirical study on keyword-based web site clustering. In: Proceedings of the 12th IEEE International Workshop on Program Comprehension. IEEE (2004)
5. Kang, S.S.: Keyword-based document clustering. In: Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages, vol. 11. Association for Computational Linguistics (2003)
6. Pereira, F., Tishby, N., Lee, L.: Distributional clustering of English words. In: Proceedings of the 31st Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics (1993)
7. Ushioda, A., Kawasaki, J.: Hierarchical clustering of words and application to NLP tasks. In: Proceedings of the Fourth Workshop on Very Large Corpora (1996)
8. Matsuo, Y., Sakaki, T., Uchiyama, K., Ishizuka, M.: Graph-based word clustering using a web search engine. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2006)
9. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: AAAI, vol. 6 (2006)
10. Cilibrasi, R.L., Vitanyi, P.: The google similarity distance. IEEE Transactions on Knowledge and Data Engineering 19(3), 370–383 (2007)
11. Bollegala, D., Matsuo, Y., Ishizuka, M.: A web search engine-based approach to measure semantic similarity between words. IEEE Transactions on Knowledge and Data Engineering 23(7), 977–990 (2011)
12. Wu, L., et al.: Flickr distance: a relationship measure for visual concepts. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(5), 863–875 (2012)
13. Budanitsky, A., Hirst, G.: Evaluating wordnet-based measures of lexical semantic relatedness. Computational Linguistics 32(1), 13–47 (2006)
14. Kaur, I., Hornof, A.J.: A comparison of LSA, WordNet and PMI-IR for predicting user click behavior. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM (2005)

15. Gledson, A., Keane, J.: Using web-search results to measure word-group similarity. In: Proceedings of the 22nd International Conference on Computational Linguistics, vol. 1. Association for Computational Linguistics (2008)
16. Zhao, Q., Rezaei, M., Chen, H., Franti, P.: Keyword clustering for automatic categorization. In: 2012 21st International Conference on Pattern Recognition (ICPR). IEEE (2012)
17. Michael Pucher, F.T.W.: Performance Evaluation of WordNet-based Semantic Relatedness Measures for Word Prediction in Conversational Speech (2004)
18. Markines, B., et al.: Evaluating similarity measures for emergent semantics of social tagging. In: Proceedings of the 18th International Conference on World Wide Web. ACM (2009)
19. Berry, M.W., Dumais, S.T., O'Brien, G.W.: Short text clustering by finding core terms. Knowledge and Information Systems 27(3), 345–365 (2011)

Quantum vs Classical Ranking in Segment Grouping

Francisco Escolano, Boyan Bonev, and Edwin R. Hancock

Department of Computer Science and AI, University of Alicante, Spain
Department of Statistics, UCLA, USA
Department of Computer Science, University of York, UK

Abstract. In this paper we explore the use of ranking as a mean of guiding unsupervised image segmentation. Starting by the well known Pagerank algorithm we introduce an extension based on quantum walks. Pagerank (rank) can be used to prioritize the merging of segments embedded in uniform regions (parts of the image with roughly similar appearance statistics). Quantum Pagerank, on the other hand, gives high priority to boundary segments. This latter effect is due to the higher order interactions captured by quantum fluctuations. However we found that qrank does not always outperform its classical version. We analyze the Pascal VOC database and give Intersection over Union (IoU) performances.

Keywords: random walks, quantum walks, ranking, segment grouping.

1 Introduction

When applied to image segmentation, random walks have been used to propagate labels in a semi-supervised way. For instance, in [1] pixels are labeled in terms of the probability that a random walk will reach them from a given seed.

However, the random walker approach assumes that the weighting function quantifying the dissimilarity between pixel intensities is symmetric. From a graph theoretic perspective this simplifies the problem since the Laplacian matrices of undirected graphs are semi-definite positive. At the same time the resulting asymmetric dissimilarity functions are richer since their directionality allows us to deal with special cases which are particularly interesting in image segmentation. For instance, a symmetric dissimilarity between adjacent segments (e.g. superpixels) imposes a misleading transitivity which may lead to an incorrect grouping. In Fig. 1 (bottom-left), segment X is very compatible with A and B in terms of having similar statistics. However A is in turn more compatible with B than with X. Therefore, A's best candidate for a merging will be B instead of X. This situation also occurs with second-order neighbors (compare X with E).

Incorporating assymetry into semi-supervised labeling has been done in the area of machine learning. For instance, in [2] conditional probabilities are introduced in the Markov chain, whereas in in [3] the graph Laplacian is symmetrized for encoding the directness of the edges. However, to the best of our knowledge

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 203–212, 2014.

there have been no attempts in the literature to investigate the propagation of information through the digraphs induced by asymmetric dissimilarity measures in an unsupervised context.

In this paper, we explore the impact of both classical and quantum ranking in the selection of the segments to merge in unsupervised segmentation. Our hypothesis is that ranking may improve significantly the quality of the segmentation, since the result of the process contains the information of random or quantum walks probing the network given by the adjacency graph. Therefore, ranking provides local-to-global information that may be critical in a greedy merging process.

The remainder of the paper is organized as follows. In Section 2 we describe the simple hierarchical grouping algorithm used for the study of ranking effects. In Section 3 we review Pagerank from a perspective of digraphs. Section 4 is devoted to the description of the quantum extension of the Pagerank method. Experiments and analysis are presented in Section 5. Finally, in Section 6, we present our conclusions and future work.

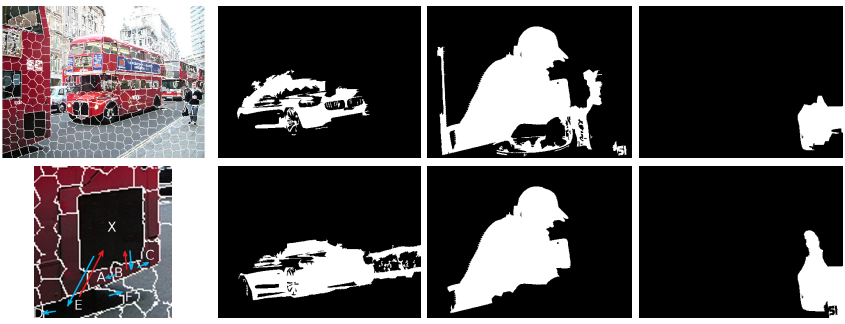


Fig. 1. Symmetric vs asymmetric dissimilarity. Top-left: Output of SLIC algorithm. Bottom-left: Asymmetric dissimilarities (in red and blue) between superpixel X and some of its 1^{st} and $2^{nd}-$order neighbors (see text). Top-row: Best unsupervised segmentation results imposing symmetry for some VOC Pascal objects. Bottom-row: results by imposing asymmetry. See the scenes analyzed in Top-left of Fig. 2 and Fig. 3.

2 Hierarchical Grouping Algorithm

The algorithm starts with a basic set of segments which are output by the SLIC algorithm [4]. We then build a segmentation hierarchy by merging/composing segments as follows.

Each segment is described by an *appearance vector* $V = (\overline{\mu}, \overline{\sigma}, c_x, c_y, w, h)$, where $\overline{\mu}, \overline{\sigma}$ are the mean and the standard deviation of $(l, a, b, \nabla_x, \nabla_y, \nabla_x^2, \nabla^2)$, and (c_x, c_y, w, h) are the centroid of the segment and the dimensions of its bounding box. These appearance vectors are designed so that they can be efficiently computed recursively for new segments composed by merging existing ones.

Each segment has a neighborhood structure. This consists of $1^{st}-$order neighbors, which are directly adjacent to the segment, and $2^{nd}-$order neighbors (i.e. those adjacent to the $1^{st}-$order neighbors). Then, we define an asymmetric similarity function $\Delta^A_{i|j}$ between segments which are 1^{st} or 2^{nd}-order neighbors. An appearance measure is defined to be:

$$\Delta^A_{i|j} = ||V_i - V_{i \bigcup j}||_2.$$

This is the change in the appearance vector of region i caused by merging it with region j. This quantity is asymmetric – i.e. $\Delta^A_{i|j} \neq \Delta^A_{j|i}$. This quantity will encourage merging neighboring regions which have similar appearance vectors.

The appearance similarity measure is modified by an edge-term ($E_{i,j}$ ranging from 0 to 1) that computes the strength of the edge on the boundary between two adjacent regions. This edge term is computed very simply using the Sobel edge detector. The underlying intuition is that we reduce the similarity between adjacent regions if there is an edge between them. We do not introduce an edge-term between segments in the 2^{nd}-order neighborhood (because we want this type of merging to jump between regions) and instead we pay a fixed penalty of size 1 (which is the maximum value the edge term can take).

This gives an asymmetric similarity function $\Delta_{i|j}$:

$$\Delta_{i|j} = \begin{cases} E_{i,j} + \Delta^A_{i|j} & \text{if } i,j \text{ are } 1^{st}-\text{order neighbors} \\ 1 + \Delta^A_{i|j} & \text{if } i,j \text{ are } 2^{nd}-\text{order neighbors} \end{cases}$$

Alternative segmentation algorithms such as the one in [5] can be used. Herein we use the simple method described above in combination with PageRank algorithm or with its quantum extension to rank the pairing between segments (i, j) on the basis of the similarity function. We allow the 30 % highest ranked segments to merge, see Fig. (1). This ranking encourages merging between segments which are most similar. However we reject merges in situations where the similarity function between two regions is too asymmetric (i.e. we do not allow merges where i "likes" j, but j does not "like" i). After these merges, we re-compute the PageRank algorithm and repeat the process.

3 Ranking Based on Random Walks

Since the similarity measure used for merging is asymmetric, we use a directed graph for encoding each segmentation level in the hierarchy.

A directed graph (digraph) $G = (V, E)$ with $N = |V|$ vertices and edges $E \subseteq V \times V$ is encoded by an adjacency matrix $\boldsymbol{A}$ where $A_{ij} > 0$ if $i \rightarrow j \in E$ and $A_{ij} = 0$ otherwise (this definition includes weighted adjacency matrices). The outdegree matrix D^{out} is a diagonal matrix where $d^{out}_i = \sum_{j \in V} A_{ij}$. The transition matrix $\boldsymbol{P}$ is defined by $P_{ij} = \frac{A_{ij}}{d^{out}_i}$ if $(i, j) \in E$ and $P_{ij} = 0$ otherwise.

The transition matrix is key to defining random walks on the digraph and P_{ij} is the probability of reaching node j from node i. Given these definitions

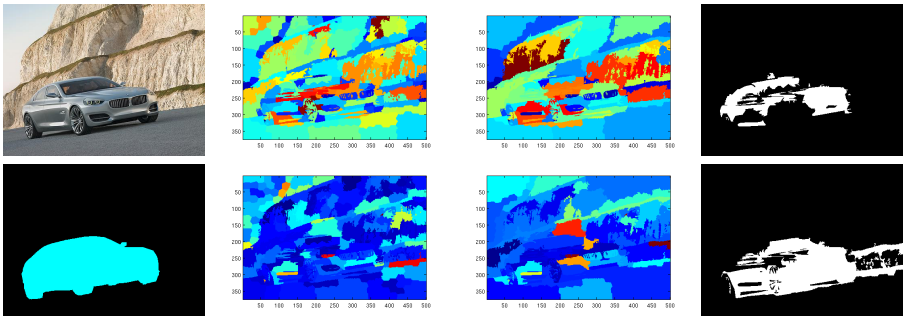


Fig. 2. Quantum vs classical ranking for car segmentation. Top-left: image. Bottom-left: ground truth. Top-center: classical ranking of segments at iterations 8 and 13. Bottom-center: quantum ranking of segments at the same iterations. Top-right: grouping result with the hierarchical algorithm based on the classical ranking (IoU = 0.43). Bottom-right: result using quantum ranking (IoU=0.61).

we have that $\sum_{j \in V} P_{ij} \neq 1$ in general. In addition, $\boldsymbol{P}$ is irreducible iff G is strongly connected (there is path from each vertex to every other vertex). If P is irreducible, the Perron-Frobenius theorem ensures that there exists a left eigenvector ϕ satisfying $\phi^T \boldsymbol{P} = \lambda \phi^T$ and $\phi_i > 0 \, \forall i$. If P is aperiodic (spectral radius $\rho = 1$) we have $\phi^T \boldsymbol{P} = \rho \phi^T$ and all the other eigenvalues have an absolute value smaller that $\rho = 1$. By ensuring strong connection and aperiodicity we also ensure that any random walk in a directed graph satisfying these two properties converges to a unique stationary distribution.

By correcting $\boldsymbol{P}$ so that $P_{ij} = \frac{1}{N}$ if $A_{ij} = 0$ and $d_i^{out} = 0$, we obtain a row stochastic matrix: $\sum_{j \in V} P_{ij} = 1 \forall i$. This strategy is adopted in Pagerank [6] and provides and allows for *teleporting* acting on the random walk to any other node in the graph. Teleporting is modeled by defining $\boldsymbol{G} = \eta \boldsymbol{P}^T + (1-\eta)\frac{\boldsymbol{e}\boldsymbol{e}^T}{N}$, where $\boldsymbol{e}^T$ is the all ones row vector and $0 < \eta < 1$. The new matrix G is column stochastic and ensures both irreducibility and aperiodicity. Under these conditions G_{ji} is the probability of reaching j from i. Teleporting means that for every node with $A_{ij} > 0$, $G_{ji} = \frac{A_{ij}}{d_i^{out}}$ is applied with probability η, whereas for nodes with $A_{ij} = 0$ we have $G_{ji} = \frac{1}{N}$ with probability $1 - \eta$. In [7] a trade-off between large values η (preserving more the structure of P') and small ones (potentially increasing the spectral gap) is recommended. For instance, in [3], where the task is to learn classifiers on directed graphs, the setting is $\eta = 0.99$, but usually $\eta = 0.85$ is recommended. In any case, when using the new P we always have that $G_{ii} \neq 0$ due to the Pagerank masking.

Finding the stationary distribution ϕ can be then formulated as an eigenvector problem $\boldsymbol{G}\phi = \phi$ subject to a normalization constraint $e^T \phi = 1$ (see [8]). Usually the power method is used. Accordingly iterate $\phi(k+1) = \boldsymbol{G}\phi(k)$ starting by $\phi(0) = e\frac{1}{N}$ until convergence (which will occur if the second eigenvalue λ_2 is smaller than $\lambda_1 = 1$). The stationary distribution is used for ranking the nodes.

4 Quantization of Random Walks

4.1 Unitary vs Stochastic Evolution

The above process for finding the stationary distribution simulates the diffusion of a discrete-time classical random walk on the directed graph $G = (V, E)$. Then the states are the nodes and top-ranked nodes are those whose stationary probability is high.

On the other hand a discrete-time quantum walk [9] diffuses in a very different way since it is subject to quantum superpositions. In this approach states $|\psi\rangle \in \mathbb{C}^N$ are assumed to belong to a Hilbert space $\mathcal{H} = span\{|j\rangle \,|\, j = 1, \ldots, N\} = \mathbb{C}^N$ where $\langle j| = (0 \ldots 1 \ldots 0)$ with a 1 at the $j - th$ position. We use the Dirac braket notation where: $|a\rangle = \boldsymbol{a}$, $\langle a| = \boldsymbol{a}^*$, $\langle a|b\rangle = \boldsymbol{a}^*\boldsymbol{b}$ is the inner product and therefore $\langle j|k\rangle = \boldsymbol{j}^*\boldsymbol{k} = \delta_{jk}$. Then, the state of the quantum walk at a given time is $|\psi\rangle = \sum_{j=1}^{N} c_j|j\rangle$ with $c_j \in \mathbb{C}$ so that $|c_1|^2 + |c_2|^2 + \ldots + |c_N|^2 = 1$ and $|c_i|^2 = \overline{c}_i c_i$. The probability that the quantum walk is at node i is given by $|\langle i|\psi\rangle|^2 = |c_i|^2$. The $|c_i|^2$ are known as the amplitudes of the wave traveling through the graph.

Given a initial state $|\psi(0)\rangle = \sum_{j=1}^{N} c_j^0|j\rangle$, a quantum walk evolves through a unitary operator instead of a stochastic one which is the case of random walks do. A $N \times N$ complex matrix $\boldsymbol{U}$ is unitary if $\boldsymbol{U}^*\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^* = I_N$, where $\boldsymbol{U}^*$ is the conjugate transpose, that is $(A^*)_{ij} = \overline{A_{ji}}$. Therefore, both the rows and columns of $\boldsymbol{U}$ form a orthonormal basis in $\mathbb{C}^N$. In addition $\boldsymbol{U}$ is by definition a normal matrix for it commutes with its conjugate transpose. In this case it is unitarily similar to a diagonal matrix, i.e., it is diagonalizable by $\boldsymbol{U} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^*$ where $\Lambda = diag(\lambda_1 \lambda_2 \ldots \lambda_N)$ contains the eigenvalues of $\boldsymbol{U}$ and $\boldsymbol{V}$ is unitary and its columns contains the eigenvectors of $\boldsymbol{U}$. Combining the latter diagonalization with the property $|det(\boldsymbol{U})| = 1$ we have that all the eigenvalues of $\boldsymbol{U}$ must lie on the unit circle. In other words, they must have either the form $e^{i\theta}$ or the form $e^{-i\theta}$, where θ is an angle on the complex plane.

Therefore we have $|\psi(t)\rangle = \boldsymbol{U}^t|\psi(0)\rangle$ with the amplitudes of $|\psi(t)\rangle$ summing to unity since $\boldsymbol{U}$ is unitary.

4.2 Szegedy's Quantization

The problem of associating a unitary operator with a Markov chain (stochastic matrix) has been posed in different ways. One of them is inspired in the Grover's search algorithm [10]. Grover's search relies on projection operators $\Pi = \sum_{j=1}^{N} |\Psi_j\rangle\langle\Psi_j|$ where, for instance, $|\Psi_j\rangle = \sum_{k=1}^{N} \frac{1}{\sqrt{N}}|k\rangle$. The projectors satisfy the condition $\Pi^2 = \Pi$ and the operator $2(\Pi - \mathbf{1})$ defines reflections (coin flips) around the subspace spanned by vectors $|\Psi_j\rangle$. In [11] Szegedy uses a product of reflections for quantizing a Markov chain.

To commence, the state space, originally placed at the N nodes, is moved to the $N \times N$ directed edges of the graph. The Hilbert space is now $\mathcal{H} = span\{|i\rangle_1|j\rangle_2 : |\, i, j = 1, \ldots, N\} = \mathbb{C}^N \otimes \mathbb{C}^N$ where $|a\rangle|b\rangle = |a, b\rangle = |a\rangle \otimes |b\rangle$ is

the tensor (Kronecker) product, and the subindexes 1 and 2 make explicit the orientation of each edge. Following the approach in [12][13] orientation is critical when computing projections onto the vector encoding the second node of the edge. Then, the superposition of the edges outgoing from node j are given by

$$|\Psi_j\rangle = \sum_{k=1}^{N} \sqrt{G_{kj}} |j\rangle_1 |k\rangle_2 \ . \tag{1}$$

Given the projector $\Pi = \sum_{j=1}^{N} |\Psi_j\rangle\langle\Psi_j|$ (coin flip) and the swap operator $\boldsymbol{S} = \sum_{j=1}^{N}\sum_{k=1}^{N} |j\rangle|k\rangle\langle k|\langle j|$ which alternates the direction of the edge (swaps both edge spaces), a step of the quantum walk is given by the unitary operator $\boldsymbol{U} = \boldsymbol{S}(2\Pi - \mathbf{1})$, where $\mathbf{1}$ is the identity matrix. However, Grover's search requires a two-step unitary operator per iteration. When translating this idea to Markov chains, Szegedy suggested the use of the operator $(2\Pi - \mathbf{1})(2\Pi' - \mathbf{1})$ in order to contemplate the case of two Markov chains (each one with its own reflection operator). When the two chains are coincident (e.g. for creating bipartite walks) then we have $(2\Pi - \mathbf{1})^2$. If we include the swap operator, which is also unitary, then the two-step evolution operator is given by $\boldsymbol{U}^2 = (2\boldsymbol{S}\Pi\boldsymbol{S} - \mathbf{1})(2\Pi - \mathbf{1})$. This operator swaps the directions of the edges an even number of times.

The initial state $|\psi(0)\rangle = \frac{1}{\sqrt{N}} \sum_{j=1}^{N}\sum_{k=1}^{N} |j\rangle|k\rangle$ assumes uniform probabilities for all the $N \times N$ edges in the digraph. Given $|\psi(0)\rangle$, we have that the state of the quantum walk at a given time t is given by $|\psi(t)\rangle = \boldsymbol{U}^{2t}|\psi(0)\rangle$. In addition, the probability of being at such state is $\langle\psi^*(t)|\psi(t)\rangle$. However, for ranking a given node i it is desirable to compute the probability of being at i at time t. This can be done by computing the probability that any edge ends at such node after t time steps. Let the state $|I_q(t)\rangle$ be the superposition of all paths ending at $|i\rangle_2$ (the second space of each edge ending at i).

The superposition state is defined as $|I_i(t)\rangle = {}_2\langle i|\psi(t)\rangle = {}_2\langle i|\boldsymbol{U}^{2t}|\psi(0)\rangle$ and it is given by the projection of $|\psi(t)\rangle$ onto the space $|i\rangle_2$. Such projection can be described more clearly if we exploit the spectral theorem, since we have $\boldsymbol{U}^{2t} = \sum_{\mu} \mu^{2t}|\mu\rangle\langle\mu|$, where the μ are the N^2 eigenvalues of $\boldsymbol{U}^2$ and the $|\mu\rangle$ are their corresponding N^2-dimensional eigenvectors. Then, ${}_2\langle i|\boldsymbol{U}^{2t} = \sum_{\mu} \mu^{2t}\, {}_2\langle i|\mu\rangle\langle\mu|$. If we consider that the structure of ${}_2\langle i|\mu\rangle$ is ${}_2\langle i|j\rangle_1|k\rangle_2$, for $|\mu\rangle$ is defined in the tensor space $\mathcal{H} = \mathbb{C}^N \otimes \mathbb{C}^N$, then we have that the proper projection is given by the contraction ${}_2\langle i|k\rangle_2|j\rangle_1$.

Consequently, we have that the probability that the quantum walk at vertex i after t time steps (that is, its *quantum ranking* at this time) is

$$I_i(t) = \langle\psi(0)|\boldsymbol{U}^{*2t}|i\rangle_2\langle i|\boldsymbol{U}^{2t}|\psi(0)\rangle = \left\| \sum_{\mu} \mu^{2t}\, {}_2\langle i|\mu\rangle\langle\mu|\psi(0)\rangle \right\|^2 \tag{2}$$

In practice, the time-averaged quantum ranking is used (although the long time average can be used since it only depends on the eigenvectors $|\mu\rangle$). The average is useful because quantum oscillation typically decrease in amplitude with $I_i(t)$. In any case, the main problem when this approach is applied to ranking segments is to compute or to approximate $\boldsymbol{U}^2$ (memory storage) and/or the its eigensystem.

4.3 The Eigensystem of the Unitary Operator

One of the nice properties of the Szegedy's formulation is that it provides a direct link between the eigenvalues and eigenvectors of $\boldsymbol{U}$ and those of a $N \times N$ matrix $\boldsymbol{D}$, where $D_{ij} = \sqrt{G_{ij}G_{ji}}$, called the *discriminant matrix*. Such matrix is linked with the projector operator Π and the *half-projection* $\boldsymbol{A} = \sum_{j=1}^{N} |\psi_j\rangle\langle j|$ through the following properties: (i) $\boldsymbol{A}^*\boldsymbol{A} = \boldsymbol{1}$, (ii) $\boldsymbol{A}\boldsymbol{A}^* = \Pi$ and (iii) $\boldsymbol{A}^*\boldsymbol{S}\boldsymbol{A} = \boldsymbol{D}$.

Let λ and $|\lambda\rangle$ be respectively the N eigenvalues and $N-$dimesional eigenvectors of the symmetric matrix $\boldsymbol{D}$. Then, if we define $|\tilde{\lambda}\rangle = \boldsymbol{A}|\lambda\rangle$ and apply the above properties, we have the following ansatz for the eigenvectors $|\mu\rangle$ and eigenvalues μ of $\boldsymbol{U}$: $|\mu\rangle = |\tilde{\lambda}\rangle - \mu\boldsymbol{S}|\tilde{\lambda}\rangle$. This means that we can easily obtain $2N$ of the N^2 eigenvalues and eigenvectors of $\boldsymbol{U}$ from the N ones of $\boldsymbol{D}$, since $\mu = e^{\pm i \times arccos\lambda}$. These latter values come from the SVD decomposition of $\boldsymbol{D}$, whose singular values lie in $(0, 1)$.

In addition, when we consider $\boldsymbol{U}^2$ we have that this operator splits $\mathcal{H}$ into the subspaces $\mathcal{H}_{dyn} = span\{|\psi_j\rangle, \boldsymbol{S}|\psi_j\rangle\}$ and its orthogonal complement $\mathcal{H}_{nodyn} = \mathcal{H}_{dyn}^{\perp}$. The dimension of $\mathcal{H}_{dyn}$ is at most $2N$. Thus, the spectrum of $\boldsymbol{U}^2$ corresponding to $\mathcal{H}_{dyn}$ is given by, at most, the $2N$ values $\{e^{\pm 2i \times arccos\lambda}\}$. The spectrum corresponding to $\mathcal{H}_{nodyn}$ is given by at least $N^2 - 2N$ 1's.

When estimating the eigenvectors $|\mu\rangle$ and eigenvalues μ for segmentations we have to confront the problem that at the lowest segmentation levels the number of SLIC superpixels is too high for building an $N^2 \times N^2$ unitary operator and then getting its eigensystem. The unitary operator is needed to extract the eigenvectors $|\mu\rangle$ corresponding to eigenvectors with value 1 (i.e. satisfying $\boldsymbol{U}^2|\mu\rangle = |\mu\rangle$). Obtaining the operator $\boldsymbol{U}^2$ (or even $\boldsymbol{U}$) is infeasible for these levels, unless a more in-depth analysis of the structure of these operators reveals a shortcut. This latter question is beyond the scope of this paper and we have approximated the instantaneous ranking $I_i(t)$ with

$$\tilde{I}_i(t) = \left\| \sum_{\mu \in \mathcal{H}_{dyn}} \mu^{2t} \, {}_2\langle i|\mu\rangle\langle\mu|\psi(0)\rangle \right\|^2 . \tag{3}$$

Therefore $\tilde{I}_i(t)$ is a low-pass approximation of $I_i(t)$.

5 Analysis: Experiments and Conclusions

We evaluate to what extent the averaged $\tilde{I}_i(t)$ can improve grouping when applied to rank segments at all levels of the hierarchy. In order to do that, we measure the Intersection-over-Union (IoU) which quantifies this quality of the segmentation of a particular object class (Pascal VOC 2010). This measure penalizes both obtaining a smaller area than the ground truth and obtaining a larger area than the ground truth. We have access to an unpublished ground truth where it has been assigned one of 57 labels to every pixel. For quantification data we use a sample of $1,100$ images of the $10,103$ in the VOC 2010

Table 1. IoU wrt the ground truth for 57 object classes of the Pascal VOC 2010

	plane	bicycle	bird	boat	bottle	bus	car	cat
CPMC	**78.6**	**64.4**	**74.8**	**71.2**	**74.9**	**78.9**	**72.1**	**85.5**
Uniform	53.7	39.7	55.8	48.3	54.6	46.5	51.7	57.1
R. Walks	53.3	41.9	57.8	56.9	51.7	52.5	53.9	60.5
Quantum	55.6	41.8	56.3	54.7	51.7	49.0	53.0	58.0
	chair	cow	d. table	dog	horse	motorbike	person	pot. plant
CPMC	51.7	**79.9**	**62.2**	**82.6**	**77.5**	**74.2**	**67.6**	**63.9**
Uniform	52.1	54.9	57.1	55.9	54.1	52.5	48.4	50.1
R. Walks	**52.9**	57.0	48.6	57.3	54.6	48.9	51.1	51.7
Quantum	49.7	57.1	51.0	57.5	51.8	49.2	50.3	51.6
	sheep	sofa	train	tv	bag	bed	bench	book
CPMC	**73.7**	**64.2**	**78.2**	**76.8**	53.2	**65.4**	36.5	36.3
Uniform	54.6	58.2	48.0	57.9	54.3	54.3	43.1	46.8
R. Walks	56.8	58.8	50.9	57.2	55.7	59.4	49.0	**52.9**
Quantum	56.1	57.5	48.5	57.5	54.9	62.4	**54.6**	55.1
	building	cabinet	ceiling	clothes	pc	cup	door	fence
CPMC	51.6	54.9	22.4	58.5	**70.7**	40.9	40.0	42.3
Uniform	51.8	55.5	54.6	57.3	48.8	47.3	50.8	44.5
R. Walks	**53.5**	54.6	**60.1**	57.8	48.6	46.8	56.2	45.7
Quantum	55.1	**59.0**	51.0	**58.8**	52.0	**51.2**	**59.4**	**47.4**
	floor	flower	food	grass	ground	keyboard	light	mountain
CPMC	54.5	**51.0**	44.9	56.5	55.0	46.8	8.3	55.2
Uniform	58.8	49.8	43.3	60.4	58.5	46.3	**43.7**	59.5
R. Walks	62.1	48.5	**56.9**	61.3	60.4	52.2	38.4	62.3
Quantum	**65.0**	49.9	50.7	**61.8**	**60.9**	**53.5**	42.5	**63.8**
	mouse	sign	plate	road	rock	shelves	sidewalk	sky
CPMC	12.6	24.6	**44.5**	56.3	**61.1**	**52.6**	53.0	65.1
Uniform	48.2	39.1	43.6	64.8	58.2	47.8	57.2	76.4
R. Walks	**48.4**	44.9	39.8	66.5	55.3	54.0	61.9	77.2
Quantum	48.3	**47.4**	39.0	**67.5**	58.2	49.9	**59.2**	**81.0**
	snow	table	track	tree	truck	wall	water	window
CPMC	**60.0**	**48.9**	**42.3**	54.5	**61.8**	51.7	**65.4**	50.9
Uniform	55.8	47.0	41.8	54.2	47.1	59.0	62.4	54.2
R. Walks	55.1	49.2	44.0	56.3	53.7	59.7	62.9	**57.9**
Quantum	59.1	47.2	41.7	**56.7**	55.8	**61.1**	65.1	56.1
	wood	**all IoU**						
CPMC	49.4	59.6						
Uniform	59.2	57.2						
R. Walks	**60.5**	58.7						
Quantum	60.3	**59.4**						

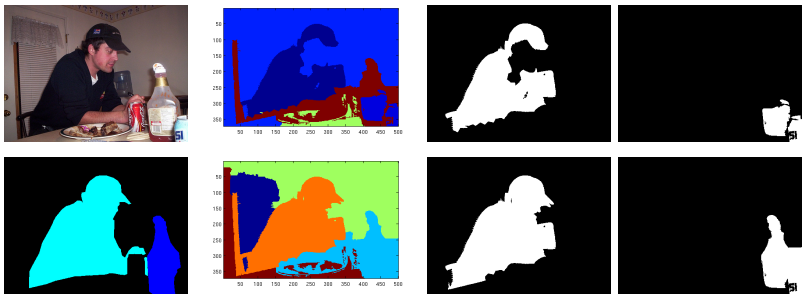


Fig. 3. Quantum vs classical ranking for person and bottle segmentation. First row, from left to right: image, classical rank at iteration 28, grouping result using classical rank for person (IoU=0.65), grouping using classical rank for bottle (IoU=0.36) . Second row, from left to right: ground truth, quantum rank at iteration 28, result using quantum rank for person (IoU=0.80), result with quantum rank for bottle (IoU=0.67).

dataset. In all cases the number of initial SLIC super pixels is 500 and there are 35 levels in the hierarchy until we reach a single segment.

Firstly, we analyze the behavior of Pagerank vs qrank (its low-pass approximation), before going through IoU analysis. In Fig. 2 we try to detect a car embedded in a textured environment. Pagerank seems to invert the priorities of qrank. It prioritizes the selection of segments inside quasi-homogeneous regions, whereas boundary segments have a low rank (blue). However, once the homogeneous region is built, its ranking decreases (see the grouping of the sky, in first row, which starts at interation 8 and it is stopped at iteration 13). Most top-ranked segments in qrank correspond to low-ranked ones in Pagerank (see the shadowed region at the top of the car). However, this "inversion" is misleading since some top/medium-ranked segments in qrank are also prioritized by qrank. On the one hand, the tendency of Pagerank to propose merging inside a quasi-homogeneous region may merge a part of the object with the background when the appearance of the background and the part of the object are similar. In the case of the car, this results in obtaining a smaller area than the ground truth. On the other hand, the behavior of qrank may lead us to merge the object with a part of the background producing a bigger area than the ground truth. These behaviors are replicated in Fig. 3 where qrank outperforms Pagerank when segmenting a person and a bottle.

We compare the IoUs obtained by both ranking methods with: (i) a state-of-the-art unsupervised segmentation algorithm (CPMC) and (ii) our hierarchical method with uniform ranking probabilities. We show the quantitative results in Table. 1. In general CPMC [14] outperforms our method. The first stage of the Constrained Parametric Min-Cuts method (CPMC) applies repeatedly a max-flow algorithm to output a set of segments based on groupings of an edge map provided by the gPb algorithm [15]. After a non-maximum supression filtering, the second stage ranks the segments using cues trained on the Pascal VOC dataset, provided the groundtruth masks of the objects.

However, our grouping method uses only low-level cues and is not trained for object-like segments. Thus, it is natural to outperform CPMC on background region classes, for which it is not trained. Also, our method tends to group similar appearance regions, while CPMC may group different appearance regions. This explains that CPMC outperforms our method on many foreground object classes and on the overall performance.

To conclude, qrank slightly outperforms its classical counterpart, and in some cases both outperform a state-of-the-art learning-based method. These experiments provide an initial insight of the power of quantum walks but we must complete the low-pass ranking with higher-order experiments which become feasible in segmentation settings.

Acknowledgements. Funding. F. Escolano: Project TIN2012-32839 (Spanish Gov.). E. R. Hancock: Royal Society Wolfson Research Merit Award.

References

1. Grady, L.: Random walks for image segmentation. TPAMI 28(11), 1768–1783 (2006)
2. Burges, C.J.C., Platt, J.C.: Semi-supervised learning with conditional harmonic mixing. In: Chapelle, O., Schölkopf, B., Zien, A. (eds.) Semi-Supervised Learning. MIT Press, Cambridge (2006)
3. Zhou, D., Huang, J., Schšlkopf, B.: Learning from labeled and unlabeled data on a directed graph. In: ICML, pp. 1041–1048 (2005)
4. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. TPAMI 34(11), 2274–2282 (2012)
5. Nock, R., Nielsen, F.: Statistical region merging. TPAMI 26(11), 1452–1458 (2004)
6. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web (1999)
7. Johns, J., Mahadevan, S.: Constructing basis functions from directed graphs for value function approximation. In: ICML, pp. 385–392 (2007)
8. Langville, A.N., Meyer, C.D.: Deeper inside pagerank. Internet Mathematics 1, 335–400 (2004)
9. Aharonov, Y., Davidovich, L., Zagury, N.: Quantum random walks. Phys. Rev. A 48, 1687–1690 (1993)
10. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: ACM Symposium on Theory of Computing, pp. 212–219 (1996)
11. Szegedy, M.: Quantum speed-up of markov chain based algorithms. In: FOCS, pp. 32–41. IEEE Computer Society (2004)
12. Paparo, G.D., Martin-Delgado, M.A.: Google in a quantum network. CoRR abs/1112.2079 (2011)
13. Paparo, G.D., M'uller, M., Comellas, F., Martin-Delgado, M.A.: Quantum google in a complex network. Scientific Reports 3, 2773 (2013)
14. Carreira, J., Sminchisescu, C.: Cpmc: Automatic object segmentation using constrained parametric min-cuts. TPAMI 34(7), 1312–1328 (2012)
15. Maire, M., Arbelaez, P., Fowlkes, C.C., Malik, J.: Using contours to detect and localize junctions in natural images. In: CVPR (2008)

Remove Noise in Video with 3D Topological Maps

Donatello Conte[1] and Guillaume Damiand[2]

[1] Université François-Rabelais de Tours, LI EA 6300, F-37200 France
[2] Université de Lyon, CNRS, LIRIS, UMR5205, F-69622 France

Abstract. In this paper we present a new method for foreground masks denoising in videos. Our main idea is to consider videos as 3D images and to deal with regions in these images. Denoising is thus simply achieved by merging foreground regions corresponding to noise with background regions. In this framework, the main question is the definition of a criterion allowing to decide if a region corresponds to noise or not. Thanks to our complete cellular description of 3D images, we can propose an advanced criterion based on Betti numbers, a topological invariant. Our results show the interest of our approach which gives better results than previous methods.

Keywords: Video denoising, 3D Topological Maps, Betti numbers.

1 Introduction

Several video analysis applications, like video surveillance or traffic monitoring, require as a preliminary sub-task the identification within the scene of the moving objects (foreground) as opposed to the static parts of the scene (background).

Many algorithms have been proposed in the literature most based on the background subtraction technique [17,3,11]. These algorithms are quite efficient but no one is the best for all situations (see [5] for a comparison of the most widely used background subtraction algorithms).

Some authors give up looking for an algorithm that directly provides the ideal foreground mask, and apply, instead, some post-processing in order to reduce or eliminate noise pixels, that is pixels erroneously detected as foreground. For example in [15] the authors show a method to remove shadows, or in [6] the authors propose some heuristics for removing some errors in the foreground mask. Even if these approaches are efficient, they are based on some assumptions that are not always true, being too dependent from the specific video characteristics.

Our paper fall in the last category: we propose an approach to reduce noise on foreground masks. But we present a general method that can be used on any video, in contrast to the more video-dependent approaches in the literature.

The basic idea of the method is that noise cannot be detected and removed analyzing a single frame of the video (as the other approaches do), but noise is easier to detect if more successive frames are examined: in fact real objects

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 213–222, 2014.

present, over the sequence, a regularity that noise seems not to have. Therefore the approach is based on a 3D structural representation of the foreground for a certain number of frames, and noise removal is done through structural operations on that data structure.

The remainder of the paper is organized as follows: in Sect. 2 the 3D structural representation of the scene is presented and explained, then in Sect. 3 the noise removal algorithm is given; the validation of the method, together with a comparison with other approaches, is made by a robust quantitative experimentation in Sect. 4; finally conclusions and perspectives are drawn in Sect. 5.

2 Definitions and Representation

We recall here the standard notions around 2D and 3D digital images, before introducing the notions of cellular subdivision and Betti numbers.

2.1 Digital 2D and 3D Images and Video

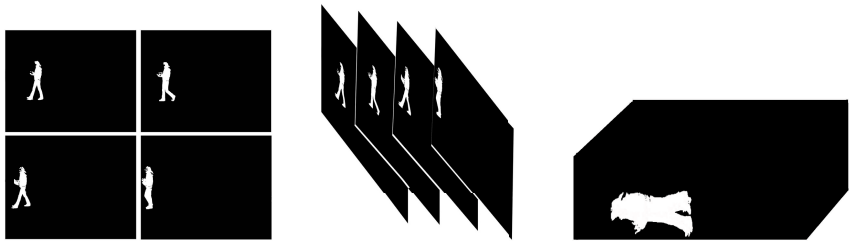
A *pixel* (resp. *voxel*) is an element of the discrete space $\mathbb{Z}^2$ (resp. $\mathbb{Z}^3$) denoted by its coordinates (x, y) (resp. (x, y, z)). A 2D (resp. 3D) *image* is a set of pixels (resp. voxels) and a mapping between these pixels (resp. voxels) and a set of colors or gray levels. Each pixel (resp. voxel) e is associated with its *color* or *gray level* $c(e)$. Furthermore, each pixel (resp. voxel) e is associated with a *label* $l(e)$ from a finite set of labels L. These labels can be obtained from the image by a segmentation algorithm.

In this work, a temporal sequence of 2D images is considered as a 3D image. Each image of the sequence is associated with a time t. Thus each pixel is now considered as a *temporal pixel*, described by three coordinates (x, y, t), (x, y) being the spatial coordinates and t being the temporal coordinate. Thus, a temporal sequence of 2D images can be seen as a 3D image, where each voxel is in fact a temporal pixel.

We use the classical notion of α-*adjacency*. Two voxels (x_1, y_1, z_1) and (x_2, y_2, z_2) are 6-adjacent if $|x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2| = 1$; they are 26-adjacent if $max(|x_1 - x_2|, |y_1 - y_2|, |z_1 - z_2|) = 1$ and they are 18-adjacent if they are 26-adjacent and if $|x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2| = 1$ or 2. Adjacency relations are extended to set of voxels: two sets of voxels S_1 and S_2 are α-adjacent if there is $v_1 \in S_1$ and $v_2 \in S_2$ such that v_1 and v_2 are α-adjacent.

Given an α-adjacency, an α-path between two voxels v_1, v_2 is a sequence of voxels starting from v_1 and finishing from v_2, such that two voxels of the sequence are α-adjacent. A set of voxels S is α-connected if there is an α-path between any pair of voxels in S having all its elements in S.

A region in 3D is a maximal set of 6-connected voxels having same label. In addition to all the regions present in the labeled image, another region is considered, called R_0, which contains all the voxels that do not belong to the image. R_0 is the complementary of the image.



(a) (b) (c)

Fig. 1. The 3D representation of a video. (a) The image sequence. (b) The construction of the 3D image. (c) The final representation.

2.2 Cellular Subdivision of Video

A video seen as a 3D image is thus decomposed in 3D regions which form a partition of the image (i.e. any voxel belongs to exactly one region and the union of all the regions is equal to the entire image). Figure 1 shows an example of the 3D representation of a video. The partition is decomposed in the following cells: 0-cells are vertices, 1-cells are edges, 2-cells are faces and 3-cells are volumes.

Volumes describe the boundaries of 3D regions. Each volume is bounded by a surface, i.e. a set of adjacent faces, each face corresponding to a maximal contact area between two adjacent regions. Faces are bounded by edges, each edge corresponding to a maximal contact between two adjacent faces; and edges are bounded by vertices. Incidence relations are defined between the cells: two cells are incident if they have different dimensions and if one belongs to the boundary of the second one.

This cellular subdivision is a generalization of a region adjacency graph (RAG [16]) which is a graph having a vertex for each region, and an edge between each pair of adjacent regions. Vertices of the graph correspond to regions, and edges correspond to faces which describe the adjacency relations. This RAG was extended in a multi-graph, called multi-RAG, in order to represent multi-adjacency relations between regions (when two regions are adjacent several times). However our cellular structure is much more rich than RAG and multi-RAG since it describes also the multi-adjacency relations but the relations are ordered (given a region we can iterate through the adjacent regions in an ordered way which is not directly possible with graphs); furthermore in our structure all the cells are represented (RAG instead describes only 3-cells and 2-cells).

The cellular subdivision is represented thanks to 3D topological maps [7], an efficient 3D model based on combinatorial maps [13,8] which represent the subdivisions in cells plus all the incidence and adjacency relations between the cells. In this work, 3D topological maps are used as an external tools and thus we do not go into detailed definitions, see above references for more details.

Having a cellular subdivision makes it possible to use some classical tool of algebraic topology, since our subdivision is an abstract cellular complex [2,12].

In this work we use Betti numbers, a well-known topological invariant [14], in order to characterize the topology of regions. These numbers are related to homology groups, but we give here their intuitive presentation. In 3D, there are three non null Betti numbers: $\mathfrak{b}_0$ is the number of connected components, $\mathfrak{b}_1$ is the number of tunnels and $\mathfrak{b}_2$ is the number of voids. For a region R, $\mathfrak{b}_0(R) = 1$ because by definition a region is connected, $\mathfrak{b}_1(R)$ is the number of *tunnels* of R (a tunnel corresponds to a path of voxels in R that cannot be contracted into a point) and $\mathfrak{b}_2(R)$ is the number of *voids* of R (a void is a set of connected voxels that do not belong to R but that are fully surrounded by voxels in R). An incremental method to compute Betti numbers and to update them during region merging is given in [10].

3 Our Method

The main principle of our method is detailed in Algo. 1. First a given video of foreground masks is cut in consecutive slices of 2D images. Then each slice is consider as a 3D image (as explained in the previous section) and a 3D topological map is built to describe the corresponding cellular subdivision. This gives a set of regions, each one being labeled 0 or 1 depending if it corresponds to a set of background voxels (0) or of foreground ones (1).

Algorithm 1: Reduce noise on foreground masks

Input: A video of foreground masks V;
A boolean function $criterion(r_1, r_2)$.
Result: V is modified by merging all the adjacent pairs of regions satisfying *criterion*.

```
T ← build the 3D topological map describing V;
foreach region R1 ∈ T labeled 1, R1 ≠ R0 do
    foreach region R2 adjacent to R1, R2 ≠ R0 do
        if criterion(R1, R2) then
            R ← merge(R1, R2);
            update region R;
return the partition described by T;
```

The 3D topological map corresponding to a given slice is built by using the algorithm given in [7] (wich is the extension in 3D of similar algorithm in 2D [9]). During this construction, a cube is created for each voxel, and 6-adjacent voxels having the same label are merged. Doing the merging during the construction allows to process large video by avoiding to build the full model composed of all the cubes describing all voxels.

Then, each pair of adjacent regions (R_1, R_2) are considered so that R_1 is labeled 1. Indeed in order to reduce the noise, it is enough to merge some white

regions with the background, thus there is no need to process black regions. If the pair (R_1, R_2) satisfies a given criterion, the two regions are merged. Merging two regions is done using the algorithm given in [10] which mainly consists in removing the faces separating the two regions, and possibly updating the edges and the vertices if needed.

Additional information associated with region R (which is the result of the merging of R_1 and R_2) must be updated. In this work, each region stores its number of voxels and its label. The number of voxels of R is the sum of the number of voxels of R_1 and the number of voxels of R_2. The label of R is always fixed to 0. Indeed, R_1 is labeled 1, thus by definition of regions, R_2 is labeled 0 (two adjacent regions can not have the same label). Since our goal is to reduce the noise, region R, considered as noise, and which is the merging of one background region and one foreground region, must stay in the background.

At the end of the algorithm, all the pairs of regions were considered and the new video is returned: this is the partition described by the modified 3D topological map.

The complexity of Algo. 1 is linear in number of adjacencies between regions times the complexity of the criterion. The number of adjacencies between regions is equivalent to the number of edges in the multi-RAG. Indeed thanks to the cellular decomposition we can iterate through all these adjacencies which are explicited by the faces, and the regions around each face are directly retrieved thanks to the incidence relations.

Now the main question is the definition of a criterion. Indeed, this is the main tool used during the reduce noise algorithm and only a correct definition will give good results. A first simple criterion, given in Eq. 1, consists in testing if the size of the white region is smaller than a threshold τ given by the user. The idea of this criterion comes from the fact that noise in image produces often small regions comparing to real objects. It is also important to highlight that for real objects there is always an overlapping between their appearances in consecutive frames, even at a low frame rate. Video used in experiments are at 10 fps and the overlapping between masks for real objects is always held.

$$size(R_1) < \tau \quad (1)$$
(R_1 being the region labeled 1)

The main interest of this criterion is to be very simple and computed in constant time since each region stores its size and the sizes are updated incrementally during the region merging. Note that this solution can be implemented using a multi-RAG data-structure instead of 3D topological maps. Indeed additional information described by topological maps are here not used.

One problem of this first criterion is that some white regions representing noise can have a size larger than the threshold and thus are not removed. By studying such regions, we have observed that they are often very porous because noise is non regular and noisy adjacent pixels have often different labels. For this reason, these regions have many voids and tunnels contrary to regions describing objects

which have generally a small number of voids and tunnels. This observation tends to show that the threshold associated with regions having many voids and tunnels must be increased in order to have an higher chance to be removed. For that, we propose in Eq. 2 a second criterion which mixes the size of the white region and its Betti numbers. This second criterion has two parameters: τ the threshold for the size of small regions, and φ, a percentage which is multiplied by the sum of the Betti numbers of R_1.

$$size(R_1) < \tau * (1 + \varphi * (\mathfrak{b}_1(R_1) + \mathfrak{b}_2(R_1))) \quad (2)$$
(R_1 being the region labeled 1)

This second criterion illustrates the interest of having an advanced description of regions (more precise than a RAG) allowing to compute and to mix several characteristics on regions. The complexity of this algorithm is equal to the complexity of the Betti number computation, i.e. linear in number of vertices, edges and faces describing region R_1. These numbers can be bounded by the number of voxels of R_1 times a constant number (8 for vertices, 12 for edges and 6 for faces).

4 Experiments

4.1 Dataset and Algorithms

We use the PETS 2010 Dataset [1]. This dataset is a standard database widely used for the performance evaluation of tracking and surveillance algorithms.

In order to evaluate the performances of the proposed denoising algorithm, we start from foreground detection masks on PETS video sequences, resulting from the application of a basic background subtraction (BS) algorithm. We expressly used the basic BS algorithm without any improvement and parameter optimization, because we want to show that the proposed algorithm can clean detection masks without any pre-processing prior. This allows to be not dependent on the specific video sequence and it avoids the optimization parameters phase that is tedious and not always possible.

Starting from the same detection masks, we compare our algorithm with:

A1 a denoising algorithm that uses only morphological operations (erosion and dilatation);
A2 the algorithm proposed in [6] that adds, to the basic subtraction algorithm, several post-processing improvements;
A3 the algorithm [6] with the addition of the grouping phase proposed by the same authors in [4].

As shown in [4], these algorithms are effective in reducing noise regardless of the method for foreground detection. Other approaches are not considered because of their high computational complexity. Note that the algorithms A2

and A3 require several parameters to set. A3 also requires a camera calibration phase (for details see [4]) for each video (taken with different camera settings). Therefore, in this experimentation we preliminary optimized these parameters, which are therefore specific for each sequence.

Our new method has two parameters: τ the threshold for the size, and nb which is the number of frames grouped in a same 3D slice. The method based on Betti numbers has an additional parameter: φ the percentage of Betti numbers added to the size.

4.2 Performance Index

We use an evaluation scheme inspired by the method presented in [18]; it takes into account one-to-one as well as many-to-one and one-to-many matches.

The goal of a detection evaluation scheme, on a frame, is to take a list of ground truth boxes $G = G_1, \ldots, G_n$ and a list of detected boxes $D = D_1, \ldots, D_m$ and to measure the quality of the match between the two lists. From the two lists D and G two overlap matrices σ and τ are created. The rows $i = 1 \ldots |G|$ of the matrices correspond to the ground truth boxes and the columns $j = 1 \ldots |D|$ correspond to the detected boxes.

The values are calculated as follows:

$$\sigma_{ij} = \frac{area(G_i \cap D_j)}{area(G_i)} \qquad \tau_{ij} = \frac{area(G_i \cap D_j)}{area(D_j)} \tag{3}$$

The matrices can be analyzed for determining the correspondences between the two lists:

One-to-One Matches: G_i matches against D_j if row i of both matrices contains only one non-zero element at column j and column j of both matrices contains only one non-zero element at row i. The overlap area needs to have a certain size compared to the rectangle in order to be considered successful ($\sigma_{ij} \geq e_1$ and $\tau_{ij} \geq e_2$).

One-to-Many Matches with One Ground Truth Box: G_i matches against several detected boxes if row i of the matrices contains several non-zero elements. The additional constraint of $\sum_j \sigma_{ij} \geq e_3$ ensures that the single ground truth rectangle is sufficiently detected.

One-to-Many Matches with One Detected Box: D_j matches against several ground truth boxes if column j of the matrices contains several non-zero elements. Also here we add the constraint of $\sum_i \tau_{ij} \geq e_4$.

Parameters $e_1, \ldots, e_4$ measure how much detected boxes against ground truth have to overlap. For most applications a value of 0.8 (80% of overlapping) is good; therefore we set $e_1 = \cdots = e_4 = 0.8$.

Based on this matching strategy, the recall and precision measures are defined as follows:

$$\text{recall} = \frac{\sum_i Match_G(G_i)}{|G|} \qquad \text{precision} = \frac{\sum_j Match_D(D_j)}{|D|} \tag{4}$$

where $Match_G(G_i)$ is defined as follows:

$$Match_G(G_i) = \begin{cases} 1 & \text{if } G_i \text{ matches against a single detected box} \\ 0 & \text{if } G_i \text{ does not match against any detected box} \\ 0.8 & \text{if } G_i \text{ matches against several detected boxes} \end{cases} \quad (5)$$

and $Match_D(D_j)$ accordingly.

The indexes Recall and Precision for a video sequence are the average values of recall and precision over all the frames of the sequence.

4.3 Results

Results of our experiments are given in Table 1 for the precision and recall measures, and in Table 2 for the F-score values (the harmonic mean of precision and recall). In all the arrays, dark grey cells are the best scores for each video, and light grey cells the second best scores. In these experiments, nb is always fix to 15. $tXXX$ is the value obtained by the method with the size criterion with $\tau = XXX$, and $tXXX\text{-}pYYY$ is the value obtained by the method with the size and Betti numbers criterion with $\tau = XXX$ and $\varphi = YYY$.

Table 1. The values of the indexes precision and recall for the considered algorithms

	v1		**v3**		**v4**		**v5**		**v6**		**v7**		**v8**	
	Rec	*Pre*	*Rec*	*Pre*	*Rec*	*Pre*	*Rec*	*Pre*	*Rec*	*Pre*	*Rec*	*Pre*	*Rec*	*Pre*
A1	**0.55**	0.09	**0.38**	0.27	**0.54**	0.24	**0.55**	0.16	0.44	0.45	**0.57**	0.12	**0.68**	0.14
A2	0.44	0.29	0.20	0.39	0.46	0.36	0.44	0.32	0.41	**0.55**	0.47	0.31	0.60	**0.38**
A3	0.49	0.22	0.31	0.42	0.52	0.33	0.50	0.20	0.43	0.47	0.52	0.19	0.65	0.24
t2000	**0.54**	0.20	0.36	0.45	**0.55**	0.37	**0.54**	0.26	**0.46**	0.50	**0.55**	0.23	**0.66**	0.25
t3000	**0.54**	0.20	0.35	**0.46**	**0.55**	**0.39**	0.53	0.28	**0.46**	0.50	**0.55**	0.22	**0.66**	0.25
t4000	0.53	0.21	0.33	**0.46**	**0.54**	**0.40**	0.53	0.28	**0.46**	0.51	0.54	0.23	**0.66**	0.26
t2000-p.05	**0.54**	0.23	0.34	**0.46**	**0.54**	0.38	0.53	0.33	**0.46**	0.53	0.54	0.33	**0.66**	0.29
t2000-p.1	0.48	0.28	0.22	**0.47**	0.31	**0.39**	0.51	**0.36**	0.44	**0.54**	0.46	**0.37**	0.56	0.32
t2000-p.15	0.39	0.36	0.08	0.34	0.09	0.37	0.45	**0.35**	0.36	0.52	0.37	**0.36**	0.47	**0.33**
t3000-p.05	0.50	0.26	0.25	0.45	0.53	0.38	0.53	**0.36**	**0.45**	**0.54**	0.50	0.34	0.62	0.31
t3000-p.1	0.37	0.37	0.07	0.28	0.08	0.32	0.45	**0.35**	0.35	0.52	0.36	**0.37**	0.46	**0.33**
t3000-p.15	0.23	0.37	0.03	0.14	0.03	0.17	0.32	0.30	0.29	0.46	0.27	0.32	0.34	0.29
t4000-p.05	0.47	0.29	0.17	0.44	0.26	0.36	0.49	**0.36**	0.44	**0.54**	0.46	**0.37**	0.55	**0.33**
t4000-p.1	0.26	**0.38**	0.03	0.14	0.03	0.17	0.36	0.32	0.32	0.48	0.29	0.34	0.39	0.30
t4000-p.15	0.14	**0.40**	0.01	0.07	0.01	0.06	0.21	0.22	0.19	0.33	0.22	0.29	0.25	0.28

These results show that our new method is competitive comparing with the three previous algorithms. Generally, merging more regions (either by increasing τ or by increasing φ) decreases the recall while increases the precision until a certain point. Thus better results are obtained by finding the good thresholds giving the best compromise for precision and recall.

These results show a second important conclusion: the method using Betti numbers can greatly improve the results. This is for example the case for video *v7* with $\tau = 2000$, where the precision is improved from 0.23 without Betti to 0.37 with Betti using $\varphi = .1$.

These results are confirmed by the F-score values given in Table 2 which allow to find the best compromise between precision and recall. For all videos, the best scores are often obtained by the method using Betti numbers with $\tau = 2000$ (best score for 3 videos, and second best score for the 4 other videos).

Table 2. The values of the F-score for the considered algorithms

	v1	**v3**	**v4**	**v5**	**v6**	**v7**	**v8**
	Fsc	*Fsc*	*Fsc*	*Fsc*	*Fsc*	*Fsc*	*Fsc*
A1	0.15	0.31	0.33	0.25	0.44	0.20	0.23
A2	0.35	0.26	0.40	0.37	0.47	**0.37**	**0.46**
A3	0.30	0.36	0.40	0.28	0.45	0.28	0.35
t2000	0.29	**0.40**	**0.45**	0.35	**0.48**	0.32	0.36
t3000	0.30	**0.40**	**0.45**	0.37	**0.48**	0.32	0.37
t4000	0.30	**0.39**	**0.46**	0.37	**0.48**	0.32	0.37
t2000-p.05	0.33	**0.39**	**0.45**	0.41	**0.49**	**0.41**	0.40
t2000-p.1	0.35	0.30	0.34	**0.42**	**0.49**	**0.41**	**0.41**
t2000-p.15	**0.37**	0.13	0.14	0.39	0.43	**0.37**	0.38
t3000-p.05	0.34	0.32	0.44	**0.43**	**0.49**	**0.41**	**0.41**
t3000-p.1	**0.37**	0.12	0.13	0.39	0.42	0.36	0.38
t3000-p.15	0.28	0.05	0.05	0.31	0.36	0.30	0.31
t4000-p.05	**0.36**	0.25	0.30	0.41	**0.49**	**0.41**	**0.41**
t4000-p.1	0.31	0.05	0.05	0.34	0.38	0.31	0.34
t4000-p.15	0.20	0.02	0.01	0.21	0.24	0.25	0.26

5 Conclusion

In this paper, we presented a new method of noise reduction on foreground video masks. Thanks to a 3D cellular description of the video, our method is defined in an high abstraction level considering regions and adjacency relations between these regions. This simplifies the denoising algorithm which consists mainly to merge foreground regions with the background. A second main advantage is the possibility of defining high level criteria on the regions. In this paper we use a simple criterion using the size of regions, and a more advanced criterion using Betti numbers (that gives better results). This second criterion can be defined thanks to the full cellular representation, while this is not directly possible using simpler data-structures such as region adjacency graph.

As future work, we first want to work on the automatic computation of the parameters of our method. A second perspective is to define other criteria. Thanks to our representation, many possibilities could be studied mixing geometrical

criteria and topological ones. A last perspective is to use similar techniques (considering the 3D cellular description of a video) in other fields of video processing such that objects or activities recognition.

Acknowledgement. This work has been partially supported by the French National Agency (ANR), project Solstice ANR-13-BS02-01.

References

1. Pets2001 dataset, `http://www.cvg.rdg.ac.uk/pets2001/`
2. Aleksandrov, P.S.: Elementary concepts of topology. Dover Publications Inc., New York (1961)
3. Barnich, O., Droogenbroeck, M.V.: Vibe: A universal background subtraction algorithm for video sequences. IEEE Transactions on Image Processing 20(6), 1709–1724 (2011)
4. Conte, D., Foggia, P., Percannella, G., Tufano, F., Vento, M.: An algorithm for recovering camouflage errors on moving people. In: Hancock, E.R., Wilson, R.C., Windeatt, T., Ulusoy, I., Escolano, F. (eds.) SSPR&SPR 2010. LNCS, vol. 6218, pp. 365–374. Springer, Heidelberg (2010)
5. Conte, D., Foggia, P., Percannella, G., Tufano, F.: Mario Vento. An experimental evaluation of foreground detection algorithms in real scenes. EURASIP Journal on Advances in Signal Processing 2010(373941), 1–11 (2010)
6. Conte, D., Foggia, P., Petretta, M., Tufano, F., Vento, M.: Evaluation and improvements of a real-time background subtraction method. In: Kamel, M.S., Campilho, A.C. (eds.) ICIAR 2005. LNCS, vol. 3656, pp. 1234–1241. Springer, Heidelberg (2005)
7. Damiand, G.: Topological model for 3d image representation: Definition and incremental extraction algorithm. CVIU 109(3), 260–289 (2008)
8. Damiand, G.: Combinatorial maps. In: CGAL User and Reference Manual. CGAL Editorial Board, 3.9 edn. (2010)
9. 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)
10. Dupas, A., Damiand, G.: Region merging with topological control. Discrete Applied Mathematics 157(16), 3435–3446 (2009)
11. Haque, M., Murshed, M.: Perception-inspired background subtraction. IEEE Trans. on Circuits and Systems for Video Technology 23(12), 2127–2140 (2013)
12. Kovalevsky, V.A.: Finite topology as applied to image analysis. CVGIP 46, 141–161 (1989)
13. Lienhardt, P.: N-Dimensional generalized combinatorial maps and cellular quasi-manifolds. Int. Journal of Computational Geometry and Applications 4(3), 275–324 (1994)
14. Munkres, J.R.: Elements of Algebraic Topology. Perseus Books (1984)
15. Prati, A., Mikic, I., Trivedi, M.M., Cucchiara, R.: Detecting moving shadows: algorithms and evaluation. IEEE Trans. on PAMI 25(7), 918–923 (2003)
16. Rosenfeld, A.: Adjacency in digital pictures. Information and Control 26(1), 24–33 (1974)
17. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE Trans. on PAMI 22(8), 747–757 (2000)
18. Wolf, C., Jolion, J.-M.: Model based text detection in images and videos: a learning approach. Technical report, LIRIS INSA de Lyon (2004)

Video Analysis of a Snooker Footage Based on a Kinematic Model

Aysylu Gabdulkhakova* and Walter G. Kropatsch

Vienna University of Technology
Pattern Recognition and Image Processing Group
{aysylu,krw}@prip.tuwien.ac.at

Abstract. Taking an inspiration from psychological studies of visual attention, the contribution of this paper lies in prediction of the critical points of the trajectory using the structure of a scene and physical motion model. On one side, we present our approach for video analysis that differs from traditional tracking techniques by predicting future states of the moving object rather than its next consecutive position using the physically-based motion functionality. On the other side, we propose to use the structure of the scene, which contains the information about the obstacles and space limits, for discovering the critical points of the trajectory. As a proof of concept we developed the use case application for analysing snooker footage.

1 Introduction

Tracking covers a vast number of applications in computer vision: from media production and augmented reality to robotics and unmanned vehicles [10]. By definition [10] it is a process which aims at defining the position of the target in subsequent video frames. For the purpose of video understanding and summarization storing the entire track of the objects is semantically and computationally redundant. Moreover, high frequency temporal sampling, high resolution data and contamination (i.e occlusions, distractors, illumination variations) makes tracking a challenging task to robustly perform in real time. Thus, we inquired into a question of reducing the amount of processed data by predicting the future locations of the analysed object and select then semantically meaningful moments which require further detailed analysis, such as recognition.

We derived our idea from psychological studies of human vision. Selective visual attention is a mechanism that aims at filtering irrelevant parts of the scene and focusing cognitive processes on the most important object at a given time slot. It is accomplished based on either spatial location or object features. It is worth mentioning that we are not aiming at modelling this phenomena. We use its main principles, in order to argue in favour of neglecting redundant data.

* Supported by the Austrian Agency for International Cooperation in Education and Research (OeAD) within the OeAD Sonderstipendien program, financed by the Vienna PhD School of Informatics.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 223–232, 2014.

As a descriptor motion is assumed to be an inherent property of an object taken apart from environment. Though, motion parameters are dependant not only on characteristics of the object, but also are restricted by the *structure of the scene*. The correct motion model enables to predict the trajectory, whereas structure of the scene defines the conditions when the trajectory deviates. *Critical point* of the trajectory - is a point of the trajectory where the parameters of motion are changed. On the example of a snooker game, motion of the ball is limited by other balls, cushion (table borders) and pockets.

The existing approaches for snooker game analysis, [3], [11], [8], aim at predicting the state of the balls in the current frame knowing its positions in previous frame(s). When the tracked data of the whole video is collected, they select a subsample for the tasks of summarization or event detection. In contrast, we propose that tracking every position of the target with known motion model is not influential in sense of video understanding. For this purpose, we predict the evolution of object's spatio-temporal changes according to its physical properties (motion functionality). This prediction is then taken in conjunction with the structure of the scene, in order to detect critical points of the trajectory and correct the parameters of original motion model.

The remaining of the paper is organized as follows. Next section introduces the idea of our approach. Section 3 is dedicated to the utilization of the proposed method for the application of tracking a snooker game. Each important issue is followed by the discussion of the state of the art approaches. The paper is concluded in Section 4.

2 Motion as a Descriptor for Tracking

One of the main components in the tracking pipeline is dedicated to modelling an object of interest, or shortly a target. There is no universal formal description of the key object properties that enable successful tracking in all possible cases. Generally, computer vision community is split into the adherents of the statistical and structural approach [2]. The common feature for both approaches is that they do not work with the real world objects, but with the image representations and their properties.

In our view the model of the real object should not be limited to the properties of the image. Tracking aims at analyzing dynamic scenes, which in turn reveal the spatio-temporal changes of the objects. These dynamic changes are mostly not an unpredictable phenomena with independent random states in each moment. By the nature motion behaviour of an object has limitations that basically relate to motion functionality of the object and/or to the structure of the scene (see Table 1).

Functionality, in our opinion, represents a set of abilities that the object possesses. From motion perspective, functionality defines how the object could deform with time according to its physical nature. On one hand, it is defined by the structure of the object (e.g. rigid/non-rigid parts, degrees of freedom). On the other hand, it depends on the local motion models of the constituent parts and the global motion model of the whole object.

Table 1. Criteria impacting the motion behaviour

motion limitations			
functionality		**structure of the scene**	
structure of the object	motion model	space limits	obstacles

Structure of the scene is a set of objects and conditions influencing the motion trajectory of the target. We consider obstacles and space limits to be of higher importance. Obstacle is an object of the scene which lies on the trajectory of the moving target. As a result the original motion model of the target is changed either after a collision or due to bypassing. Obstacle is not a permanent motionless part of a structure of the scene and may change its position. In contrast, space limits are constant, they constraint further motion of the target, such that bypassing is not feasible. Though collision and reflection scenarios are possible.

The awareness about the above phenomena provides advantages as opposed to other sources of guidance in the following current tracking problems:

- keep tracking in case of a partial or/and complete occlusion;
- prediction of the critical points of trajectory;
- lower the computational costs since processing only meaningful data.

Multiple Facets of Motion Functionality. Prediction based on functionality makes a crucial impact and importance in a wide range of applications varying from computer graphics to robotics.

In robotics time is of critical importance since interacting with dynamically changing environment. Since 1980s the motion model of the flying ping pong ball is used for its trajectory prediction [1, 5], in order to configure the paddle for a successful ball return. Another example relates to a motion planner, where the incorrectness of the robot's navigation may lead to an injury of a human. Robot uses the predicted trajectories of the moving human, in order to prevent the collision risks [13].

Reconstructive facial surgery is a complex, radical (drastic) and irreversible procedure. The functionality of facial muscles enables accurate bio-mechanical facial soft tissue modelling and post-operative result simulation. Recent achievements in this direction relate to preoperative simulation of craniofacial surgery on bones with respective soft tissue alterations [14]. Other areas of application are human face visualization and mimics recognition [7], where the functionality provides robustness for the approach.

Approaches based on Kalman filter [6] are widely used in navigation systems and computer vision. This recursive physics-based method supports the estimation of the current position of the object taking into consideration previous states, measurements of the current state and a Gaussian nature of noise.

The distinct feature of the proposed approach is that the motion of the object is not taken apart and is observed in conjunction with the structure of the scene. It enables not only to predict the future spatio-temporal states of the object according to the given motion model, but also to be aware of the critical points where the trajectory can deviate from original.

3 Analysing a Snooker Game

The intention of this section is to show the applicability of the proposed approach to the real problem. For this purpose we selected the snooker game footage analysis, but the same methodology can be used in other domains for tracking objects with defined motion model and structure of the scene.

3.1 Motivation

Snooker is a variety of pool played with 21 balls of 6 distinct colors and a white ball (cue ball). The goal is to pot the color balls with a cue ball in a particular order and gain more points than the opponent. While watching or playing this game people are not tracking the positions of the moving balls as the time flows. On the contrary, they try to predict future positions and pay attention to the prominent ones. The correct model of the ball's movement should consider physical properties of a cue stick, baze, table and balls as well as several forces: rolling resistance, sliding friction, self-rotation [12]. In order to achieve planned trajectories, players use the effect of ball spin, the reflection from the cushion, or both. In case of disadvantageous position it makes sense to create a loosing situation for the opponent while his turn. Overall, it is preferable to hit as less balls as it is needed, in order to be able to predict the next state of the game. From this hypothesis it is obvious that tracking positions of all the balls in all the frames is not needed. Moreover, tracking the moving balls in all the frames is redundant when this move has a predictable trajectory. Using the abstract concept of visual attention together with the structure of the scene and functionality for predicting time and location of critical points of the trajectory becomes natural.

3.2 State of the Art in Snooker Video Analysis

Sport video analysis is widely represented in computer vision community: from semantic event detection and summarization to computer-assisting referee systems. A frequent engineering approach is to combine several existing methods for solving a particular task. Thus, the tools that seem to work on different problems have issues in common. For example, Denman et al. [3] introduced several approaches for video parsing, event detection and shot activity summarization in snooker footage analysis. This includes table shots detection rested on geometry and Hough Transformation, tracking a cue ball using color-based particle filter and detecting pots by histogram analysis of pocket regions. On the basis of this

work with a modified ball tracking method Rea et al. [11] build a system for semantic event detection. For 3D reconstruction from snooker video [8] consider ball movements to be of more semantic importance as opposed to players and cue stick. In this manner they apply detection, classification and tracking of the balls. Tracking the objects of interest is a building block in these tools. According to its definition [10], searching for the object of interest is performed in subsequent frames. In contrast, the idea of our approach is to predict the evolution of object's temporal changes using its physical properties.

3.3 Description

The above mentioned human attention strategy (see Section 3.1) is modelled using motion functionality and structure of the scene. Prediction of future trajectories of the moving balls in conjunction with the space limits and positions of obstacles guide the selection of the prominent parts of the scene with corresponding time slots.

Functionality. Having the set of the balls B=$\{b_1, b_2, ..., b_{21}, b_{cue}\}$, the functionality is defined by a set $b_i \in B = \{S, V, a\}$, where S is a vector of positions of the ball in consecutive video frames $(s_i = (x_i, y_i), s_i \in S, i = \overline{1, N})$, V - a vector of velocity values for consecutive pairs in vector S, a - acceleration of the ball. The motion of the ball is constrained to uniformly accelerated linear model.

Remark 1. The velocity values $v_i \in V$ are computed as the first derivative of distance with respect to time, and acceleration a as a second derivative. The quality of the ordinary footage does not allow to precisely measure the velocity for obtaining a real value of acceleration. For that we consider Gaussian distribution of acceleration for getting the most likely value of it. When the velocity is close to 0, it is assumed that the target does not move.

Remark 2. The velocity of the ball decreases in every consecutive time slot. The exception of this rule occurs when during cushion collision the vector of ball-spin supplements with the vector of ball-movement. As a result both the rebound angle and rebound velocity increase.

Remark 3. The accurate model of the ball's movement should consider physical properties of a cue stick, baze, table and balls as well as several forces: rolling resistance, sliding friction, self-rotation [12]. Particularly, the rotational component of the moving ball makes an impact on the resultant trajectory such that the rebound angle from the cushion is deviating from perfect reflection, and the motion parameters change.

In order to obtain the consecutive positions S of the balls, traditional tracking approaches are combined in the following way. The first movement before each shot corresponds either to the cue ball, or to the cue stick. Optical flow

technique [4] is combined with the blob detection algorithm to get the moving parts on the table. When the motion is detected, we check this region for the white ball using a circular Hough transform combined with color thresholding. The idea of color thresholding is to find such a circle in a region, which has the highest density of pixels exceeding value of 200 at each of RGB levels. Next a "snapshot" of this ball is taken and in the next frames apply template matching technique. Template matching [9] is a procedure of finding a region in an image, that correlates the most with the given template. Due to the changes of the template caused by perspective projection and occlusions by the players or other balls, we perform Kalman filter, in order to obtain comparably robust track.

Structure of the Scene. According to the definition in Section 2, structure of the scene contains obstacles and space limits. In this application, obstacles are represented by the balls other than the cue (white) ball. After the hit they change the trajectory of the cue ball and move under their own functionality. The space limits are represented by the area of the table and the billiard-pockets. Under the rules, balls cannot cross the table borders. After cushion collision the ball's trajectory is changed due to the laws of reflection. Billiard-pocket is a part of the structure of the scene where the ball's trajectory ends.

Obstacles. At the beginning of each shot all the balls except the white are static. Obstacle-ball becomes of importance when belonging to the trajectory of the moving ball. The route of the moving ball is represented as a binary mask:

$$I(x,y) = \begin{cases} 1, (x,y) \in route \\ 0, otherwise \end{cases}$$

where I(x,y) - value of a pixel at position (x,y). Having a prediction of the motion vector and the distance until the ball stops, the vertices of the quadrangular *route* are calculated in the following way. Two vertices, P_1 and P_2, represent the intersection points between the circle of the ball and the line perpendicular to the motion vector passing through the center of the circle. The other two vertices, P_3 and P_4, are plotted on the line perpendicular to the motion vector passing through the end point of the trajectory $s_{predicted}$ at a distance ($radius$ + δ) in both directions from the $s_{predicted}$, where $radius$ - radius of the

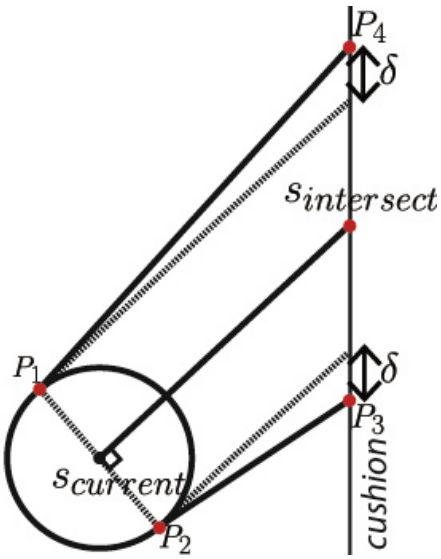


Fig. 1. Points estimation of quadrangular route in case of cushion collision

ball, δ - parameter that copes with small deviations of the ball's positions (see Figure 1). There are two cases for computing the values of P_3 and P_4 that should be distinguished. First, when the distance to the end point $s_{predicted}$ is smaller than the distance to the cushion. Second, when the distance to the end point $s_{predicted}$ is greater than the distance to the cushion. In the latter case, the intersection point between the trajectory and the cushion should be preliminary measured.

For the purpose of detecting, whether the obstacle-ball is inside the *route*, we multiply the binary masks of this ball and the *route*. In case the result is positive, we assume that there will be a collision and predict the time slot when it will happen. If there exist several obstacle-balls on the *route*, those which will be hit first is taken into account.

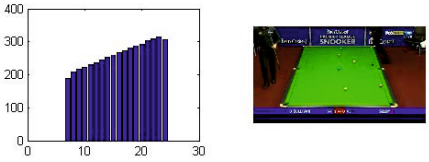


Fig. 2. Preliminary table detection based on RGB histogram approach

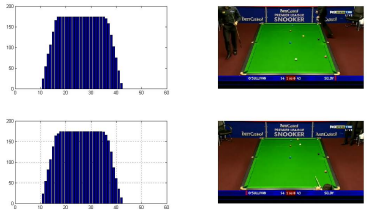


Fig. 3. Comparison of the first successful table view histogram with the current video frame histogram

Space Limits. In snooker broadcasts the effect of presence and involvement in a game is created via multiple camera views. In this paper we are particularly interested in a full-table view from the top (see Figure 4). The reason is that it has sufficient information for predicting a trajectory of the moving object. As opposed to other camera positions, it provides clear representation of the scene and is not dependant on 3D information for correct time and location estimations. Identification of such full-table view shots and parameters of the table is a vital step in obtaining the structure of the scene.

Fig. 4. Finding the position of middle pockets as lying on the intersection of diagonals

With the purpose to obtain the parameters of the full-table view frames, we initially accomplish a histogram approach in HSV color space. First, color thresholding and morphological closing are applied, in order to get a binary image of the green areas of the shot. For the resultant image 8-neighbor-connected components are found and the largest of them is assumed to be a candidate for a perspective view of the table. According to the perspective projection, the amount of green color increases with approaching to

the bottom of the image. A candidate region is, finally, tested on satisfying this criterium.

When the first successful frame is detected, the corresponding candidate region is utilized to collect the information about the table – histogram, boundaries and pocket positions. The boundaries of the table are obtained by Hough transform as it finds the most prominent lines on the given binary image. After that the intersection points between the boundaries are assumed to be the corner table pockets. Two pockets in the center are estimated as the intersections between lateral boundaries of the table and a straight line parallel to the remaining boundaries through the intersection point of diagonals [3] (see Figure 4). For the upcoming video frames we manage a histogram comparison with the first frame. The above procedure is illustrated in Figures 2– 3. In case one of the pockets lies on the *route* of the moving ball, it is then further analysed for potting.

Prediction. This part of the paper is dedicated to the method of predicting the trajectory of the moving ball. It is assumed that the motion of the ball is limited to uniformly accelerated linear model. Having a track of a moving ball $S = \{s_0, s_1, .., s_k\} = \{(x_0, y_0), (x_1, y_1), .., (x_k, y_k)\}$, the relation between the parameters x and y is recovered using one-dimensional linear regression:

$$y_i = \alpha x_i + \beta + \epsilon_i, i = \overline{0, k}$$

$$(a, b) = \underset{\alpha,\beta}{\operatorname{argmin}} \sum_{i=0}^{k} (y_i - \alpha x_i - \beta - \epsilon_i)^2$$

where α, β - motion model parameters (angular coefficient, absolute term); ϵ_i-precision error; a,b - point estimates of α and β. We decided to restrict the motion to the linear model, though, the extension to non-linear model is possible.

The distance from the current position $s_{current} = (x_{current}, y_{current})$ with the velocity $v_{current}$ and acceleration a until the stop of the ball is computed using the physical equation for uniformly accelerated motion:

$$s_{predicted} = \frac{v_{end}^2 - v_{current}^2}{2a} = \frac{0 - v_{current}^2}{2a}$$

Analysis of the video frames which correspond to the range $s_{current}$ and $s_{predicted}$ is eliminated.

Experiments. Existing methods that aim at analysing snooker footage either do not provide the results for tracking, or only give a few details. In this situation a reasonable comparison of approaches is hardly achievable, and, thus, we can only provide the summarization of our results.

This approach was tested on 17 minute snooker footage that is equal to 24980 video frames. Hereof 15352 frames(61%) contain full-table view, 868 frames(3.5%) were neglected due to the absence of motion, 4083 frames(16%) were reduced using prediction. The analysed videos have a frame rate from 15 fps to 29 fps. Extreme conditions for the current system are the following. First, a high speed

of the target causes the loss of a track. Second, a small distance between the target and other objects of the scene makes it impossible to compute the future trajectory. We tested the quality of prediction by comparing to the the results of a Kalman filter. It is worth reminding that Kalman provides a prediction/correction of the current target position. In contrast, our approach predicts/corrects the future important positions of the target which are collisions with the cushion or other balls. In general case, when the trajectory is close to a straight line, the proposed approach enables faster analysis by neglecting at once in average 1-3 seconds of video (20-60 video frames) with an average deviation of 10 pixels (diameter of the ball is 8-12 pixels). The advantage of performing sequential tracking with Kalman filter can be shown on the example of non-linear trajectory (Figure 5). In Figure 6 it is shown that Kalman corrects the position step by step and follows the real path. As opposed to it, our system predicts the linear future position and loses the robustness of a track.

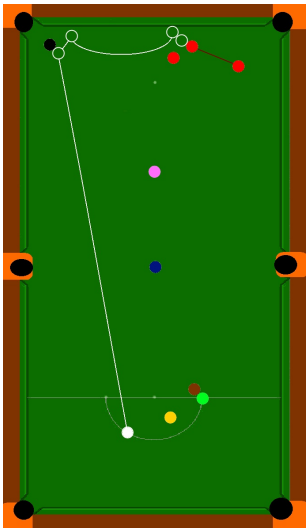


Fig. 5. The outline of the shot with non-linear motion

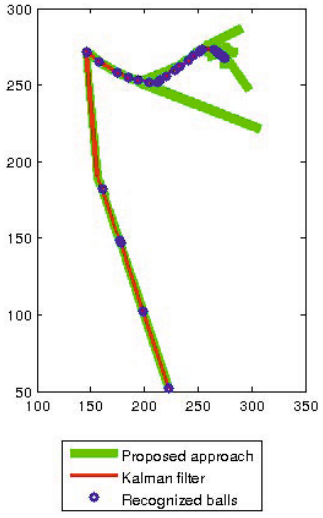


Fig. 6. The results of processing the motion of the ball with non-linear motion

4 Conclusion

This paper presented a framework for snooker video analysis. The future positions of the moving balls are predicted using physically-based linear motion model with respect to the structure of the scene. Motion model is characterized by notions of velocity, acceleration and previous states of the tracked object. Structure of the scene represents the obstacles and space limits that impact the trajectory and motion parameters of the target. In terms of snooker application they are cushion, pockets and balls other than the target. For the future work

we plan to research the rotational component of the ball-motion. This feature makes a valuable impact on motion model, as well as on the reflection angle while hitting the cushion or other balls.

References

1. Anderson, R.L.: A robot ping-pong player: experiment in real-time intelligent control. MIT Press (1988)
2. Bunke, H., Günter, S.: Towards bridging the gap between statistical and structural pattern recognition: Two new concepts in graph matching. In: Singh, S., Murshed, N., Kropatsch, W.G. (eds.) ICAPR 2001. LNCS, vol. 2013, pp. 1–11. Springer, Heidelberg (2001)
3. Denman, H., Rea, N., Kokaram, A.: Content-based analysis for video from snooker broadcasts. Computer Vision and Image Understanding 92(2), 176–195 (2003)
4. Horn, B., Schunck, B.: Determining optical flow. In: 1981 Technical Symposium East, pp. 319–331. International Society for Optics and Photonics (1981)
5. Huang, Y., Xu, D., Tan, M., Su, H.: Trajectory prediction of spinning ball for ping-pong player robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3434–3439. IEEE (2011)
6. Kalman, R.E.: A new approach to linear filtering and prediction problems. Journal of basic Engineering 82(1), 35–45 (1960)
7. Lee, D., Glueck, M., Khan, A., Fiume, E., Jackson, K.: A survey of modeling and simulation of skeletal muscle. ACM Transactions on Graphics 28(4), 1–13 (2010)
8. Legg, P.A., Parry, M.L., Chung, D.H., Jiang, R.M., Morris, A., Griffiths, I.W., Marshall, D., Chen, M.: Intelligent filtering by semantic importance for single-view 3d reconstruction from snooker video. In: 2011 18th IEEE International Conference on Image Processing (ICIP), pp. 2385–2388. IEEE (2011)
9. Lewis, J.: Fast template matching. In: Vision interface, vol. 95, pp. 15–19 (1995)
10. Maggio, E., Cavallaro, A.: Video tracking: theory and practice. John Wiley & Sons, Chichester (2011)
11. Rea, N., Dahyot, R., Kokaram, A.: Semantic event detection in sports through motion understanding. In: Enser, P.G.B., Kompatsiaris, Y., O'Connor, N.E., Smeaton, A.F., Smeulders, A.W.M. (eds.) CIVR 2004. LNCS, vol. 3115, pp. 88–97. Springer, Heidelberg (2004)
12. Resal, H.-A.: Commentaire à la théorie mathématique du jeu de billard. Nelineinaya Dinamika (Russian Journal of Nonlinear Dynamics) 6(2), 415–438 (2010)
13. Sisbot, E.A., Marin-Urias, L.F., Alami, R., Simeon, T.: A human aware mobile robot motion planner. IEEE Transactions on Robotics 23(5), 874–883 (2007)
14. Weiser, M., Zachow, S., Deuflhard, P.: Craniofacial surgery planning based on virtual patient models. it-Information Technology 52(5), 258–263 (2010)

Evaluating Classification Performance with only Positive and Unlabeled Samples

Siamak Hajizadeh, Zili Li, Rolf P.B.J. Dollevoet, and David M.J. Tax

Delft University of Technology
Stevinweg 1, 2628 CN Delft, The Netherlands
{S.Hajizadeh,Z.Li,R.P.B.J.Dollevoet,D.M.J.Tax}@TUDELFT.NL
http://www.tudelft.nl

Abstract. Testing binary classifiers usually requires a test set with labeled positive and negative examples. In many real-world applications however, some positive objects are manually labeled while negative objects are not labeled explicitly. For instance in the detection of defects in a large collection of objects, the most obvious defects are normally found with ease, while normal-looking objects may just be ignored. In this situation, datasets will consist of *only positive and unlabeled* samples. Here we propose a measure to estimate the performance of a classifier with test sets lacking *labeled* negative examples. Experiments are performed to show the effect of several criteria on the accuracy of our estimation, including that of the assumption of "random sampling of the labeled positives". We put the measure into use for classification of real-world defect detection data with no available validation sets.

Keywords: Classifier performance measure, unlabeled examples, binary classification.

1 Introduction

Unlabeled samples are often easily accessible in most artificial learning applications. They provide information about the structure of the data and combined with the labeled samples, they can help a learner to distinguish between different data clusters. A special situation arises when the labeled samples are only from one class, called the *target* class. In this case, data consists of a number of positively labeled samples and a large set of unlabeled samples with unidentified positives and negatives. Such datasets are commonly said to have only *positive and unlabeled* samples and abbreviated as PU datasets.

In the literature, several techniques are proposed for training classifiers with PU data. Testing classifiers on the other hand, is a task that requires fully labeled data. Performance measures normally provide a summary of the performance of a classifier on test data. Almost all of these measures include the 4 components of the *confusion matrix*; the True Positive, False Negative, True Negative, and False Positive rates. Baldi et al [1] argue that any performance measure depending on less than all four of these numbers, is missing a part of the information

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 233–242, 2014.

and is bound to be biased. In PU learning, and therefore by the absence of labeled negative samples, the False Positive and True Negative ratios cannot be calculated reliably. Presence of a fully labeled *validation set* comprising fair numbers of positive and negative samples, has thus been a necessity for the testing, and occasionally for parameter adjustments [17,15,28].

Naturally, providing a fully labeled validation set is expensive, in terms of cost and time effort. Elkan and Noto for example [11] explain how they had to use manually identified [8] samples from a larger unlabeled set to construct a validation set. In our case, identification of railway defects from a PU vibration data has been a motivation for this paper. The rail data has a relatively small number of positively labeled (defective) samples and a large set of unlabeled ones. Providing a validation set has proven to be too expensive due to high costs of physical defect detection tests. This has led us to the question: "Is it possible to robustly estimate the performance of a classifier, using only positive and unlabeled data?".

In [15] by Lee and Liu, a PU performance measure is proposed that is inspired by the F1 score. It equals to $p.r/\Pr[y=1]$, where p is the *precision*, r is the *recall*, and $\Pr[y=1]$ represents the prior probability of the positive class. Precision p and $\Pr[y=1]$ cannot be calculated from PU data directly. But Lee and Liu derive that $p.r/\Pr[y=1]$ equals to $r^2/\Pr[f(\mathbf{x})=1]$, with f being a trained classifier, allowing the measure to be calculated from only PU samples. Unfortunately though, it makes the implicit assumption [11] (originally noted in [10]) that the labeled object are *sampled randomly* from the positive class distribution. This means that each actually positive object has had an equal chance of being labeled. Satisfying this assumption can sometimes be difficult [3]. In the case of the railway defect identification for example, the ground-truth labeling is carried out by visual inspection of the rail, and therefore highly visible rail defects are more likely to be identified and labeled. Another similar measure proposed by B. Clavo et al [4] is called Pseudo-F and is calculated as $2\times r/(\Pr[f(\mathbf{x})=1]+\Pr[y=1])$. $\Pr[y=1]$ is assumed provided to both the classifier and the performance evaluator.

We propose a measure of performance for PU data (section 2) that is not dependent either on the assumption of random sampling or on $\Pr[y=1]$ in its definition. We call this measure PULP for Positive and Unlabeled Learning Performance. PULP calculates the probability for a classifier, to manage to randomly *correctly predict* a certain number of labeled samples as positives. Our proposition is that the less probable it is to accidentally detect a certain number of labeled positive examples, given the total number of positive predictions made, the better the predicting classifier has performed.

Techniques for training a classifier with PU data is not a subject of this paper. This has been studied in the literature of PU learning resulting several models tailored for exploiting information of the unlabeled samples. A number of approaches propose methods for identifying some potential negative samples from the unlabeled set and applying a normal binary classification algorithm [12,27,19]. A less common approach, is to assign weights to unlabeled

samples [11,20] that describe their likelihood of belonging to the positive class. Expectation-maximization algorithms are also a common family of algorithms for most notably: Biased SVM [17,18] and fitting logistic regressors [26]. PU learning techniques are widely applied to several learning domains such as: bioinformatics [11,26,5], geographic image processing [16,29], and document classification [12,19,17,15].

If the random sampling assumption is valid, one can also fairly use a conventional error measure (e.g. AUC) to estimate performance from a PU test set. We compare five conventional classification performance measures from [1] that are widely used in testing classifiers. These are the AUC, F1 score, Mean Average Precision (MAP), Pearson correlation (PEAR), and the mutual information (MI) measure. We also test the proposed PU measures by Lee and Liu (referred to by L&L) and Pseudo-F in our comparison. To examine the applicability of all these measures to PU data, we perform several experiments. We first define this applicability in terms of accuracy in section 3. We then show by experiments in section 4 that PULP is affected the least when the random sampling assumption is violated, making it a better choice in such cases as the rail defect detection problem. We also investigate the effect of the prior distribution probability of the unlabeled positive samples, on the accuracy of the tested measures. We conclude increasing positive class prior probability translates to decreasing estimation accuracy, for all of the studied measures.

2 PULP – Positive and Unlabeled Learning Performance

PULP calculates the probability of randomly making a number of positive predictions, and yet managing to *hit* some of the positively labeled samples. Assume that from a set of N samples, t samples are labeled as positive and the rest are given unlabeled. Also assume that a classifier has predicted b out of the total N samples as positive, and such that k of those predictions match the t labeled samples. Clearly, k satisfies $k \leq \min(t, b)$. We are interested in the probability of hitting *at least* k of the t known positive samples; doing so merely by random assignment of b positive tags to the N samples. We are keen to know how easy it is to score at least as good as this classifier, just by random predictions. To get that probability, we should first note that the probability of randomly hitting *exactly* k of t labeled samples is equal to the *hypergeometric* probability mass function f at point k, with parameters N, t, and b being respectively the population size, number of successes, and number of draws:

$$f(k \mid N, t, b) = \frac{\binom{t}{k}\binom{N-t}{b-k}}{\binom{N}{b}}. \tag{1}$$

The denominator simply equals all permutations of assigning b tags to N samples. The numerator of the fraction is all permutations of such assignment where exactly k hits are made. To get the probability of hitting at least k labeled samples, it suffices to integrate $f(i \mid N, t, b)$ over all values i where $k \leq i \leq b$. This probability $\overline{F}(k \mid N, t, b)$ is the cumulative hypergeometric function F,

which makes PULP closely related to Fisher's exact probability test (see e.g. [22]).

$$\begin{aligned}\overline{F}(k \mid N,t,b) = \sum_{i=k}^{b} f(i \mid N,t,b) &= 1 - \sum_{i=0}^{k-1} f(i \mid N,t,b) \\ &= 1 - F(k-1 \mid N,t,b)\end{aligned} \tag{2}$$

If b is larger than t in the first sum, then f will automatically yield zero, meaning that in the summation i needs not to go any further than $\min(t,b)$. Lower $\overline{F}$ values mean that it is less likely to perform at least as good as that classifier just by random assignment, and indicate that the classifier has more *information* about the positive class distribution. By taking $1 - \overline{F}(k \mid N,t,b) = F(k-1 \mid N,t,b)$ as the measure, this relation becomes straight so that higher values now indicate better performances.

In standard testing and error estimation, it is a common practice to average over all threshold values of a classifier decision boundary to summarize its performance. This avoids dependence on a specific value for the operating point. The Area Under the ROC Curve (AUC), and the Mean Average Precision (MAP) are such summarizations. Applying the same rationale, PULP is also the average over all probabilities $\overline{F}(k \mid N,t,b)$ obtained by integrating over all operating points of the classifier while recalculating k each time. Let $S : \{s_1, s_2, ..., s_N\}$ be the sorted list of classified objects according to the output of the tested classifier. Then $\mathrm{PULP}(S \mid N,t)$ can be evaluated as:

$$\mathrm{PULP}(S \mid N,t) = \frac{1}{N+1} \sum_{i=0}^{N} F(k_{\{S,i\}} - 1 \mid N,t,i) \tag{3}$$

where $k_{\{R,i\}}$ is the number of hits given that objects s_1 to s_i are predicted to be positives. Here it is assumed that PULP is calculated for a binary classification test with one target class. In the literature, this condition is usually referred to as one-class classification [9]. While it is possible to extend PULP for application to mutli-class classification tests, here we only present PULP as a one-class performance estimator for PU test data.

In practice, evaluating $f(k \mid N,t,b)$ for moderately large test set sizes can become problematic due to large factorial computations. We have used the workaround to rearrange combination terms to exponential function of natural logarithms as:

$$\begin{aligned}f(k \mid N,t,b) = \exp[&\ln t! + \ln(N-t)! + \ln b! + \ln(N-b)! \\ &- \ln k! - \ln(t-k)! - \ln(N-t-b+k)! - \ln(b-k)! - \ln N!].\end{aligned} \tag{4}$$

Doing so, it will be possible to take advantage of natural logarithm of the Gamma function which is approximated with a number of available techniques, including Chebyshev approximation [7]. One other option is to approximate hypergeometric mass function by other distributions such as Poisson [13].

3 The Evaluation of a Performance Measure

Defining an "accurate" measure is an intricate matter. For the analysis and evaluation of a PU performance measure, one can artificially create PU classification problems from standard two-class classification problems. In the standard binary classification, each object $\mathbf{x}$ has a true label y that is positive or negative [2]. In the derived PU problem only part of the positive class is labeled and the rest of the samples are left unlabeled. To compare the applicability of the performance measures to PU data we test them on both PU data and fully-labeled data and check for the similarity of the two sets of results.

To quantify this similarity, we use the Mean Absolute Deviation (MAD) and the rank (Spearman) correlation. The mean absolute deviation between the two sets, is aimed at assessing the difference between fully-labeled and PU results on average. Although this is informative, when a performance measure is used to compare and select the best (PU) classifier, a correct ranking of the performances can be even more important. A high correlation between the PU measure and their fully-labeled pair indicates the suitability of a measure for evaluating classifiers based on PU data. To be able to make a fair comparison, the rest of the measures (i.e. F1, MI, PEAR, L&L, and Pseudo-F) are also averaged over all classifier operating point similar to the way we calculate PULP. For calculation of Pseudo-F we simply take the ratio of the positively labeled to all samples as the prior probability of the positive class.

The prior probability that a sample is positive $Pr[y = 1]$, has a strong influence on a PU evaluation measure. We aim to choose measures that are least sensitive to $Pr[y = 1]$. Therefore, all experiments are performed over a increasing range of the ratio of the actual positive objects in the unlabeled subset. We also investigate the effect of the assumption that the labeled samples are drawn randomly from the positive class. Experiments are performed that compare the measures on 3 scenarios, where the labeled samples are either selected at random, or selected so that the most *representative* ([24]) or the least representative positives are labeled. These three situations are shown in Figure 1. First, from the fully labeled data (Figure 1(a)) a random subset of the positive objects is labeled, while the rest is unlabeled (Figure 1(b)). Next, a Gaussian mixture model is trained on the whole class of the actual positives, and its output posterior probabilities are used as indicators of representativeness. In Figure 1(c) the positive samples that are less representative are left unlabeled. While in Figure 1(d) those which are considered to be more representative are left unlabeled. Our experiments will investigate the sensitivity of PU performance measures to these non-random sampling strategies.

Three datasets are selected from the literature of PU learning for experiments. The first is the popular 20 newsgroup dataset which consists of 20 news categories where each have 1000 sample documents. Similar to the procedure in [15] we extract bag-of-words features and we apply TF-IDF normalization. The second is the MNIST dataset of hand-written digits [14]. It includes 1000 samples per each digit 0 to 9. Here, we simply take the raw pixels as the features and extract a 16×16 down-sampled version of the original images. Finally, we test an originally

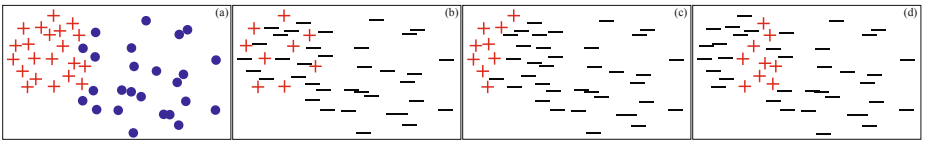


Fig. 1. Diagram (a) shows the *complete labeling* [2] of positive and a negative objects. Diagram (b) shows a PU labeling where selection of the labeled objects is random, while the negative object are always unlabeled. In (c) the selection is made in a way that the least representative positives are left unlabeled. On the contrary, in (d) the most representative ones are unlabeled.

PU labeled dataset by [11]. It contains 7359 text records of protein specifications. These records contain 2453 positive records from a special set of proteins [23] that are known membrane transport proteins. The dataset also contains 4906 unlabeled protein records from [25] that are manually identified by [8] to have 348 positive records and 4558 negative ones. A compilation of this data is made available by [11]. We extract bag-of-words features from these records and apply TF-IDF normalization as well.

In accord with a generalization in [11] of a several PU learning techniques, we first calculate weights for all unlabeled samples. The calculation of weights is either done by heuristics mostly inspired by [18] or a weight assigning technique in [11]. These weights are regarded as the likelihood that an unlabeled sample belongs to the positive class. An SVM classifier [6] is trained on the weighted datasets.

4 Experiments and Results

From experiments on individual classifiers, we hypothesized that $Pr[y = 1]$ can strongly influence accuracy of a measure. To examine this, we have calculated the mean absolute deviation and the rank correlation for a broad range of training and testing criteria. In total, 260 different configurations of datasets and classifiers were tested while at first, the labeled positives were selected randomly from all positive samples. The results are show in the left column of Figure 2 for the correlation in upper row and for the mean absolute deviation in the lower row.

We also performed 2 rounds of similar experiments while labeling positives according to the 2 non-random scenarios discussed in section 3. The middle and right column of Figure 2 show these results where respectively the least and the most representative positives are left unlabeled. Our first observation is that while the rank correlations stay close for all measures in the random labeling case, they tend to be dispersed more in the non-random cases. PULP is affected the least by the non-random sampling. This suggests it can be a robust choice compared to L&L and Pseudo-F if the random sampling assumption does not hold. In such cases, PULP is also a better choice compared to the non-PU measures of performance that as we anticipated, are affected more by the non-random sampling scenarios. It is evident that all measures lose estimation accuracy as $Pr[y = 1]$ increases over the unlabeled data samples. However for

all $Pr[y = 1]$ values below 0.5, PULP scores a rank correlation that is above 90 percent regardless of the sampling scenario.

Mean absolute deviation is also a helpful accuracy evaluation for a PU performance measure. The Mutual Information measure (MI) seems to have the least mean absolute deviation for the two non-random labeling scenarios. This is slightly difficult to interpret though, because the standard deviation calculated over all MI outcomes is also the smallest (Table 1). This suggests that the range of values produced by MI is narrower than the rest of the measures. Therefore, interpretation of accuracy based on the mean absolute deviation can only be meaningful in combination with the total Standard Deviation of its evaluations. This helps avoid choosing measures that score a small mean absolute deviation only because they have a limited or biased function range.

By taking an average over all results of different values $Pr[y = 1]$, Table 1 gives a summary of all evaluations of rank correlation and mean absolute deviation for the 8 measures. Under the experiment setting in this paper, we believe the PU measure introduced by Lee and Liu (L&L) is a more accurate PU estimate than Pseudo-F, in both random and non-random labeling cases. This is visible in the rank correlation evaluations, as well as the mean absolute deviations where the deviation for L&L is always less despite having a higher overall standard deviation. PULP in our opinion is a better choice that both of these measures in cases that the "selected completely at random" assumption might be invalid.

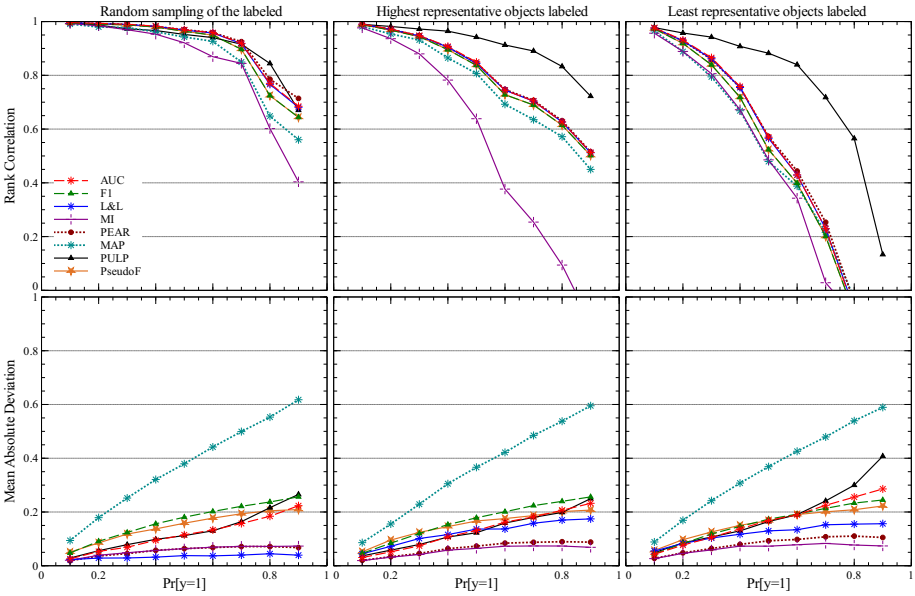


Fig. 2. The rank correlations (upper row) and mean absolute deviations (lower row) between the fully labeled data and the PU data for the seven performance measures on 260 combination of datasets and classifiers. The 3 columns are corresponding to random selection and two non-random selection scenarios for partially labeling positives samples.

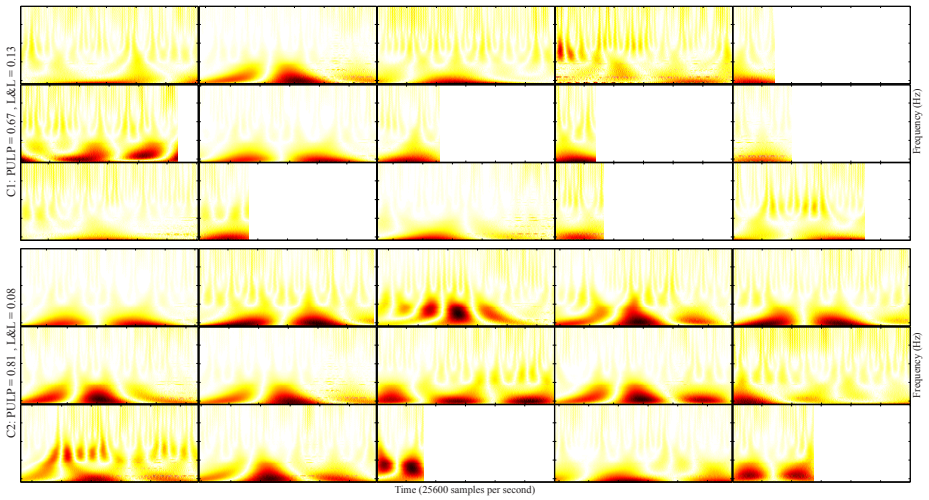


Fig. 3. The first 3 rows show the top positive samples according to classifier C1 selected by a higher L&L result. The second 3 rows are the same for classifier C2 selected by a higher PULP result. C2 has managed to find more unlabeled potential positives. Each sample is the wavelet spectrogram for a window of the signal. Vertical axes are frequency scales. Horizontal axes are time. In the domain of rail defect detection, high frequency excitations are usually associated with potential defects.

Table 1. Results from Figure 2 averaged over all measurements of individual $p(y = 1)$ values. Table presents this total results for the same 3 scenarios for selection of labeled positive samples. Bold numbers are best scores in each row. All results are averaged over a 8-fold cross-validation.

		AUC	F1	L&L	MI	PEAR	MAP	PULP	PseudoF
Rank Correlation	Random	0.918	0.903	0.916	0.837	**0.923**	0.870	0.916	0.903
	Scenario 1	0.806	0.796	0.804	0.535	0.806	0.765	**0.912**	0.796
	Scenario 2	0.495	0.474	0.494	0.422	0.504	0.465	**0.769**	0.474
Mean Absolute Deviation	Random	0.117	0.168	**0.034**	0.055	0.057	0.370	0.127	0.148
	Scenario 1	0.130	0.166	0.123	**0.056**	0.065	0.353	0.132	0.150
	Scenario 2	0.165	0.160	0.120	**0.065**	0.0814	0.356	0.185	0.157
Overall Standard Deviation		0.197	0.099	0.248	0.061	0.100	0.147	0.324	0.143

Rail vibration data consists of 10 datasets of various sizes where features are wavelet power spectrogram values and objects are frames from the vibration signal. Using 6 out of 10 datasets in a train and test cross validation, we selected two SVM classifiers: C1 and C2 that have disagreeing PULP and L&L results, such that their PULP results are: 0.67, and 0.81 and L&L results are: 0.13 and 0.08 correspondingly.

We then tested these classifier on another test set of 139 samples with 23 labeled positives. For both classifiers, we sorted the 139 samples in descending

order of their output and selected highest 15 top scored after excluding the already labeled. These top 15 object for both classifiers are illustrated in Figure 3. The first 3 rows are the top positive samples according to C1, the second 3 rows are the same for C2. In at least 10 out of 15 unlabeled samples selected by C2, there are traces of high frequency excitations that can indicate defects on the rail [21]. For C1 this is down to around 5 samples. We conclude that PULP has probably made a better choice between the two classifiers.

5 Conclusions

We proposed PULP: a measure PULP to evaluate the learning performance of a classifier with only positive and unlabeled data. Besides a well-defined theory, our experiments show that PULP has the advantage over the rest of the tested measures, that it is affected the least by a non-random sampling of the labeled positives. We test PULP and other performance measures, on a number of datasets from the literature in these experiments. We have as well used PULP to compare classifiers that are trained and tested on a real-world PU dataset consisting of rail vibration signal, to detect rail defects. We were able to visually confirm that a stronger classifier according to PULP, detects more potential positives compared to a weaker PULP classifier.

Acknowledgements. This research is part of ExploRail project funded by ProRail - Dutch rail infrastructure manager, and the Netherlands organisation for scientific research (STW/NWO). We used advice from Dr. Keith Noto on protein dataset feature extraction.

References

1. Baldi, P., Brunak, S., Chauvin, Y., Andersen, C.A., Nielsen, H.: Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics 16(5), 412–424 (2000)
2. Bishop, C.M., et al.: Pattern recognition and machine learning, vol. 1. Springer, New York (2006)
3. Blanchard, G., Lee, G., Scott, C.: Semi-supervised novelty detection. The Journal of Machine Learning Research 11, 2973–3009 (2010)
4. Calvo, B., Inza, I., Larrañaga, P., Lozano, J.A.: Wrapper positive bayesian network classifiers. Knowledge and Information Systems 33(3), 631–654 (2012)
5. Cerulo, L., Elkan, C., Ceccarelli, M.: Learning gene regulatory networks from only positive and unlabeled data. Bmc Bioinformatics 11(1), 228 (2010)
6. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2(3), 1–27 (2011)
7. Cody, W.J., Hillstrom, K.: Chebyshev approximations for the natural logarithm of the gamma function. Mathematics of Computation 21(98), 198–203 (1967)
8. Das, S., Saier Jr., M.H., Elkan, C.: Finding transport proteins in a general protein database. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 54–66. Springer, Heidelberg (2007)

9. David, M.: Tax. one-class classification; concept-learning in the absence of counter-examples. ASCI Dissertation Series 65 (2001)
10. Denis, F.: PAC learning from positive statistical queries. In: Richter, M.M., Smith, C.H., Wiehagen, R., Zeugmann, T. (eds.) ALT 1998. LNCS (LNAI), vol. 1501, pp. 112–126. Springer, Heidelberg (1998)
11. Elkan, C., Noto, K.: Learning classifiers from only positive and unlabeled data. In: The 14th ACM SIGKDD International Conference, pp. 213–220 (2008)
12. Fung, G.P.C., Yu, J.X., Lu, H., Yu, P.S.: Text classification without negative examples revisit. IEEE Transactions on Knowledge and Data Engineering 18(1), 6–20 (2006)
13. Harkness, W.L.: Properties of the extended hypergeometric distribution. The Annals of Mathematical Statistics, 938–945 (1965)
14. LeCun, Y., Cortes, C.: The mnist database of handwritten digits (1998)
15. Lee, W.S., Liu, B.: Learning with positive and unlabeled examples using weighted logistic regression. In: ICML, vol. 3, pp. 448–455 (2003)
16. Li, W., Guo, Q., Elkan, C.: A positive and unlabeled learning algorithm for one-class classification of remote-sensing data. IEEE Transactions on Geoscience and Remote Sensing 49(2), 717–725 (2011)
17. Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S.: Building text classifiers using positive and unlabeled examples. In: Third IEEE International Conference on Data Mining, ICDM 2003, pp. 179–186. IEEE (2003)
18. Liu, B., Lee, W.S., Yu, P.S., Li, X.: Partially supervised classification of text documents. In: ICML, vol. 2, pp. 387–394. Citeseer (2002)
19. Liu, B., Li, X., Lee, W.S., Yu, P.S.: Text classification by labeling words. In: AAAI, vol. 4, pp. 425–430 (2004)
20. Liu, Z., Shi, W., Li, D., Qin, Q.: Partially supervised classification – based on weighted unlabeled samples support vector machine. In: Li, X., Wang, S., Dong, Z.Y. (eds.) ADMA 2005. LNCS (LNAI), vol. 3584, pp. 118–129. Springer, Heidelberg (2005)
21. Molodova, M., Li, Z., Núñez, A., Dollevoet, R.: Automatic detection of squats in railway infrastructure. IEEE Intelligent Transportation Systems (2014)
22. Rivals, I., Personnaz, L., Taing, L., Potier, M.C.: Enrichment or depletion of a go category within a class of genes: which test? Bioinformatics 23(4), 401–407 (2007)
23. Saier, M.H., Tran, C.V., Barabote, R.D.: Tcdb: the transporter classification database for membrane transport protein analyses and information. Nucleic acids research 34(suppl 1), D181–D186 (2006)
24. Tenenbaum, J.B., Griffiths, T.L., et al.: The rational basis of representativeness. In: Proceedings of the 23rd Annual Conference of the Cognitive Science Society, pp. 1036–1041. Citeseer (2001)
25. UniProt: Consortium et al.: Activities at the universal protein resource (uniprot). Nucleic Acids Research 42(D1), D191–D198 (2014)
26. Ward, G., Hastie, T., Barry, S., Elith, J., Leathwick, J.R.: Presence-only data and the em algorithm. Biometrics 65(2), 554–563 (2009)
27. Yu, H., Han, J., Chang, K.C.: Pebl: Web page classification without negative examples. IEEE Transactions on Knowledge and Data Engineering 16(1), 70–81 (2004)
28. Zhang, D., Lee, W.S.: A simple probabilistic approach to learning from positive and unlabeled examples. In: Proceedings of the 5th Annual UK Workshop on Computational Intelligence (UKCI), pp. 83–87. Citeseer (2005)
29. Zhu, C., Liu, B., Yu, Q., Liu, X., Yu, W.: A spy positive and unlabeled learning classifier and its application in hr sar image scene interpretation. In: 2012 IEEE Radar Conference (RADAR), pp. 0516–0521. IEEE (2012)

Who Is Missing? A New Pattern Recognition Puzzle

Ludmila I. Kuncheva and Aaron S. Jackson

School of Computer Science,
Bangor University, United Kingdom

Abstract. Consider a multi-class classification problem. Given is a set of objects, for which it is known that there is at most one object from each class. The problem is to identify the missing classes. We propose to apply the Hungarian assignment algorithm to the logarithms of the estimated posterior probabilities for the given objects. Each object is thereby assigned to a class. The unassigned classes are returned as the solution. Its quality is measured by a consistency index between the solution and the set of known missing classes. The Hungarian algorithm was found to be better than the rival greedy algorithm on two data sets: the UCI letter data set and a bespoke image data set for recognising scenes with LEGO parts. Both algorithms outperformed a classifier which treats the objects as iid.

Keywords: Pattern recognition, set classification, Bayes-optimal classifier, Hungarian algorithm.

1 Introduction

Who is missing? At your lecture, you are taking a class register from a single snapshot of the audience. If the number of students in the audience is the same as the size of your class, and you are satisfied that there are no impostors, then all your program needs to do is count the faces in the snapshot. However, if the number of attendees is smaller than the class list, you will need to find out *who* the missing students are.

Suppose that you have a trained classifier to recognise the students' faces. If the face detection program and the classifier were ideal, all faces would be correctly detected and recognised, hence the missing students will be identified instantly. However, if the classifier is imperfect, classifying each face individually may not be the optimal strategy. First, individual labelling will not prevent assigning the same label to several objects. Second, individual labelling cannot take into account any class dependencies. For example, suppose that students X and Y are friends, and are always present or absent together. Individual labelling will not be able to take advantage of this piece of knowledge. Therefore, some form of set classification would be a more prudent strategy.

One of the standard assumptions in classical pattern recognition is that the data points to be classified come as an independent identically distributed (iid)

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 243–252, 2014.

sequence. In many problems this assumption does not hold. For examples of non-iid classification paradigms are listed below.

1. *The multiple-instance Problem.* This problem arises in complex machine learning applications where the information about the instances is incomplete or ambiguous [4, 7, 13, 19], e.g., in drug activity prediction [4]. The training examples come in "bags" labelled either positive or negative. For a positive bag, it is known that at least one instance in the bag has true positive label. For a bag labelled negative, all instances are known to be negative. The problem is to design a classifier that can label as accurately as possible an unseen bag of instances.
2. *Set Classification.* In this problem, all the instances in a set are assumed to have come from the same unknown class [16]. This problem may arise in face recognition where multiple images of the same person's face are submitted as a set.
3. *Collective Recognition.* In this scenario, a set of instances are labelled together [14, 18]. The crucial assumption is that the instances within the set are related, so that the dependencies can be used to improve the classification accuracy. For examples, in classifying web pages into topic categories, hyperlinked web pages are more likely to share common class labels than non-linked pages [18].
4. *Full-Class Set Classification* [11]. Here a set of instances has to be classified together, knowing that the set contains at most one instance from each class. In other words, the c objects must be matched one-to-one to the c classes. We can call this problem 'who-is-who' to distinguish it from the 'who-is-missing' problem. Simultaneous classification of a set of instances has been used in tracking. For example, a moving object can be regarded as a patchwork of parts [1] or a set of tracklets [10], which are matched from one image frame to the next. This fits within the framework considered here because each part/tracklet on the object can be referred to as a class label, and the segmented pieces in the image have to be distributed to the different class labels. However, the instances within the set are not iid, as the parts are spatially linked within the object, and also follow physical motion laws. Other potential applications include karyotyping (labelling the chromosomes in a cell) [9,15] and identifying footballers on the pitch in a live-play video [3].

The who-is-missing problem is a variant of paradigm #4, the full set classification [11]. In this study we formulate the *who-is-missing* problem and propose a solution based on the Hungarian assignment algorithm.

2 The Who-Is-Missing Problem

2.1 Problem Description

Let $\Omega = \{\omega_1, \ldots, \omega_c\}$ be the set of class labels. The set of objects presented for classification is $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$, where $k < c$, and the true labels of the objects

in $\mathbf{Z}$, denoted $\{y_1, \ldots, y_k\}$ are all different. The task is to find the set of missing classes, that is the set

$$\Omega_{(-)} = \Omega \setminus \Omega_{(+)}, \tag{1}$$

where

$$\Omega_{(+)} = \bigcup_{i=1}^{k} y_i. \tag{2}$$

Because of the strict inequality $k < c$, $\Omega_{(-)}$ is a non-empty set.

Denote by $P(\omega_j|\mathbf{x})$ the probability that the true class label for an observed $\mathbf{x}$ is $\omega_j \in \Omega$. We can arrange the posterior probabilities in a $k \times c$ matrix

$$P = \begin{bmatrix} P(\omega_1|\mathbf{z}_1) & \ldots & P(\omega_c|\mathbf{z}_1) \\ \vdots & \vdots & \vdots \\ P(\omega_1|\mathbf{z}_k) & \ldots & P(\omega_c|\mathbf{z}_k) \end{bmatrix}. \tag{3}$$

The task is to determine the $c-k$ missing classes based on P and the knowledge that labels $y_1, \ldots, y_k$ are different. The probability that the class label for a given $\mathbf{x}$ is *not* ω_j is $1 - P(\omega_j|\mathbf{x})$. If $\mathbf{Z}$ contained k iid objects, the probability that class ω_i is not represented in $\mathbf{Z}$ would be

$$P_{\text{iid}}(\sim \omega_i|\mathbf{Z}) = \prod_{j=1}^{k} (1 - P(\omega_i|\mathbf{z}_j)) \ . \tag{4}$$

The $c-k$ classes with the largest P_{iid} should be returned as $\Omega_{(-)}$. This approach, however, is based on the false iid assumption and hence does not take advantage of the fact that the elements of $\mathbf{Z}$ have different class labels.

Accurate identification of the missing classes is equivalent to accurate assignment of the present classes. Therefore the solution can be found using the Hungarian assignment algorithm[1]. Proposed originally for $c \times c$ matrices, the algorithm was extended for rectangular matrices [2].

It has been shown [11] that the Bayes-optimal solution of the who-is-who problem ($k = c$) is the permutation of labels $\langle s_1, s_2, \ldots s_c \rangle$ which maximises the criterion

$$\sum_{i=1}^{c} \log P(\omega_{s_i}|\mathbf{z}_i) \ . \tag{5}$$

The underlying assumption is that the object from each class is picked independently of the objects from the other classes.

2.2 An Example

As an example, consider three objects, $\mathbf{z}_1$, $\mathbf{z}_2$ and $\mathbf{z}_3$, coming from three classes, ω_1, ω_2 and ω_3. The class assignment of the three objects is not known, apart from the fact that there is one object from each class. Let P be

[1] Further developed by Kuhn and Munkres, also known as Kuhn-Munkres algorithm.

$$P = \begin{bmatrix} 0.65 & 0.07 & 0.28 \\ 0.43 & 0.50 & 0.07 \\ 0.24 & 0.57 & 0.19 \end{bmatrix} . \quad (6)$$

Table 1 shows all possible label permutations, the corresponding posterior probabilities and the sum-log criterion value (5) for each permutation. In addition, the sum of the posterior probabilities is also shown.

Table 1. An example of the class assignment problem

Class			Posteriors				
			$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$	$\sum \log()$	sum
3	2	1	0.28	0.50	0.24	-3.3932	1.02
3	1	2	0.28	0.43	0.57	-2.6791	1.28
2	3	1	0.07	0.07	0.24	-6.7456	0.38
2	1	3	0.07	0.43	0.19	-5.1640	0.69
1	2	3	0.65	0.50	0.19	-2.7847	1.34
1	3	2	0.65	0.07	0.57	-3.6522	1.29

According to the table, the best solution is $\mathbf{z}_1 \in \omega_3$, $\mathbf{z}_2 \in \omega_1$ and $\mathbf{z}_3 \in \omega_2$. Interestingly, this is not the solution which maximises the sum of the posterior probabilities.

A greedy approach would assign class ω_1 to $\mathbf{z}_1$ (0.65), next assign class ω_2 to $\mathbf{z}_3$ (0.57), and finally assign class ω_3 to the remaining object $\mathbf{z}_1$ (0.07). This permutation, (1,3,2), is ranked 4th of 6 on the sum-log criterion.

2.3 Proposed Solution

We propose to use the Hungarian assignment algorithm to a full $c-by-c$ matrix where $c-k$ objects will be "dummy" objects. Their respective rows with posterior probabilities are filled with values $\frac{1}{c}$, indicating a complete lack of preference of a class label. The class labels assigned by the algorithm to the dummy objects will be the missing classes.

The hypothesis is that the Hungarian algorithm will provide better solution to the who-is-missing problem compared to a greedy algorithm or independent classification that assumes iid data.

2.4 Evaluation of the Solution

To find out how successful the proposed strategy is, we need a measure of match between the true missing classes and the obtained missing classes. Simple measures based on the intersection between the two sets will not be adequate because such measures will depend on the number of the missing classes and are not corrected for chance. Therefore, we propose to use a *consistency index* [12].

The Consistency Index $I_C(A, B)$ for two subsets $A \subset X$ and $B \subset X$, such that $|A| = |B| = k$, where $0 < k < |X| = n$, is defined as

$$I_C(A, B) = \frac{r - \frac{k^2}{n}}{k - \frac{k^2}{n}} = \frac{rn - k^2}{k(n - k)}. \tag{7}$$

where $r = |A \cap B|$. The maximum value of the index, $I_C(A, B) = 1$, is achieved when $r = k$. The minimum value of the index is bound from below by -1. The limit value is attained for $k = \frac{n}{2}$ and $r = 0$. Note that $I_C(A, B)$ is not defined for the trivial cases $k = 0$ and $k = n$. They are not interesting from the point of view of comparing subsets, so the lack of values for $I_C(A, B)$ in these cases is not important. Finally, $I_C(A, B)$ will assume values close to zero for independently drawn A and B.

The hypothesis will be supported if the consistency index for the results from the Hungarian algorithm is higher than the indices for the rival methods.

3 Experiments

The purpose of this experiment is rather a proof of concept than comparison of possible alternatives. Both the Hungarian and the greedy algorithms were applied with criterion (5).[2]

Suitable data sets for the who-is-missing problem should have a large number of classes.

3.1 Letter Data Set

We chose the Letter data set from the UCI Machine Learning Repository [5]. The number of classes is 26 (letters from the Latin alphabet), and the number of objects is 20,000. The experimental protocol was as follows.

1. The data set was first standardised, and subsequently divided into a training part (the first 10,000 objects) and a testing part (the latter 10,000 objects).
2. A linear discriminant classifier was trained on the training part.[3] This classifier was chosen on purpose so that there is sufficient scope for improvement. Both the training and the testing errors are approximately 30%.
3. A level of noise η was chosen from the set $\{0.0, 0.1, 0.2, \ldots, 0.8\}$. Gaussian noise with mean zero and standard deviation η was added independently to each value in the testing data set. The perturbed testing data was classified using the classifier trained on the original training data. The posterior probabilities for all objects were stored.
4. The number of present classes k was chosen from the set $\{2, 3, \ldots, 25\}$.

[2] Since the logarithm is a monotonic transformation, the greedy algorithm would give exactly the same result if applied straight on the posterior probabilities.

[3] We used the `classify` function from the Statistics Toolbox of MATLAB.

5. 1000 runs were carried out with the chosen η and k. In each run, a random subset of k classes was sampled. One object was picked randomly from each of the present classes. The posterior probabilities for the selected objects (given by the classifier in Step 3) were retrieved and collated in matrix $\hat{P}$. The matrix was augmented to 26×26 by adding $26 - k$ dummy rows with values $\frac{1}{26}$. The Hungarian and the Greedy algorithms were applied and the respective sets of missing classes (assigned to the dummy rows) were recorded. Let HM be the set of missing classes according to the Hungarian algorithm, GM, for the Greedy algorithm, and TM be the TRUE set of missing classes. We applied the original classifier assuming iid data. Let CM be the set of non-assigned classes. The respective values of the consistency index were calculated as

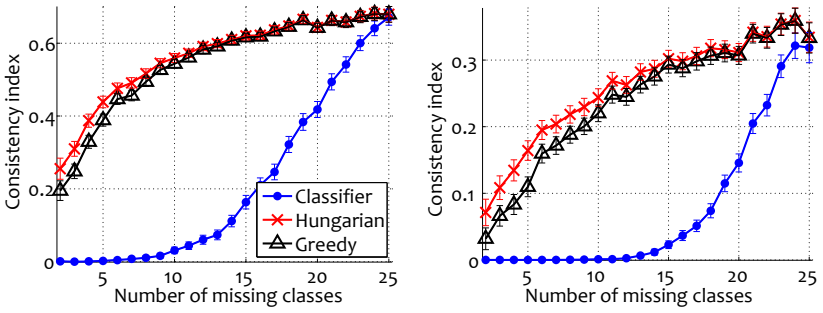
$$I_{\text{Hungarian}}(k,\eta) = I_C(TM, HM), \quad I_{\text{Greedy}}(k,\eta) = I_C(TM, GM)$$

and

$$I_{\text{Classifier}}(k,\eta) = \begin{cases} I_C(TM, CM), & \text{if } |CM| = |TM|, \\ 0, & \text{otherwise.} \end{cases}$$

The values of the consistency indices, averaged across the 1000 runs, for noise levels $\eta = 0$ and $\eta = 0.8$, are shown in Figure 1. The graphs for the remaining noise levels followed similar patterns. Plotted are also error bars spanning the 95% confidence intervals calculated from the 1000 values of the respective run.

The upward trend of the curves can be explained with the following argument. When only a few objects are missing, their correct labelling depends on the correct labelling of all the remaining objects. The scope for error is high. On the other hand, when only a few classes are present, there is less room for error in classifying all these objects correctly. Finally, when there is only one object present ($c - 1$ absent), all methods converge to the original classifier.



(a) No noise (b) Noise 0.8

Fig. 1. Consistency indices for the sets of missing classes using the three algorithms on the Letter data set. The error bars indicate the 95% confidence intervals.

Clearly, the classifier alone is a poor choice according to the chosen measure. The Hungarian and the Greedy algorithms behave similarly but the Hungarian algorithm had an edge, giving support to our hypothesis. To demonstrate this finding, Figure 2 shows the results from a statistical test between the results of the two methods. We carried out a paired, two-sided test of the hypothesis that the difference between the matched 1000 samples of consistency indices comes from a distribution whose median is zero (Wilcoxon signed rank test). We chose to show the results as a heat map. The grey level intensity corresponds to the p-value. White indicates $p = 1$ and black, $p = 0$. Each square is the result from one comparison across the 1000 iterations for the respective number of missing classes $c - k$ and noise level η. The comparisons where there was significant difference at $\alpha = 0.5$ are marked with dots. As the Hungarian method was always superior or equivalent to the Greedy method, the dots mark the combinations of parameters where the Hungarian method wins.

As expected, larger noise level showcase the proposed method. This is shown by the larger number of dots in the top rows. For noise-free data (bottom row), the algorithms tie for a smaller number of missing classes. For large number of missing classes (rightmost column), the algorithms are similar.

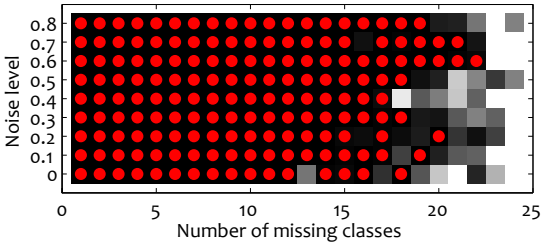


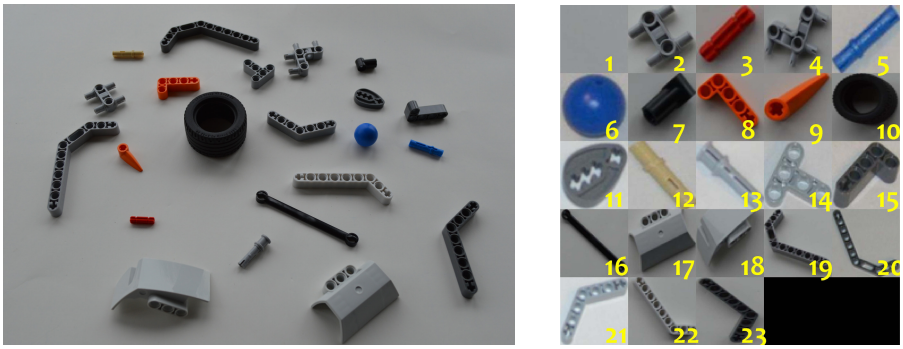
Fig. 2. Heat map of the p-value of the Wilcoxon signed rank test for equal medians of the consistency indices for the Hungarian and the Greedy algorithms. Dots signify statistically significant difference at $\alpha = 0.05$.

3.2 Who-Is-Missing: Objects in an Image

To illustrate the proposed solution in a real-life scenario, we took images of a set of 22 parts from a LEGO Mindstorms NXT kit (Figure 3). Each of the 28 images contained all 22 LEGO parts. After segmentation, 706 objects were detected, labelled, and saved as the training data. Five position-invariant and rotation-invariant features were extracted: eccentricity, solidity,[4], and the RGB colour of the object.

Each feature was standardised to mean 0 and standard deviation 1. To evaluate the potential of the data set, we applied Principal Component Analysis (PCA) to the data set and plotted the 23 classes in the space of the first two

[4] The `regionprops` MATLAB function was used.



(a) (b)

Fig. 3. (a) An image with all 22 LEGO pieces; (b) types of LEGO pieces with their class labels. Class 1 corresponds to "other".

principal components (Figure 4). The plot indicates that the classes are highly overlapping, which will prevent the a classifier of iid objects from achieving a high consistency for the who-is-missing problem.

Next we ran 10-fold cross-validation experiments with a small selection of classifiers from WEKA [8], using the default parameter settings. The classification accuracy ranged from 16.28% for AdaBoost.M1 [6] to 82.72% for Rotation Forest [17], revealing that the data set is not too easy and, at the same time, high classification accuracy is possible. This suits our purposes, as an ideal classifier will not need an assignment algorithm to solve the who-is-missing problem.

We trained and tested the nearest mean classifier for the set, obtaining a rather mediocre testing classification accuracy of 38.63%.

A new set of 100 images was collected, 25 with two random missing class, 25 with three random missing classes, 25 with four missing classes and the last 25 with five missing classes. Each image was segmented and the features of the objects were extracted. To eliminate the effect of inaccurate segmentation on the comparison of the Hungarian and the greedy algorithms, we accepted for this analysis only images where the number of segmented objects tied with the true number of non-missing classes. The averaged consistency indices for the different number of classes are shown in Table 2.

Table 2. Average values of the consistency indices for the LEGO data and the p-value of the Wilcoxon signed rank test

	Number of missing classes					
	2	3	4	5	Mean	p-value
# images	15	22	10	11		
$I_{\text{Hungarian}}$	0.2667	0.3333	0.1444	0.2706	0.2716	**0.0452**
I_{Greedy}	0.1200	0.2632	0.1750	0.1529	0.1900	

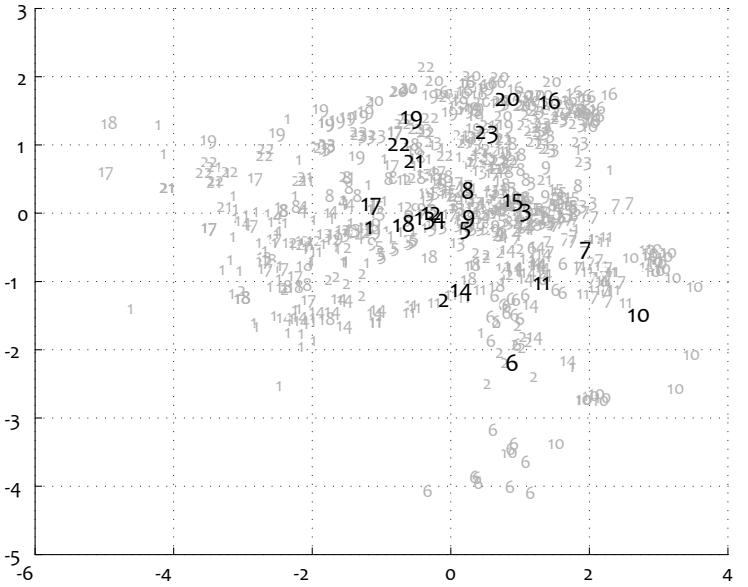


Fig. 4. A scatterplot of the 23 classes in the space of the first two principal components of the training data

While the recognition of the missing classes has not been very accurate, the results with the Hungarian Algorithm are markedly better for 2, 3 and 5 classes. We ran the Wilcoxon signed rank test for the zero median of the pairwise differences of the two consistency indices for all numbers of missing classes. The p-value, also shown in the table, indicates that the Hungarian algorithm outperforms significantly ($\alpha = 0.5$) the greedy algorithm for the who-is-missing problem.

4 Conclusions

This study formulates the who-is-missing problem and proposes a solution, completed with a measure of its quality. The Hungarian assignment algorithm, applied on the logarithms of the posterior probabilities of the objects in the set was found to dominate the intuitive alternative, called here the Greedy algorithm.

Many extensions and variants of the who-is-missing problem are yet to be formulated, for example, recognising one or more impostors in the given set, dealing with known, larger than 1, numbers of objects from each class classes, taking into the solution possible dependencies between the classes. Last but not least, important application niches for these new pattern recognition puzzles are yet to be discovered.

References

1. Amit, Y., Trouvé, A.: POP: patchwork of parts models for object recognition. International Journal of Computer Vision 75, 267–282 (2007)
2. Bourgeois, F., Lassalle, J.C.: An extension of the Munkres algorithm for the assignment problem to rectangular matrices. Communications ACM 14(12), 802–804 (1971)
3. Dearden, A., Demiris, Y., Grau, O.: Tracking football player movement from a single moving camera using particle filters. In: Proceedings of CVMP-2006, pp. 29–37. IET Press (2006)
4. Dietterich, T.G., Lathrop, R.H., Lozano-Perez, T.: Solving the multiple-instance problem with axis-parallel rectangles. Artificial Intelligence 89, 31–71 (1997)
5. Frank, A., Asuncion, A.: UCI machine learning repository (2010), `http://archive.ics.uci.edu/ml`
6. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
7. Fu, Z., Robles-Kelly, A., Zhou, J.: MILIS: multiple instance learning with instance selection. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(5), 958–977 (2011)
8. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations 11 (2009)
9. Kamisugi, Y., Furuya, N., Iijima, K., Fukui, K.: Computer-aided automatic identification of rice chromosomes by image parameters 1(3), 189–196 (1993)
10. Kaucic, R., Perera, A.G.A.: J., G.B., Kaufhold, Hoogs, A.: A unified framework for tracking through occlusions and across sensor gaps. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, vol. 1, pp. 1063–1069 (2005)
11. Kuncheva, L.I.: Full-class set classification using the Hungarian algorithm. International Journal of Machine Learning and Cybernetics 1(1), 53–61 (2010)
12. Kuncheva, L.: A stability index for feature selection. In: Proc. IASTED, Artificial Intelligence and Applications, Innsbruck, Austria, pp. 390–395 (2007)
13. Mangasarian, O.L., Wild, E.W.: Multiple instance classification via successive linear programming. Journal of Optimization Theory and Applications 137, 555–568 (2008)
14. McDowell, L.K., Gupta, K.M., Aha, D.W.: Cautious inference in collective classification. In: Processdings of AAAI, pp. 596–601 (2007)
15. Ming, D., Tian, J.: Automatic pattern extraction and classification for chromosome images. Journal of Infrared Milli Terahz Waves 31, 866–877 (2010)
16. Ning, X., Karypis, G.: The set classification problem and solution methods. In: Proceedings of SIAM Data Mining, pp. 847–858 (2009)
17. Rodríguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation forest: A new classifier ensemble method. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(10), 1619–1630 (2006)
18. Sen, P., Namata, G.M., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classification in network data. AI Magazine 29, 93–106 (2008)
19. Wang, J., Zucker, J.D.: Solving the multiple-instance problem: A lazy learning approach. In: Proceedings 17th International Conference on Machine Learning, pp. 1119–1125 (2000)

Edit Distance Computed by Fast Bipartite Graph Matching*

Francesc Serratosa and Xavier Cortés

Universitat Rovira i Virgili, Tarragona, Catalonia, Spain
{francesc.serratosa,xavier.cortes}@urv.cat

Abstract. We present a new algorithm to compute the Graph Edit Distance in a sub-optimal way. We demonstrate that the distance value is exactly the same than the one obtained by the algorithm called Bipartite but with a reduced run time. The only restriction we impose is that the edit costs have to be defined such that the Graph Edit Distance can be really defined as a distance function, that is, the cost of insertion plus deletion of nodes (or arcs) have to be lower or equal than the cost of substitution of nodes (or arcs). Empirical validation shows that higher is the order of the graphs, higher is the obtained Speed up.

Keywords: Graph Edit Distance, Bipartite Graph Matching, Munkres' algorithm.

1 Introduction

Attributed Graphs have been of crucial importance in pattern recognition throughout more than 3 decades [1], [2], [3], [4], [5], [6], [7], [8], [9] and [10]. If elements in pattern recognition are modelled through attributed graphs, error-tolerant graph-matching algorithms are needed that aim to compute a matching between nodes of two attributed graphs that minimizes some kind of objective function. Unfortunately, the time and space complexity to compute the minimum of these objective functions is very high. For this reason, some graph prototyping methods have appeared with the aim of reducing the run time while querying a graph in a large database [11], [12], [13].

Since its presentation, Bipartite algorithm [14] has been considered one of the best graph-matching algorithms due to it obtains a sub-optimal distance value almost near to the optimal one but with a considerable decrease on the run time. Other algorithms are [15] or [16]. There is an interesting survey in [2].

This paper presents a new algorithm that obtains exactly the same distance value of the Bipartite algorithm but with a reduced run time. The only restriction we impose is that the edit costs have to be defined such that the Graph Edit Distance can be really defined as a distance function, that is, the cost of insertion plus deletion of nodes (or arcs) have to be lower or equal than the cost of substitution of nodes (or arcs). Experimental validation shows a Speed up of 5 on well-known databases. In fact,

* This research is supported by the CICYT project DPI2013-42458-P.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 253–262, 2014.

higher is the order of graphs, higher is also the Speed up of our algorithm. This property is interesting since in the next years, we will see a need on representing the objects (social nets, scenes, proteins…) on larger structures.

The outline of the paper is as follows, in the next section, we define the attributed graphs and the graph-edit distance. On section 3, we explain how to compute the graph edit distance using the Bipartite algorithm. Finally, on section 4, we present our new method and we schematically show our algorithm. On section 5, we show the experimental validation and we finish the article with some conclusions.

2 Graphs and Graph Edit Distance

In this section, we first define the Attributed Graphs, Cliques and Graph matching and then we explain the Graph Edit Distance.

Attributed Graph and Cliques

Let $\Delta_v$and $\Delta_e$denote the domains of possible values for attributed vertices and arcs, respectively. An attributed graph (over Δ_v and Δ_e) is defined by a tuple $G = (\Sigma_v, \Sigma_e, \gamma_v, \gamma_e)$, where $\Sigma_v = \{v_a \mid a = 1, \dots, n\}$ is the set of vertices (or nodes), $\Sigma_e = \{e_{ab} | a, b \in 1, \dots, n\}$ is the set of arcs (or edges), $\gamma_v: \Sigma_v \rightarrow \Delta_v$ assigns attribute values to vertices and $\gamma_e: \Sigma_e \rightarrow \Delta_e$ assigns attribute values to arcs. The order of graph G is n.

We define a clique K_a on an attributed graph G as a local structure composed of a node and its outgoing edges $K_a = (v_a, \{e_{ab} | b \in 1, \dots, n\}, \gamma_v, \gamma_e)$.

Error Correcting Graph Isomorphism

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 attributed graphs of initial order n and m. To allow maximum flexibility in the matching process, graphs are extended with null nodes [17] to be of order $n + m$. We will refer to null nodes of G^p and G^q by $\hat{\Sigma}_v^p \subseteq \Sigma_v^p$ and $\hat{\Sigma}_v^q \subseteq \Sigma_v^q$ respectively. We assume null nodes have indices $a \in [n+1, \dots, n+m\}$ and $i \in [m+1, \dots, n+m\}$ for graphs G^p and G^q, respectively. Let T be a set of all possible bijections between two vertex sets Σ_v^p and Σ_v^q. We define the non-existent or null edges by $\hat{\Sigma}_e^p \subseteq \Sigma_e^p$ and $\hat{\Sigma}_e^q \subseteq \Sigma_e^q$.

Bijection $f^{p,q}: \Sigma_v^p \rightarrow \Sigma_v^q$, assigns one vertex of G^p to only one vertex of G^q. The bijection between arcs, denoted by $f_e^{p,q}$, is defined accordingly to the bijection of their terminal nodes.

Graph Edit Distance between Two Graphs

One of the most widely used methods to evaluate an error-correcting graph isomorphism is the Graph Edit Distance [18, 19, 20]. The dissimilarity is defined as the minimum amount of required distortion to transform one graph into the other. To this end, a number of distortion or edit operations, consisting of insertion, deletion and substitution of both nodes and edges are defined. Then, for every pair of graphs (G^p and G^q), there is a sequence of edit operations, or an edit path $editPath(G^p, G^q) = (\varepsilon_1, \dots, \varepsilon_k)$ (where each ε_i denotes an edit operation) that

transform one graph into the other. In general, several edit paths may exist between two given graphs. This set of edit paths is denoted by ϑ. To quantitatively evaluate which edit path is the best, edit cost functions are introduced. The basic idea is to assign a penalty cost to each edit operation according to the amount of distortion that it introduces in the transformation.

Each $editPath(G^p, G^q) \in \vartheta$ can be related to a univocal graph isomorphism $f^{p,q} \in T$ between the involved graphs. In this way, each edit operation assigns a node of the first graph to a node of the second graph. Deletion and insertion operations are transformed to assignations of a non-null node of the first or second graph to a null node of the second or first graph. Substitutions simply indicate node-to-node assignations. Using this transformation, given two graphs, G^p and G^q, and a bijection between their nodes, $f^{p,q}$, the graph edit cost is given by (Definition 7 of [21]):

$$EditCost(G^p, G^q, f^{p,q}) = \sum_{\substack{v_a^p \in \Sigma_v^p - \hat{\Sigma}_v^p \\ v_i^q \in \Sigma_v^q - \hat{\Sigma}_v^q}} C_{vs}(v_a^p, v_i^q) + \sum_{\substack{v_a^p \in \Sigma_v^p - \hat{\Sigma}_v^p \\ v_i^q \in \hat{\Sigma}_v^q}} C_{vd}(v_a^p, v_i^q) + \sum_{\substack{v_a^p \in \hat{\Sigma}_v^p \\ v_i^q \in \Sigma_v^q - \hat{\Sigma}_v^q}} C_{vi}(v_a^p, v_i^q) + \sum_{\substack{e_{ab}^p \in \Sigma_e^p - \hat{\Sigma}_e^p \\ e_{ij}^q \in \Sigma_e^q - \hat{\Sigma}_e^q}} C_{es}(e_{ab}^p, e_{ij}^q) + \sum_{\substack{e_{ab}^p \in \Sigma_e^p - \hat{\Sigma}_e^p \\ e_{ij}^q \in \hat{\Sigma}_e^q}} C_{ed}(e_{ab}^p, e_{ij}^q) + \sum_{\substack{e_{ab}^p \in \hat{\Sigma}_e^p \\ e_{ij}^q \in \Sigma_e^q - \hat{\Sigma}_e^q}} C_{ei}(e_{ab}^p, e_{ij}^q)$$

Where $f^{p,q}(v_a^p) = v_i^q$ and $f_e^{p,q}(e_{ai}^p) = e_{ij}^q$

where C_{vs} is the cost of substituting node v_a^p of G^p for node $f^{p,q}(v_a^p)$ of G^q, C_{vd} is the cost of deleting node v_a^p of G^p and C_{vi} is the cost of inserting node v_i^q of G^q. Equivalently for edges, C_{es} is the cost of substituting edge e_{ab}^p of graph G^p for edge $f_e^{p,q}(e_{ab}^p)$ of G^q, C_{ed} is the cost of assigning edge e_{ab}^p of G^p to a non-existing edge of G^q and C_{ei} is the cost of assigning edge e_{ab}^q of G^q to a non-existing edge of G^p. Note that we have not considered the cases in which two null nodes or null arcs are mapped. This is because this cost is zero by definition.

Finally, the Graph Edit Distance is defined as the minimum cost under any bijection in T:

$$EditDist(G^p, G^q) = \min_{f^{p,q} \in T} EditCost(G^p, G^q, f^{p,q})$$

Using this definition, the Graph Edit Distance essentially depends on C_{vs}, C_{vd}, C_{vi}, C_{es}, C_{ed} and C_{ei} functions and several definitions of these functions exist.

We say the optimal bijection, $f^{p,q*}$, is the one that obtains the minimum cost,

$$f^{p,q*} = \operatorname*{argmin}_{f^{p,q} \in T} EditCost(G^p, G^q, f^{p,q})$$

We define the distance and the optimal bijection between two cliques in a similar way as the distance between two graphs since they are local structures of graphs. We name the cost of substituting clique K_a^p by K_i^q as $\boldsymbol{C}_{a,i}$. The cost of deleting clique K_a^p as $\boldsymbol{C}_{a,\varepsilon}$ and the cost of inserting clique K_i^q as $\boldsymbol{C}_{\varepsilon,i}$.

3 Edit Distance Computation by Bipartite Algorithm (BP)

The assignment problem considers the task of finding an optimal assignment of the elements of a set A to the elements of another set B, where both sets have the same cardinality $n = |A| = |B|$. Let us assume there is a nXn cost matrix C. The matrix elements $C_{i,j}$ correspond to the cost of assigning the i-th element of A to the j-th element of B. An optimal assignment is the one that minimises the sum of the assignment costs and so, the assignment problem can be stated as finding the permutation p that minimises $\sum_{i=1}^{n} C_{i,p(i)}$. Munkres' algorithm [22] solves the assignation problem. It is a refinement of an earlier version by Kuhn [23] and is also referred to as Kuhn-Munkres or Hungarian algorithm. The algorithm repeatedly finds the maximum number of independent optimal assignments and in the worst case the maximum number of operations needed by the algorithm is $O(n^3)$. Later, an algorithm to solve this problem applied to non-square matrices where presented [24].

Bipartite, or BP for short [14], is an efficient algorithm for edit distance computation for general graphs that use the Munkres' algorithm. That is, they generalised the original Munkres' algorithm that solve the assignment problem to the computation of the graph edit distance by defining a specific cost matrix. In experiments on artificial and real-world data, authors demonstrate BP obtains an important speed-up of the computation respect other methods while at the same time the accuracy of the approximated distances is not much affected [14]. For this reason, since its publication, it has become one of the most used graph-matching algorithms.

Given attributed graphs G^p and G^q, the $(n+m)x(n+m)$ cost matrix C is defined as follows,

$$\mathbf{C} = \left[\begin{array}{cccc|cccc} c_{1,1} & c_{1,2} & \cdots & c_{1,m} & c_{1,\varepsilon} & \infty & \cdots & \infty \\ c_{2,1} & c_{2,2} & \cdots & c_{2,m} & \infty & c_{2,\varepsilon} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \infty \\ c_{n,1} & c_{n,2} & \cdots & c_{n,m} & \infty & \cdots & \infty & c_{n,\varepsilon} \\ \hline c_{\varepsilon,1} & \infty & \cdots & \infty & 0 & 0 & \cdots & 0 \\ \infty & c_{\varepsilon,2} & \ddots & \vdots & 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \infty & \vdots & \ddots & \ddots & 0 \\ \infty & \cdots & \infty & c_{\varepsilon,m} & 0 & \cdots & 0 & 0 \end{array}\right]$$

Where $\boldsymbol{C}_{a,i}$ denotes the cost of substituting clique K_a^p by K_i^q, $\boldsymbol{C}_{a,\varepsilon}$ denotes the cost of deleting clique K_a^p and $\boldsymbol{C}_{\varepsilon,i}$ denotes the cost of inserting clique K_i^q. On the basis of this cost matrix definition, Munkres' algorithm can be executed to find the minimum cost for all permutations. Obviously, as described in [14], this minimum cost is a

sub-optimal Edit Distance value between the involved graphs since cost matrix rows are related to cliques of graph G^p and columns are related to cliques of G^q. Moreover, it is considered a *correct permutation* the one that $\sum_{a=1}^{n+m} C_{a,p(a)} < \infty$. That is, all costs are assigned to non-infinitive values.

Note that Munkres' algorithm used in its original form is optimal for solving the assignment problem, but it is suboptimal for solving the graph edit distance. This is due to the fact that cliques are considered individually. The distance values obtained by this method are equal to or smaller than the distance values obtained in an optimal method (with exponential cost). The computational cost of the method is $O((n+m)^3)$.

4 Fast Bipartite Algorithm (FBP)

We define the following edit cost,

$$EditCost'^{(G^p,G^q,f^{p,q})} = EditCost(G^p,G^q,f^{p,q}) - C_{ve\_di}(G^p,G^q)$$

Where,

$$C_{ve\_di}(G^p,G^q) = \sum_{v_a^p \in \Sigma_v^p} C_{vd}(v_a^p, v_\phi^q) + \sum_{v_i^q \in \Sigma_v^q} C_{vi}(v_\phi^p, v_i^q) + \\ + \sum_{e_{ab}^p \in \Sigma_e^p} C_{ed}(e_{ab}^p, e_\phi^q) + \sum_{e_{ij}^q \in \Sigma_e^q} C_{ei}(e_\phi^p, e_{ij}^q)$$

and $v_\phi^q \in \hat{\Sigma}_v^q$, $v_\phi^p \in \hat{\Sigma}_v^p$, $e_\phi^q \in \hat{\Sigma}_e^q$ and $e_\phi^p \in \hat{\Sigma}_e^p$

The new edit cost definition subtracts from the original edit cost the costs of deleting nodes and arcs from G^p and inserting nodes and arcs from G^q. Nodes v_ϕ^q and v_ϕ^p (or arcs e_ϕ^q and e_ϕ^p) represent any extended node (or arc). Note that in cases that nodes or arcs where the extended ones (null nodes o arcs), that is, $v_a^p \in \hat{\Sigma}_v^p$, $v_i^q \in \hat{\Sigma}_v^q$, $e_{ab}^p \in \hat{\Sigma}_e^p$ or $e_{ij}^q \in \hat{\Sigma}_e^q$, then their corresponding cost is zero by definition. Moreover, the subtracted cost C_{ve_di} does not depend on any bijection $f^{p,q}$.

In a similar way than the original edit cost, the optimal bijection, $f'^{p,q*}$, is the one that obtains the minimum cost,

$$f'^{p,q*} = \underset{f^{p,q} \in T}{\operatorname{argmin}}\, EditCost'(G^p,G^q,f^{p,q})$$

We define D as an $n+m$ vector. The first n positions are filled with the costs of deleting cliques K_a^p that we named $\boldsymbol{C}_{a,\varepsilon}$, and the other m positions are filled with zeros: $D = [\boldsymbol{C}_{1,\varepsilon}, \ldots \boldsymbol{C}_{n,\varepsilon}, 0 \ldots 0]$. Moreover, we define I as an $m+n$ vector. The first m positions are filled with the costs of inserting cliques K_i^q that we named $\boldsymbol{C}_{\varepsilon,i}$, and the other n positions are filled with zeros: $I = [C_{\varepsilon,1}, \ldots C_{\varepsilon,m}, 0 \ldots 0]$. Note that zeros in both vectors represent the cost of deleting or inserting null cliques. Besides, it is easy to demonstrate that $C_{ve\_di}(G^p,G^q) = \langle \bar{1}, (I_a + D_a) \rangle$.

With these two vectors, we define two $(n + m)x(m + n)$ matrices, $\hat{D}$ and $\hat{I}$. The first one is obtained by the replication of vector D through columns and the second one is obtained by the replication of vector I through rows.

We are ready to define our cost matrix C′ as follows, C′=C-$\left(\hat{D} + \hat{I}\right)$,

$$C' = \left[\begin{array}{ccc|ccc} C_{1,1} - (C_{1,\varepsilon} + C_{\varepsilon,1}) & \cdots & & & C_{1,m} - (C_{1,\varepsilon} + C_{\varepsilon,m}) & \\ \vdots & & & & \vdots & \\ & & & & & \\ \vdots & & & & \vdots & \\ C_{n,1} - (C_{n,\varepsilon} + C_{\varepsilon,1}) & \cdots & & & C_{n,m} - (C_{n,\varepsilon} + C_{\varepsilon,m}) & \\ \hline 0 & \infty & \cdots & & \infty & \\ \infty & \ddots & & & & \\ \vdots & & 0 & & & \\ & & & 0 & \vdots & \\ & & & \infty & \ddots & \infty \\ \infty & & & \cdots & \infty & 0 \end{array}\right] \quad \left[\begin{array}{ccc} 0 & \infty & \cdots \\ \infty & \ddots & \\ \vdots & & 0 \\ & & \\ & & \\ & & \\ & & \end{array}\right] \quad \left[\begin{array}{ccc} & & \infty \\ & & \\ & & \\ 0 & & \vdots \\ \ddots & & \infty \\ \cdots & \infty & 0 \end{array}\right]$$

Similarly to cost matrix C, this matrix is composed of four quadrants. The dimensions of each quadrant is: $CQ1' = n\,X\,m$, $CQ2' = n\,X\,n$, $CQ3' = m\,X\,m$ and $CQ4' = m\,X\,n$. Cells in the first one are filled with the value of substituting the cliques (as in C) but the cost of deleting and inserting the respective cliques is subtracted. The second and third quadrants are composed of infinitive values except at the diagonal that is filled with zeros. All cells on the fourth quadrant have a zero.

On the basis of the new cost matrix C′ defined above, Munkres' algorithm [22] can be executed and it finds the optimal permutation p that minimises $\sum_{a=1}^{n+m} C'_{a,p(a)}$. Note that any correct permutation on p is equivalent to a bijection $f^{p,q}$ between nodes of graphs G^p and G^q.

It was demonstrated in [25] that, on the one hand, the equality $f'^{p,q^*} = f^{p,q^*}$ holds for all pair of graphs G^p and G^q. On the other hand, in the case that $C_{vs} \leq C_{vi} + C_{vd}$ and $C_{es} \leq C_{ei} + C_{ed}$ then the value of EditCost′ is equal to a correct permutation cost of C′. These demonstrations give us a way to compute the $EditCost$ through applying the Munkres' algorithm to matrix C′. Nevertheless, due to the dimensions of C′ are the same than the original C, the computational cost would be equivalent. Again, in [25], it was demonstrated that, in the case that $C_{vs} \leq C_{vi} + C_{vd}$ and $C_{es} \leq C_{ei} + C_{ed}$ then minimising a permutation on C′ is exactly the same than minimising a permutation on the sub-matrix composed by the first quadrant composed of the first n rows and m columns, that we call it $CQ1'$.

Algorithm 1 computes the $Fast\ Bipartite$.

Algorithm 1. Fast Bipartite

$CQ1' = \text{Computation\_Cost}(G^p, G^q)$

// $CQ1'$ is the nXm first quadrant of cost matrix C′

$P = \text{Munkres}(CQ1')$.

// P is the nXm permutation matrix

$\text{EditCost}' = \text{Sum}\left(Sum(P.*\ CQ1')\right)$.

// .∗ represents the multiplication element by element

$\text{EditCost} = \text{EditCost}' + C_{ve\_di}$

// Final distance value

End.

As commented, the Munkres algorithm was initially implemented to find the permutation of a quadratic matrix. In case $m \neq n$, matrix $C\mathrm{Q}1'$ can be extended with negative values (lower than any original cost). Nevertheless, it is usual the implemented functions of the Munkres' algorithm to automatically enlarge the cost matrix. The worst computational cost of Fast Bipartite is the cost of the Munkres' algorithm, that is: $O(\max(m,n)^3)$. The cost of Bipartite algorithm [14] is $O((m+n)^3)$.

5 Experimental Validation

The goodness of the Bipartite algorithm has been tested in several papers, for this reason, we only want to present the Speed up of our method respect the classical one [14]. We do not present new recognition-ratio tests or correlation tests between the sub-optimal distance and the optimal one since we obtain exactly the same distance value (as described above) when the costs are defined such as $C_{vs} \leq C_{vi} + C_{vd}$ and $C_{es} \leq C_{ei} + C_{ed}$.

We present two different tests. The aim of the first one is to execute again the tests published in paper [14] where the Bipartite algorithm was presented and show the Speed up of our method on these largely used databases known as IAM graph database repository. These databases have been used during some years to test different a types of algorithms related to graphs such as, classification, clustering, prototyping or graph embedding. We do not want to add any comment to these databases since a lot of literature has been written talking about them. The first explanations of these databases can be found in [26] and also they have been commented in [14]. They can be downloaded from the IAPR-TC15 web page [27].

In the experiments, we compared all graphs on the test set respect all graphs on the reference set as authors did in [14]. Then, from this large number of comparisons, we have extracted the mean computational times $\bar{t}_{BP}$ and $\bar{t}_{FBP}$ and the Speed up (table 1). We also show the mean and maximum number of graphs of each database. The source code in Matlab can be downloaded from [28].

Table 1. Mean computational time of Bipartite and Fast Bipartite and Speed Up of Fast Bipartite respect Bipartite

	Mean Order	Max Order	$\bar{t}_{BP}$ (mS)	$\bar{t}_{FBP}$ (mS)	Speed Up
Letter (L)	4.5	8	1.6	1.4	1.08
Letter (M)	4.6	10	2.1	1.8	1.13
Letter (H)	4.7	9	1.8	1.6	1.14
COIL	3.0	11	2.2	1.9	1.18
GREC	12.9	39	29.3	13.1	2.23
Fingerprint	5.4	26	9.8	6.2	1.60
Molecules	9.5	85	391.3	99.7	3.92
Proteins	32.6	126	1460.6	278	5.25

The Speed up is higher than one on the whole experiments therefore it is always worth to use the Fast Bipartite instead of the Bipartite algorithm. Moreover, the higher is the mean and the maximum number of nodes, the higher is the Speed up.

The aim of the second test is to show how the Fast Bipartite algorithm performs when it is not guarantee that $C_{vs} \leq C_{vi} + C_{vd}$ and $C_{es} \leq C_{ei} + C_{ed}$. In some applications, this is a too strong constrain. For instance, palmprint identification is an interesting application since the number of minutiae is around 1000 and it is usual to represent the palmprint image in an attributed graph where nodes represent minutiae and arcs represent the proximity relation (Delaunay triangulation or Nearest-neighbours). Attributes on nodes are the angle of minutiae and edges do not have attributes. The distance between node attributes is the angular distance between the angles of both minutiae. Therefore, if angles are presented by degrees, the maximum distance is 180. For this reason, if we want to fulfil constrain $C_{vs} \leq C_{vi} + C_{vd}$ then we need C_{vi} and C_{vd} to be bigger or equal than 90. Nevertheless, it is usual to consider two minutiae cannot be mapped if the difference between the angles is lower than around 30 degrees. So, from the application point of view we wish $C_{vi} = 30$ and $C_{vd} = 30$. In this second experiment, we show in which extend our algorithm obtains the same distance value obtained by the Bipartite algorithm although the constrain $C_{vs} \leq C_{vi} + C_{vd}$ is not fulfilled.

We used images contained in the Tsinghua 500 PPI Palmprint Database [29]. It is a public high-resolution palmprint database composed of 500 palmprint images of 2040 x 2040 resolution and captured with a commercial palmprint scanner from Hisign. From each person, 16 palmprints are enrolled (8 ones per each hand). We used the algorithm presented in [30] to extract the minutiae from each image. Table 1 shows the mean and max order of the generated graphs as well as the average run time and the Speed up of FBP respect BP. Our database, called Tarragona Palmprint, is public available at [31].

Table 2. Mean computational time of Bipartite and Fast Bipartite and Speed Up of Fast Bipartite respect Bipartite

	Mean Order	Max Order	$\bar{t}_{BP}$ (S)	$\bar{t}_{FBP}$ (S)	Speed-Up
Palmprint	987	1505	365,35	40.46	9.03

Table 3 shows the average Distance Error computed as $Distance\ Error = 1 - \frac{Edit\ Distance_{BP}}{Edit\ Distance_{FBP}}$ while computing these two algorithms with several edit costs C_{vi} and C_{vd}. Recall that always holds $Edit\ Distance_{BP} \leq Edit\ Distance_{FBP}$ and $Edit\ Distance_{BP} = Edit\ Distance_{FBP}$ when $C_{vs} \leq C_{vi} + C_{vd}$ and $C_{es} \leq C_{ei} + C_{ed}$. If we wish $C_{vi} = C_{vd} = 30$, as commented above, the average error is lower than 1% and we achieve a Speed Up higher than 9. The run time of these algorithms is independent of the edit costs.

Table 3. Distance between the distance value obtained using BP and FBP when several costs are applied

$C_{vi} = C_{vd}$	10	20	30	40	60	80
Distance Error	0.145	0.019	0.009	0	0	0

6 Conclusions

This paper presents a new algorithm called Fast Bipartite to compute the Graph Edit Distance that obtains exactly the same distance value than the Bipartite algorithm but with a reduced run time. The only restriction we impose is that the edit costs have to be defined such that the insertion plus deletion have to be greater or equal than the substitution. This is a logical restriction since it is needed to be the Edit Distance defined as a distance function. Empirical evaluation shows the Fast Bipartite is always faster than the Bipartite algorithm and higher is the order of both graphs, better is the Speed up we obtain. Moreover, in cases that the application imposes not to hold this restriction, the algorithm also achieves a high Speed up with a reduced error.

References

1. Sanfeliu, A., Alquézar, R., Andrade, J., Climent, J., Serratosa, F., Vergés, J.: Graph-based Representations and Techniques for Image Processing and Image Analysis. Pattern Recognition 35(3), 639–650 (2002)
2. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty Years of Graph Matching in Pattern Recognition. IJPRAI 18(3), 265–298 (2004)
3. Vento, M.: A One Hour Trip in the World of Graphs, Looking at the Papers of the Last Ten Years. In: Kropatsch, W.G., Artner, N.M., Haxhimusa, Y., Jiang, X. (eds.) GbRPR 2013. LNCS, vol. 7877, pp. 1–10. Springer, Heidelberg (2013)
4. Edwin, R., Hancock, R.C.: Pattern analysis with graphs: Parallel work at Bern and York. Pattern Recognition Letters 33(7), 833–841 (2012)
5. Serratosa, F., Cortés, X., Solé-Ribalta, A.: Component Retrieval based on a Database of Graphs for Hand-Written Electronic-Scheme Digitalisation. Expert Systems With Applications, ESWA 40, 2493–2502 (2013)
6. Serratosa, F., Alquézar, R., Amézquita, N.: A Probabilistic Integrated Object Recognition and Tracking Framework. Expert Systems With Applications 39, 7302–7318 (2012)
7. Solé-Ribalta, A., Serratosa, F.: Graduated Assignment Algorithm for Multiple Graph Matching based on a Common Labelling. International Journal of Pattern Recognition and Artificial Intelligence, IJPRAI 27 (1), 1–27 (2013)
8. Solé, A., Serratosa, F.: Models and Algorithms for computing the Common Labelling of a set of Attributed Graphs. Computer Vision and Image Understanding, CVIU 115(7), 929–945 (2011)
9. Sanromà, G., Alquézar, R., Serratosa, F.: A New Graph Matching Method for Point-Set Correspondence using the EM Algorithm and Softassign. Computer Vision and Image Understanding, CVIU 116(2), 292–304 (2012)
10. Sanromà, G., Alquézar, R., Serratosa, F., Herrera, B.: Smooth Point-set Registration using Neighbouring Constraints. Pattern Recognition Letters, PRL 33, 2029–2037 (2012)

11. 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)
12. Ferrer, M., Valveny, E., Serratosa, F.: Median graphs: A genetic approach based on new theoretical properties. Pattern Recognition 42(9), 2003–2012 (2009)
13. Ferrer, M., Valveny, E., Serratosa, F.: Median graph: A new exact algorithm using a distance based on the maximum common subgraph. Pattern Recognition Letters 30(5), 579–588 (2009)
14. Riesen, K., Bunke, H.: Approximate graph edit distance computation by means of bipartite graph matching. Image Vision Comput 27(7), 950–959 (2009)
15. Gold, S., Rangarajan, A.: A Graduated Assignment Algorithm for Graph Matching. IEEE TPAMI 18(4), 377–388 (1996)
16. Rebagliati, N., Solé, A., Pelillo, M., Serratosa, F.: Computing the Graph Edit Distance Using Dominant Sets. In: International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan, pp. 1080–1083 (2012)
17. Wong, A., You, M.: Entropy and Distance of Random Graphs with Application to Structural Pattern Recognition. Transaction on Pattern Analysis and Machine Intelligence PAMI-7(5), 599–609 (1985)
18. 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)
19. Gao, X., et al.: A survey of graph edit distance. Pattern Analysis and Applications 13(1), 113–129 (2010)
20. Solé, A., Serratosa, F., Sanfeliu, A.: On the Graph Edit Distance cost: Properties and Applications. International Journal of Pattern Recognition and Artificial Intelligence 26(5) (2012)
21. Bunke, H.: Error Correcting Graph Matching: On the Influence of the Underlying Cost Function. Trans. on Pattern Analysis and Machine Intelligence 21(9), 917–922 (1999)
22. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society for Industrial & Applied Mathematics 5, 32–38 (1957)
23. Kuhn, H.: The Hungarian method for the assignment problem. Naval Research Logistic Quarterly 2, 83–97 (1955)
24. Bourgeois, F., Lassalle, J.: An extension of the Munkres algorithm for the assignment problem to rectangular matrices. Communications of the ACM 14(12), 802–804 (1971)
25. Serratosa, F.: Fast Computation of Bipartite Graph Matching. Pattern Recognition Letters (2014)
26. 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)
27. http://iapr-tc15.greyc.fr/links.html
28. http://deim.urv.cat/~francesc.serratosa/SW/
29. Funada, J., et al.: Feature Extraction Method for Palmprint Considering Elimination of Creases. In: Proc. 14th Int. Conf. Pattern Recognition, pp. 1849–1854 (1998)
30. Jain, A.K., Feng, J.: Latent Palmprint Matching. IEEE Trans. on PAMI (2009)
31. http://deim.urv.cat/~francesc.serratosa/databases/

Statistical Method for Semantic Segmentation of Dominant Plane from Remote Exploration Image Sequence

Shun Inagaki[1] and Atsushi Imiya[2]

[1] School of Advanced Integration Science, Chiba University
[2] Institute of Management and Information Technologies, Chiba University
Yayoicho 1-33, Inage-ku, Chiba, 263-8522, Japan

Abstract. For the application of well-established image analysis algorithms to low frame-rate image sequences, which are common in bio-imaging and long-distance extrapolation, we are required to up-convert the frame-rate of image sequences. For the motion analysis of low frame-rate image sequences, we introduce a method for semantic segmentation of the dominant plane, which is the largest planar area on an image plane, from a low frame-rate image sequence, which is common for image sequence obtained by remote extrapolation.

1 Introduction

In this paper, we introduce a method for semantic segmentation of the dominant plane from the optical flow field of a low frame-rate image sequence combining image registration [5] and optical flow computation [8, 7].

The optical flow field is a fundamental feature for the interpretation of temporal image sequences [6]. For the motion analysis from low frame-rate image sequence, we are required to generate a dense optical flow fields, since well-established algorithms for motion analysis do not assume the use of low frame-rate image sequences.

For the detection of safe areas for navigation, the robot probe detects the dominant plane, which is the largest planar area on an image plane, from a sequence of images captured by a camera mounted on the robot. In the image, the safe areas and obstacle areas for navigation are detected using the optical flow field and homography of the ground plane [2–4]. Figure 1 (a) shows a cycle for autonomous navigation using optical flow.

Using an uncalibrated monocular camera as a sensor for obtaining information on the environment, in ref. [1], a featureless robot navigation method based on a planar area and an optical flow field computed from a pair of successive images is proposed. A planar area in an environment is called a dominant plane, and it corresponds to the largest part of an image. We accept the following five assumptions.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 263–272, 2014.

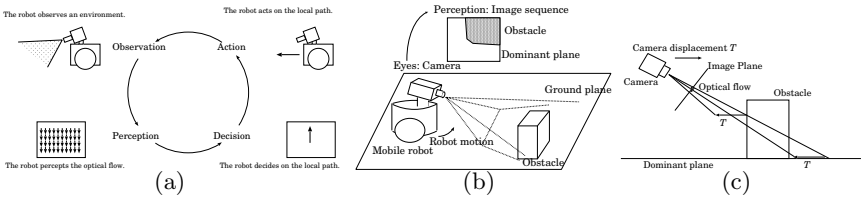


Fig. 1. Observation-Perception-Decision-Action cycle for vision-based robot navigation. (a) First, a mobile robot equipped with a camera observes the environment. Next, an optical flow field relative to the robot motion is computed from images obtained by the camera. The optical flow field is used to decide the local path. (b) The mobile robot has a camera, which corresponds to its eyes. The robot perceives an optical flow field from its ego-motion. (c) If the camera moves a distance T approximately parallel to the dominant plane, the optical flow vectors on the obstacle and on the dominant plane areas have the same distance T. However, they differ at the same time.

1. The ground plane is the planar area.
2. The camera mounted on the mobile robot is looking downward.
3. The robot observes the environment using the camera mounted on itself for navigation.
4. The camera on the robot captures a sequence of images while the robot is moving.
5. The planar area occupies more than 1/2 on the image.

These assumptions are illustrated in Figs. 1 (b) and (c).

Since the planar flow vector on the ground plane is equal to the optical flow vector $\dot{\boldsymbol{x}}$ on the dominant plane, we use the difference between these two flows to detect the dominant plane. From the assumption 1 and 2, we have the following property.

Property 1. *Corresponding points on a dominant plane in a pair of successive images are combined by homography.*

2 Dominant Plane and Optical Flow

Setting $\boldsymbol{H}$ to be a 3×3 matrix [14], the homography between two images of a planar surface can be expressed as $\boldsymbol{\xi}' = \boldsymbol{H}\boldsymbol{\xi}$, where $\boldsymbol{\xi} = (x, y, 1)^\top$ and $\boldsymbol{\xi}' = (x', y', 1)^\top$ are the homogeneous coordinates of corresponding points in two successive images. Assuming that the camera displacement is small, the matrix $\boldsymbol{H}$ can be approximated by affine transformations. These geometrical and mathematical assumptions are valid when the camera is mounted on a mobile robot moving on the dominant plane. Therefore, the corresponding points $\boldsymbol{x} = (x, y)^\top$ and $\boldsymbol{x}' = (x', y')^\top$ on the dominant plane are related by $\boldsymbol{x}' = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$, where $\boldsymbol{A}$ and $\boldsymbol{b}$ are a 2×2 affine-coefficient matrix and a two-dimensional vector, respectively, which are approximations of $\boldsymbol{H}$. This geometric relation implies that $\boldsymbol{u}(x, y, t+1) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} - \boldsymbol{x}$ and that the next property is satisfied.

Property 2. *On the dominant plane, the optical flow vector is stationary and the planar-flow vector on the ground plane is equal to the optical flow vector* $\dot{\boldsymbol{x}} = \boldsymbol{u}$.

We call $\boldsymbol{x}' - \boldsymbol{x}$ the *planar flow*. The RANSAC-based method [11] for the estimation of the affine coefficients is described as Algorithm 1.

Algorithm 1. Planar Flow

Require: Planar area u
Ensure: Affine coefficients $\boldsymbol{A}$ and $\boldsymbol{b}$
  Set the region counter $m \leftarrow 0$
  repeat
    Randomly select three points from $\{\boldsymbol{x}\}$;
    Estimate $\boldsymbol{A}$ and $\boldsymbol{b}$ in $\boldsymbol{x}' = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$
    Compute planar flow field $\boldsymbol{u}' = \boldsymbol{x}' - \boldsymbol{x}$
    if $|\boldsymbol{u} - \boldsymbol{u}'| < \varepsilon$ and $\#(|\boldsymbol{u} - \boldsymbol{u}'| < \varepsilon) > m$ **then**
      assign these points as the plane;
      $m \leftarrow \#(|\boldsymbol{u} - \boldsymbol{u}'| < \varepsilon)$
    end if
  until predetermined number of times;

Once the affine coefficients are estimated, we can extract a segment for the dominant plane. We use the difference between these two flows for semantic segmentation of the dominant plane. Therefore, if $|\boldsymbol{u}(x, y, t) - \boldsymbol{u}(x, y, t+1)| < \epsilon$ for a small positive number ϵ, we conclude that the point $\boldsymbol{x} = (x, y)^\top$ lies on the dominant plane. If an obstacle exists in front of the robot, the planar flow on the image plane differs from the optical flow on the image plane as shown in Fig. 1 (c).

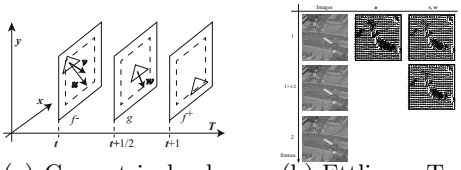
3 Semantic Segmentation Using Subframe Motion

We develop an algorithm to compute the optical flow field of a temporal image sequence $f(\boldsymbol{x}, t + \frac{1}{2})$, in which we set $\boldsymbol{u}_{\frac{1}{2}}(\boldsymbol{x}, t)$ from $f(\boldsymbol{x}, t)$ and $f(\boldsymbol{x}, t+1)$. For the convenience of analysis, we set

$$f^+(\boldsymbol{x}) = f(\boldsymbol{x}, t+1),\ f^-(\boldsymbol{x}, t) = f(\boldsymbol{x}, t),\ g(\boldsymbol{x}) = f(\boldsymbol{x}, t + \frac{1}{2}), \tag{1}$$

$$\boldsymbol{v} = \boldsymbol{u}_{\frac{1}{2}}(\boldsymbol{x}, t),\ \ \boldsymbol{w} = \boldsymbol{u}_{\frac{1}{2}}(\boldsymbol{x}, t + \frac{1}{2}). \tag{2}$$

Algorithm 2 shows the procedure for dominant plane detection using subframe optical flow computation.



(a) Geometrical relation (b) Ettlinger-Tor

Fig. 2. Geometrical relations among images sequence and computational results. (a) Geometrical relations among g, f^-, f^+, $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{w}$. (b) Computational results the original and subframe optical flow fields for the Etllinger-Tor sequence.

For large-displacement image sequences, we compute the dominant plane from $\boldsymbol{v}$ and $\boldsymbol{w}$, which are computed from $f(x, y, t)$ and $f(x, y, t + 1/2)$, and from $f(x, y, t + 1/2)$ and $f(x, y, t + 1)$, respectively, using the subframe optical flow computation method derived in the previous section. We conclude that the point $\boldsymbol{x} = (x, y)^\top$ lies on the dominant plane if $|\boldsymbol{u} - \boldsymbol{w}| < \epsilon$.

For $f(\boldsymbol{x}, t)$ and $f(\boldsymbol{x}, t + 1)$, if the relation

$$\nabla f(\boldsymbol{x}, t) \cong \nabla f(\boldsymbol{x}, t + \frac{1}{2}) = \nabla f(\boldsymbol{x}, t) + \delta(\boldsymbol{x}) \tag{3}$$

is satisfied, we have the next theorem.

Theorem 1. *The relation* $\nabla f^\top(\boldsymbol{v}+\boldsymbol{w})+f_t \cong 0$*, where* $\nabla f(\boldsymbol{x}, t)^\top \boldsymbol{v}+f_t(\boldsymbol{x}, t) = 0$ *and* $\nabla f(\boldsymbol{x}, t + \frac{1}{2})^\top \boldsymbol{w} + f_t(\boldsymbol{x}, t + \frac{1}{2}) = 0$*, is satisfied* [1]*.*

Setting $g(\boldsymbol{x}) = f^+(\boldsymbol{x} - \boldsymbol{w})$, $g(\boldsymbol{x}) = f^-(\boldsymbol{x} + \boldsymbol{v})$ and $\boldsymbol{u} = \boldsymbol{v} + \boldsymbol{w}$, Theorem 1 implies that we can have g, $\boldsymbol{v}$ and $\boldsymbol{w}$ as the minimisers of

$$J(g, \boldsymbol{v}, \boldsymbol{w}) = I_+ + I_- + \alpha G + \beta U + \gamma V, \tag{4}$$

where

$$I_+ = \int_{\boldsymbol{\Omega}} (g(\boldsymbol{x}) - f^+(\boldsymbol{x} - \boldsymbol{w}))^2 d\boldsymbol{x},\ I_- = \int_{\boldsymbol{\Omega}} (g(\boldsymbol{x}) - f^-(\boldsymbol{x} + \boldsymbol{v}))^2 d\boldsymbol{x}, \tag{5}$$

$$G = \int_{\boldsymbol{\Omega}} |\nabla g|^2 d\boldsymbol{x},\ U = \int_{\boldsymbol{\Omega}} (|\nabla \boldsymbol{v}|^2 + |\nabla \boldsymbol{w}|^2) d\boldsymbol{x},\ V = |\boldsymbol{v} + \boldsymbol{w} - \boldsymbol{u}|^2. \tag{6}$$

Figure 2 shows the relations among $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$ and g and computational results of them.

Using the subframe optical flow, we extract a segment corresponding to the dominant plane D_H using Algorithm 2. In Algorithm 2, for a pair of successive images f_i and f_{i+1}, $\boldsymbol{u}_i$ is the optical flow field between f_i and f_{i+1}, and $\boldsymbol{v}_i$ and $\boldsymbol{w}_i$ are the subframe optical flow fields between f_i and g, and between g and

[1] From eqs. (1) and (3), we have the relation $\nabla f(\boldsymbol{x}, t)^\top (\boldsymbol{v} + \boldsymbol{w}) + f_t \cong 0$, since $f_t(\cdot, t + \frac{1}{2}) = f(\boldsymbol{x}, t + 1) - f(\boldsymbol{x}, t + \frac{1}{2})$ and $f_t(\cdot, t + 1) = f(\boldsymbol{x}, t + 1\frac{1}{2}) - f(\boldsymbol{x}, t)$.

f_{i+1}, respectively, where g is the interframe image. Furthermore, the procedure Compute InterFrame$(f_i, f_{i+1}, \boldsymbol{u}_i, \alpha, \beta, \gamma)$ computes g_i, $\boldsymbol{v}_i$ and $\boldsymbol{w}_i$ from f_i, f_{i+1} and $\boldsymbol{u}_i$ using the method proposed in the previous section. Moreover, in the algorithm $\boldsymbol{u}_i$ is computed from f_i and f_{i+1} using the large-displacement optical flow computation technique in [13].

Algorithm 2. Plane Detection

Require: Images f_i, f_{i+1}, f_{i+2}, Flow field $\boldsymbol{u}_i, \boldsymbol{u}_{i+1}$, Parameters α, β, γ
Ensure: Plane D
  $(g_i, \boldsymbol{v}_i, \boldsymbol{w}_i) \Leftarrow$ Compute InterFrame$(f_i, f_{i+1}, \boldsymbol{u}_i, \alpha, \beta, \gamma)$
  Set the region counter $m \leftarrow 0$
  repeat
    Compute affine coefficients by Planar-Flow$(\boldsymbol{v}_i)$;
    Estimate planar flow field $\hat{\boldsymbol{v}}_i$ from affine coefficients;
    if $|\boldsymbol{w}_i - \hat{\boldsymbol{v}}_i| < \varepsilon$ and $\#(|\boldsymbol{w}_i - \hat{\boldsymbol{v}}_i| < \varepsilon) > m$ **then**
      assign these points as the plane d;
      $m \leftarrow \#(|\boldsymbol{w}_i - \hat{\boldsymbol{v}}_i| < \varepsilon)$
    end if
  until predetermined number of times;
  output the plane D_H as a binary image;

From $J(g, \boldsymbol{v}, \boldsymbol{w})$, for g, $\boldsymbol{v}$ and $\boldsymbol{w}$, we have the system of partial differential equations

$$\Delta g - \frac{1}{\beta} F(g, \boldsymbol{v}, \boldsymbol{w}) = 0,\ \Delta \boldsymbol{v} - \frac{1}{\alpha} G^-(g, \boldsymbol{u}, \boldsymbol{w}) = 0,\ \Delta \boldsymbol{w} - \frac{1}{\alpha} G^+(g, \boldsymbol{u}, \boldsymbol{w}) = 0, \quad (7)$$

where

$$\begin{aligned} F(g, \boldsymbol{v}, \boldsymbol{w}) &= 2g(\boldsymbol{x}) - (f^-(\boldsymbol{x} + \boldsymbol{v}) + f^+(\boldsymbol{x} - \boldsymbol{w})), \\ G^-(g, \boldsymbol{v}, \boldsymbol{w}) &= \gamma(\boldsymbol{v} + \boldsymbol{w} - \boldsymbol{u}) + (f^-(\boldsymbol{x} + \boldsymbol{v}) - g(\boldsymbol{x}))\nabla f^-(\boldsymbol{x} + \boldsymbol{v}), \\ G^+(g, \boldsymbol{u}, \boldsymbol{w}) &= \gamma(\boldsymbol{v} + \boldsymbol{w} - \boldsymbol{u}) + (g(\boldsymbol{x}) - f^+(\boldsymbol{x} - \boldsymbol{w}))\nabla f^+(\boldsymbol{x} - \boldsymbol{w}). \end{aligned} \quad (8)$$

The mimimisation of $J(g, \boldsymbol{v}, \boldsymbol{w})$ is achieved by numerically solving eq. (7). Using semi-implicit discretisation of the associated diffusion equations such that

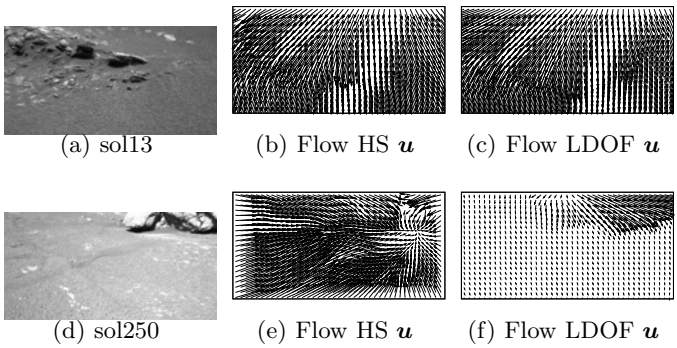
$$\begin{aligned} \partial_t g &= \Delta g - \frac{1}{\beta} F(g, \boldsymbol{v}, \boldsymbol{w}), \\ \partial_t \boldsymbol{v} &= \Delta \boldsymbol{v} - \frac{1}{\alpha} G^-(g, \boldsymbol{u}, \boldsymbol{w}), \\ \partial_t \boldsymbol{w} &= \Delta \boldsymbol{w} - \frac{1}{\alpha} G^+(g, \boldsymbol{u}, \boldsymbol{w}), \end{aligned} \quad (9)$$

we solve the system of iteration forms [12]

$$\begin{aligned}
\frac{g^{(n+1)} - g^{(n)}}{\tau} &= \Delta g^{(n+1)} - \frac{1}{\beta} F(g^{(n)}, \boldsymbol{v}^{(n)}, \boldsymbol{w}^{(n)}), \\
\frac{\boldsymbol{v}^{(n+1)} - \boldsymbol{v}^{(n)}}{\tau} &= \Delta \boldsymbol{v}^{(n+1)} - \frac{1}{\alpha} G^{-}(g^{(n)}, \boldsymbol{u}^{(n)}, \boldsymbol{w}^{(n)}), \\
\frac{\boldsymbol{w}^{(n+1)} - \boldsymbol{w}^{(n)}}{\tau} &= \Delta \boldsymbol{w}^{(n+1)} - \frac{1}{\alpha} G^{+}(g^{(n)}, \boldsymbol{u}^{(n)}, \boldsymbol{w}^{(n)}).
\end{aligned} \tag{10}$$

4 Numerical Examples

In experiments, we compared the statistics of the optical flow field computed by our method and by the pyramid-based Horn-Schunck method [8]. On the ground plane which corresponds to the dominant plane, since the optical flow field is smooth, we adopt the L_2 regularisation terms, that is, the regularisation term of the Horn-Schunck method is the square of the Frobenius norm $tr\nabla \boldsymbol{u} \nabla \boldsymbol{u}^{\top}$ of the vector gradient $\nabla \boldsymbol{u}$ of the optical flow field $\boldsymbol{u}$.

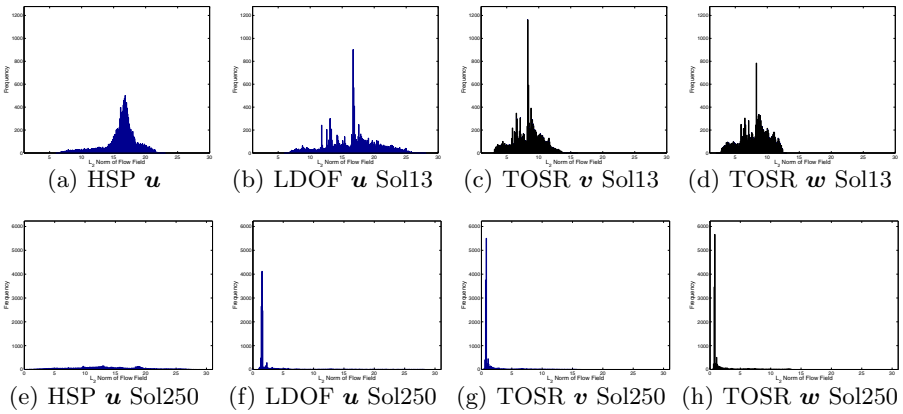


(a) sol13 (b) Flow HS $\boldsymbol{u}$ (c) Flow LDOF $\boldsymbol{u}$

(d) sol250 (e) Flow HS $\boldsymbol{u}$ (f) Flow LDOF $\boldsymbol{u}$

Fig. 3. Images and optical flows of Sol13 and Sol250. (a) Sol13. (b) Optical flow of Sol13 computed by the Horn-Schanck method with pyramid transform [9]. (c) Optical flow of Sol13 computed by the large-displacement method [13]. (d) Sol250. (e) Optical flow of Sol250 computed by the Horn-Schanck method with the pyramid transform [9]. (f) Optical flow of Sol250 computed by the large-displacement method [13].

Figure 3 shows images and their optical flow fields computed by two methods. The top and bottom rows show images and optical flow fields of Sol13 and Sol205, respectively. (b) and (e) show optical flow fields of Sol13 and Sol250, respectively, computed by the Horn-Schanck method [8] with pyramid transform (HSP). (c) and (f) show optical flow fields of Sol13 and Sol 250, respectively, computed by the Large-Displacement Optical Flow method [13] (LDOF).



(a) HSP $\boldsymbol{u}$ (b) LDOF $\boldsymbol{u}$ Sol13 (c) TOSR $\boldsymbol{v}$ Sol13 (d) TOSR $\boldsymbol{w}$ Sol13

(e) HSP $\boldsymbol{u}$ Sol250 (f) LDOF $\boldsymbol{u}$ Sol250 (g) TOSR $\boldsymbol{v}$ Sol250 (h) TOSR $\boldsymbol{w}$ Sol250

Fig. 4. Norm Histogram of Sol13 and Sol250. The top and bottom rows shows results for Sol13 and Sol250, respectively. From left to right, the histograms of $\boldsymbol{u}$ computed by the Horn-Schunck method with pyramid transform, $\boldsymbol{u}$ computed by the large-displacement optical flow method, $\boldsymbol{v}$ and $\boldsymbol{w}$, which are computed by our method -the temporal optical flow superresolurion, respectively.

Table 1. Kurtosis of histograms

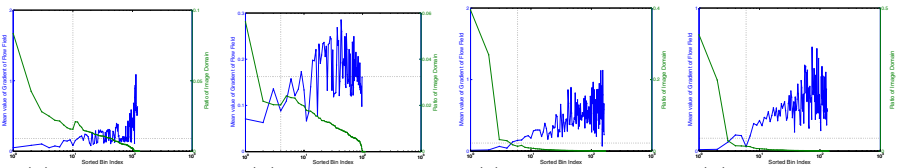
	HS $\boldsymbol{u}$	LDOF $\boldsymbol{u}$	TOSR $\boldsymbol{v}$	TOSR $\boldsymbol{w}$
Sol13	7.04	50.33	33.66	15.82
Sol250	2.71	115.51	92.75	81.63

Figure 4 shows the histograms for the l_2-norms of the optical flow vectors for the results of Fig. 3. In Fig. 4, the top and bottom rows show results for Sol13 and Sol250, respectively. Furthermore, from left to right, the histograms of $\boldsymbol{u}$ computed by the HSP method, $\boldsymbol{u}$ computed by the LDOF method, $\boldsymbol{v}$ and $\boldsymbol{w}$, which are computed by our flow-field up-conversion based on the Temporal Optical flow Superresolution (TOSR).

Figure 5 shows the second time derivatives of the histograms, since the second time derivatives of distributions allow to detect peaks in the distributions. Table 1 shows the kurtoses of optical flow fields computed by the HSP method, the LDOF method and TOSR method. Kurtoses of optical flow field computed by

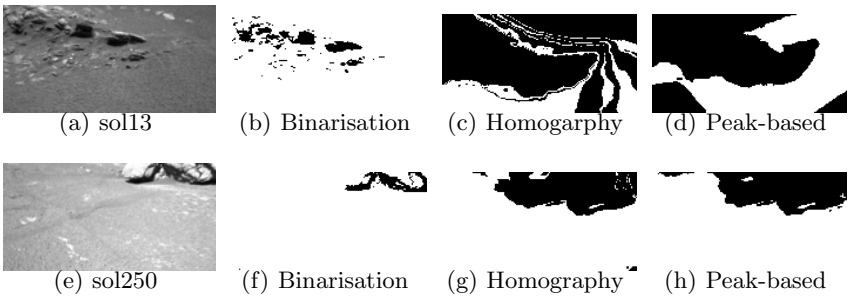
Table 2. Unification of dominant plane segments by three methods for Sol13 and Sol250. The entries are $|D_\alpha|/|D_k|$ for $\alpha\{H, B, P\}$ and $k = 1, 2, 3$.

Sol13	D_B	D_P	D_H	average	Sol250	1.00	0.80	0.80	0.87
D_1	0.99	0.34	0.51	0.61	D_1	1.00	0.80	0.80	0.8
D_2	0.63	0.71	0.88	0.74	D_2	0.82	0.98	0.99	0.93
D_3	0.23	0.87	0.71	0.60	D_3	0.78	0.99	0.98	0.92



(a) h_{tt} $\boldsymbol{v}$ Sol13 (b) h_{tt} $\boldsymbol{w}$ Sol13 (c) h_{tt} $\boldsymbol{v}$ Sol250 (d) h_{tt} $\boldsymbol{w}$ Sol250

Fig. 5. Second Time derivatives of the histograms of Sol13 and Sol250. The vertical line in each figure show the peak of the histogram. The horizontal line in each figure corresponds to the total variation of the optical flow vectors $\int_{\Omega} |\nabla \boldsymbol{u}| d\boldsymbol{x}$. The green curve in each figure shows the ration $|\Omega(k)|/|\Omega|$, where $\Omega(k)$ is the region corresponding to the k-th bin in the histogram.



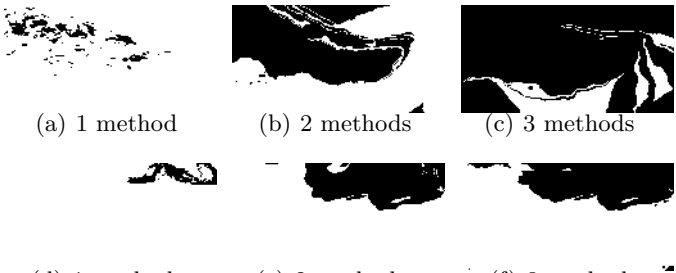
(a) sol13 (b) Binarisation (c) Homogarphy (d) Peak-based

(e) sol250 (f) Binarisation (g) Homography (h) Peak-based

Fig. 6. Results for Sol13 and Sol250. The top and bottom rows show results for Sol13 and Sol250, respectively. From left to right, original images and results computed by image binarisation, the peak detection and the homograpy-based method, are shown.

the LDOF and TOSR methods are both larger than those by the HSP method. In each frame, the histograms of the norm of the optical flow vectors computed by the HSP method are unimodal with wide divergence. Therefore, we cannot detect the dominant plane from the optical field. The histograms of the norm of the optical flow vectors computed by our method are, however, multimodal or with small divergence. If semantic planar segments exist in a frame, the optical flow vectors are stationary on this region. Then, the peaks appear in the histogram of the norm of th optical flow vectors. Comparison of the optical flow fields by two methods leads to the conclusion that the LDOF method can detect these peaks and that the HSP method fails to detect these peaks in both examples.

However, since as described in the previous sections, for the semantic segmentation of the dominant plane as a safe area in the workspace, we are required to have three successive frames of images. Therefore, we generate two successive optical flow fields from a pair of successive images using the temporal superresolution of the optical flow field. As shown in Fig. 4, we can extract a semantic segment using statistical bias of the optical flow vectors on the dominant plane.



(a) 1 method (b) 2 methods (c) 3 methods

(d) 1 method (e) 2 methods (f) 3 methods

Fig. 7. Unification of the results obtained by three methods. The top and bottom row are results for sol13 and sol250, respectively. From left to right, the unification results for one-, tow- and three methods, respectively.

Setting $h(\boldsymbol{u}(\boldsymbol{x}), t)$ for $\boldsymbol{x} = (x, y)^\top$to be histogram of $|\boldsymbol{u}(\boldsymbol{x})|$, the peaks t in the histogram are bins which satisfy the relations $h(\boldsymbol{u}(\boldsymbol{x}), t) > 2\text{average}_t h(\boldsymbol{u}(\boldsymbol{x}))$ and $h(\boldsymbol{u}(\boldsymbol{x}), t) > 2\text{average}_t h_{tt}(\boldsymbol{u}(\boldsymbol{x}))$. The peak-based segmentation detects the region $D_P = \{\boldsymbol{x} | t \in T \text{ for } h(\boldsymbol{u}(\boldsymbol{x}), t)\}$. Furthermore, we extract a segment $D_B = \{\boldsymbol{x} | f(x, y, t) > A\}$ by binarising grey-values of images using A computed by the Otsu threshold method [15].

Figure 6 shows the results. For the comparison, we also computed semantic segments using D_P and D_B. These results show that our proposing method, which combines the homgraphy and temporal optical flow superrealution, achieves semantic segmentation from a video sequence for navigation. Table 2 and Figure 7 show the results of the unification of the results obtained by three methods. Using the Boolean functions

$$f_H(\boldsymbol{x}) = \begin{cases} 1, \text{ if } \boldsymbol{x} \in D_H \\ 0, \text{ otherwise,} \end{cases} \quad f_B(\boldsymbol{x}) = \begin{cases} 1, \text{ if } \boldsymbol{x} \in D_B \\ 0, \text{ otherwise,} \end{cases} \quad f_P(\boldsymbol{x}) = \begin{cases} 1, \text{ if } \boldsymbol{x} \in D_P \\ 0, \text{ otherwise,} \end{cases} \tag{11}$$

we define the three regions,

$$D_1 = \{\boldsymbol{x} | f_H(\boldsymbol{x}) \vee f_B(\boldsymbol{x}) \vee f_P(\boldsymbol{x}) = 1\}, \tag{12}$$

$$D_2 = \{(f_H(\boldsymbol{x} \wedge f_B(\boldsymbol{x})) \vee (f_B(\boldsymbol{x}) \wedge f_P(\boldsymbol{x})) \vee (f(\boldsymbol{x}) \wedge f_H(\boldsymbol{x})) = 1\}, \tag{13}$$

$$D_3 = \{f_H(\boldsymbol{x}) \wedge f_B(\boldsymbol{x}) \wedge f_P(\boldsymbol{x}) = 1\}. \tag{14}$$

For $k = 1, 2, 3$, D_k is the collection of pixels which are categorised as elements of dominant plane by the k different algorithms. Then, we evaluate $|D_\alpha|/|D_k|$ for $\alpha\{H, B, P\}$ and $k = 1, 2, 3$, where $|D|$ is the area of region D, since the ground truths are not prepared for these image sequences. In Fig. (7), the top and bottom row are results for sol13 and sol250, respectively, and From left to right, the unification results for one-, tow- and three methods, respectively, are shown. These results show that unification of the three methods improves the results.

5 Conclusions

In this paper, we introduced a method for semantic segmentation of the dominant plane on an image sequence from a low-frame rate optical flow field.

In refs. [1] and [10], a dominant plane detection was achieved from a triplet of successive images. We have proposed a method for the detection of the dominant plane from low-frame rate image sequences using sub-frame optical flow computation.

References

1. Ohnishi, N., Imiya, A.: Featureless robot navigation using optical flow. Connection Science 17, 23–46 (2005)
2. Braillon, C., Pradalier, C., Crowley, J.L., Laugier, C.: Real-time moving obstacle detection using optical flow models. Intelligent Vehicles Symposium, 466–471 (2006)
3. Liang, B., Pears, N.: Visual navigation using planar homographies. In: IEEE International Conference on Robotics and Automation, pp. 205–210 (2002)
4. Young-Geun, K., Hakil, K.: Layered ground floor detection for vision-based mobile robot navigation. In: International Conference on Robotics and Automation, pp. 13–18 (2004)
5. Fischer, B., Modersitzki, J.: Ill-posed medicine- an introduction to image registration. Inverse Problem 24, 1–17 (2008)
6. Vardy, A., Moller, R.: Biologically plausible visual homing methods based on optical flow techniques. Connection Science 17, 47–89 (2005)
7. Beauchemin, S.S., Barron, J.L.: The computation of optical flow. ACM Computer Surveys 26, 433–467 (1995)
8. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artificial Intelligence 17, 185–204 (1991)
9. Hwang, S.-H., Lee, U.K.: A hierarchical optical flow estimation algorithm based on the interlevel motion smoothness constraint. Pattern Recognition 26, 939–952 (1993)
10. Ohnishi, N., Imiya, A.: Dominant plane detection from optical flow for robot navigation. Pattern Recognition Letters 27, 1009–1021 (2006)
11. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Comm. of the ACM 24, 381–395 (1991)
12. Varga, R.S.: Matrix Iteration Analysis, 2nd edn. Springer (2000)
13. Brox, T., Malik, J.: Large displacement optical flow: descriptor matching in variational motion estimation. IEEE Trans. PAMI 33, 500–513 (2011)
14. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
15. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Sys., Man., Cyber. 9, 62–66 (1979)

Analyses on Generalization Error of Ensemble Kernel Regressors

Akira Tanaka[1], Ichigaku Takigawa[2], Hideyuki Imai[1], and Mineichi Kudo[1]

[1] Division of Computer Science, Hokkaido University,
N14W9, Kita-ku, Sapporo, 060-0814 Japan
{takira,imai,mine}@main.ist.hokudai.ac.jp
[2] Creative Research Institution, Hokkaido University,
N21W10, Kita-ku, Sapporo, 001-0021 Japan
takigawa@cris.hokudai.ac.jp

Abstract. Kernel-based learning is widely known as a powerful tool for various fields of information science such as pattern recognition and regression estimation. For the last few decades, a combination of different learning machines so-called ensemble learning, which includes learning with multiple kernels, have attracted much attention in this field. Although its efficacy was revealed numerically in many works, its theoretical grounds are not investigated sufficiently. In this paper, we discuss regression problems with a class of kernels and show that the generalization error by an ensemble kernel regressor with the class of kernels is smaller than the averaged generalization error by kernel regressors with each kernel in the class.

Keywords: kernel regressor, ensemble learning, orthogonal projection, generalization error.

1 Introduction

Kernel-based learning machines [1], represented by the support vector machine [2] and the kernel ridge regressor [3], are widely recognized as powerful tools for various fields of information science such as pattern recognition and regression estimation. In general, an appropriate model selection is required in order to obtain a desirable learning result by kernel machines. Although the model selection in a fixed model space, such as selection of a regularization parameter, is sufficiently investigated in terms of theoretical and practical senses (see [4, 5] for instance), the selection of a model space is not sufficiently investigated in terms of a theoretical sense, while many practical algorithms for selection of a kernel (or its parameters) are proposed. In our previous works [6–9], we discussed the generalization error of a model space specified by a kernel and obtained some theoretical results.

For the last few decades, learning based on multiple kernels have attracted much attention in this field, which can be regarded as one of model selection schemes. The ensemble kernel learning (see [2] for instance), which is a combination of kernel-based learning machines, is representative one, whose theoretical

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 273–281, 2014.

grounds are not also investigated sufficiently. In this paper, we discuss generalization errors of the ensemble kernel regressor with a class of kernels and the kernel regressor using an individual kernel in the class; and show that the generalization error of the ensemble kernel regressor is smaller than the averaged generalization error of kernel regressors by each kernel.

2 Mathematical Preliminaries for the Theory of Reproducing Kernel Hilbert Spaces

In this section, we give some mathematical preliminaries concerned with the theory of reproducing kernel Hilbert spaces [10, 11].

Definition 1. *[10] Let $\mathbf{R}^n$ be an n-dimensional real vector space and let $\mathcal{H}$ be a class of functions defined on $\mathcal{D} \subset \mathbf{R}^n$, forming a Hilbert space of real-valued functions. The function $K(\boldsymbol{x}, \tilde{\boldsymbol{x}})$, $(\boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathcal{D})$ is called a reproducing kernel of $\mathcal{H}$, if the following two conditions hold.*

1. *For every $\tilde{\boldsymbol{x}} \in \mathcal{D}$, $K(\cdot, \tilde{\boldsymbol{x}})$ is a function belonging to $\mathcal{H}$.*
2. *For every $\tilde{\boldsymbol{x}} \in \mathcal{D}$ and every $f(\cdot) \in \mathcal{H}$,*

$$f(\tilde{\boldsymbol{x}}) = \langle f(\cdot), K(\cdot, \tilde{\boldsymbol{x}}) \rangle_{\mathcal{H}}, \tag{1}$$

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, abbreviated by RKHS. Eq.(1) is called the reproducing property of a kernel and it enables us to treat a value of a function at a point in $\mathcal{D}$. Note that reproducing kernels are positive definite [10]:

$$\sum_{i,j=1}^{N} c_i c_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0, \tag{2}$$

for any $N \in \mathbf{N}$, $c_1, \ldots, c_N \in \mathbf{R}$, and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathcal{D}$. In addition, $K(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = K(\tilde{\boldsymbol{x}}, \boldsymbol{x})$ holds for any $\boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathcal{D}$ [10]. If a reproducing kernel $K(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ exists, it is unique [10]. Conversely, every positive definite function $K(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ has the unique corresponding RKHS [10]. Hereafter, the RKHS corresponding to a reproducing kernel $K(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is denoted by $\mathcal{H}_K$. In the following contents, we simply use the symbol K for a kernel by omitting $(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ except the cases where it is needed.

Next, we introduce the Schatten product [12] that is a convenient tool to reveal the reproducing property of kernels.

Definition 2. *[12] 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*

$$(g \otimes h) f = \langle f, h \rangle_{\mathcal{H}_1} g, \quad f \in \mathcal{H}_1. \tag{3}$$

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 $h, v \in \mathcal{H}_1$, $g, u \in \mathcal{H}_2$.

$$(h \otimes g)^* = (g \otimes h), \quad (h \otimes g)(u \otimes v) = \langle u, g \rangle_{\mathcal{H}_2} (h \otimes v), \tag{4}$$

where the superscript * denotes the adjoint operator.

We give the following theorem concerned with the sum of reproducing kernels, which is used in the following contents.

Theorem 1. *[10] If K_i is the reproducing kernel of the class F_i with the norm $||\cdot||_i$, then $K = K_1 + K_2$ is the reproducing kernel of the class F of all functions $f(\cdot) = f_1(\cdot) + f_2(\cdot)$ with $f_i(\cdot) \in F_i$, and with the norm defined by*

$$||f(\cdot)||^2 = \min \left[||f_1(\cdot)||_1^2 + ||f_2(\cdot)||_2^2 \right], \tag{5}$$

the minimum taken for all the decompositions $f(\cdot) = f_1(\cdot) + f_2(\cdot)$ with $f_i(\cdot) \in F_i$.

The most important consequence of Theorem 1 is that the RKHS corresponding to $K = K_1 + K_2$ includes $\mathcal{H}_{K_1}$ and $\mathcal{H}_{K_2}$.

3 Formulation of Regression Problems

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

$$y_i = f(\boldsymbol{x}_i) + n_i, \tag{6}$$

where $f(\cdot)$ denotes the unknown true function and n_i denotes zero-mean additive noise. The aim of the regression problem is to estimate the unknown function $f(\cdot)$ by using the given training data set and statistical properties of the noise.

In this paper, we assume that the unknown function $f(\cdot)$ belongs to the RKHS $\mathcal{H}_K$ corresponding to a certain kernel K. If $f(\cdot) \in \mathcal{H}_K$, then Eq.(6) is rewritten as

$$y_i = \langle f(\cdot), K(\cdot, \boldsymbol{x}_i) \rangle_{\mathcal{H}_K} + n_i, \tag{7}$$

on the basis of the reproducing property of kernels. Let $\boldsymbol{y} = [y_1, \ldots, y_\ell]'$ and $\boldsymbol{n} = [n_1, \ldots, n_\ell]'$ with the superscript $'$ denoting the transposition operator, then applying the Schatten product to Eq.(7) yields

$$\boldsymbol{y} = \left(\sum_{k=1}^{\ell} [\boldsymbol{e}_k^{(\ell)} \otimes K(\cdot, \boldsymbol{x}_k)] \right) f(\cdot) + \boldsymbol{n}, \tag{8}$$

where $\boldsymbol{e}_k^{(\ell)}$ denotes the ℓ-dimensional unit vector whose k-th element is unity. For a convenience of description, we write

$$A_{K,X} = \left(\sum_{k=1}^{\ell} [\boldsymbol{e}_k^{(\ell)} \otimes K(\cdot, \boldsymbol{x}_k)] \right), \tag{9}$$

where $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\ell\}$. Note that $A_{K,X}$ is a linear operator from $\mathcal{H}_K$ onto $\mathbf{R}^\ell$ and Eq.(8) can be written by

$$\boldsymbol{y} = A_{K,X} f(\cdot) + \boldsymbol{n}, \tag{10}$$

which represents the relationship between the unknown true function $f(\cdot)$ and an output vector $\boldsymbol{y}$. Therefore, a regression problem can be interpreted as an inversion problem of the linear equation Eq.(10) [13].

4 Kernel Specific Generalization Error and Some Known Results

In general, a learning result by kernel machines is represented by a linear combination of $K(\cdot, \boldsymbol{x}_k)$, which implies that the learning result is an element in the range space of the linear operator $A^*_{K,X}$, written as $\mathcal{R}(A^*_{K,X})$, since

$$\begin{aligned} \hat{f}(\cdot) = A^*_{K,X}\boldsymbol{\alpha} &= \left(\sum_{k=1}^{\ell} [K(\cdot, \boldsymbol{x}_k) \otimes \boldsymbol{e}_k^{(\ell)}] \right) \boldsymbol{\alpha} \\ &= \sum_{k=1}^{\ell} \alpha_k K(\cdot, \boldsymbol{x}_k) \end{aligned} \tag{11}$$

holds, where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_\ell]'$ denotes an arbitrary vector in $\mathbf{R}^\ell$. The point at issue in this paper is to discuss goodness of a model space, that is, the generalization error of $\mathcal{R}(A^*_{K,X})$ which is independent from learning criteria. Therefore, we define the generalization error of kernel machines specified by a kernel K and a set of input vectors X as the distance between the unknown true function $f(\cdot)$ and $\mathcal{R}(A^*_{K,X})$ written as

$$J_{K_e}(f(\cdot); K, X) = ||f(\cdot) - P_{K,X} f(\cdot)||^2_{\mathcal{H}_{K_e}}, \tag{12}$$

where $P_{K,X}$ denotes the orthogonal projector onto $\mathcal{R}(A^*_{K,X})$ in $\mathcal{H}_K$ and $||\cdot||_{\mathcal{H}_{K_e}}$ denotes the induced norm of $\mathcal{H}_{K_e}$. Note that K_e may or may not be identical to K. Selection of an element in $\mathcal{R}(A^*_{K,X})$ as a learning result is out of the scope of this paper since the selection depends on learning criteria. We also ignore the observation noise in the following contents since the noise does not affect Eq.(12).

Here, we give some propositions as preparations to evaluate Eq.(12).

Lemma 1. *[6]*

$$P_{K,X} = \sum_{i,j=1}^{\ell} (G^+_{K,X})_{ij} \left[K(\cdot, \boldsymbol{x}_i) \otimes K(\cdot, \boldsymbol{x}_j) \right], \tag{13}$$

where $G_{K,X}$ denotes the Gramian matrix of K with X and the superscript $^+$ denotes the Moore-Penrose generalized inverse[14].

From Lemma 1, the orthogonal projection of $f(\cdot) \in \mathcal{H}_K$ onto $\mathcal{R}(A^*_{K,X})$ is given as

$$P_{K,X} f(\cdot) = \sum_{i,j=1}^{\ell} f(\boldsymbol{x}_i)(G^+_{K,X})_{ij} K(\cdot, \boldsymbol{x}_j). \tag{14}$$

5 Analyses on Generalization Error of Ensemble Kernel Regressors

We consider a class of kernels $\mathcal{K} = \{K_1, \ldots, K_n\}$ and corresponding RKHS written as $\mathcal{H}_{K_p}$, $(p \in \{1, \ldots, n\})$. We assume that

$$S = \overline{\cap_{p=1}^{n} \mathcal{H}_{K_p}} \tag{15}$$

forms a non-empty linear class.

In this section, we discuss the generalization error of the ensemble kernel regressor with the kernels in $\mathcal{K}$ and the mean of the generalization errors of the kernel regressors with each kernel in $\mathcal{K}$. For any $f(\cdot) \in S$, the learning result by the kernel regressor with K_p, $(p \in \{1, \ldots, n\})$ and the set of input vectors X is written as

$$\hat{f}_p(\cdot) = \sum_{i,j=1}^{\ell} f(\boldsymbol{x}_i)(G^+_{K_p,X})_{i,j} K_p(\cdot, \boldsymbol{x}_j) = P_{K_p,X} f(\cdot) \tag{16}$$

from Eq.(14); and its estimation error is reduced to

$$e_p(\cdot) = f(\cdot) - \hat{f}_p(\cdot). \tag{17}$$

Since $e_p(\cdot) \in \mathcal{H}_{K_p}$, we adopt an RKHS including all $\mathcal{H}_{K_p}$, $(p \in \{1, \ldots, n\})$ in order to evaluate the norm of $e_p(\cdot)$, $(p \in \{1, \ldots, n\})$ uniformly. As such an RKHS, we adopt $\mathcal{H}_{K_e}$ whose corresponding kernel is given as

$$K_e = \sum_{p=1}^{n} K_p \tag{18}$$

from Theorem 1. Therefore, the expected squared error of the kernel regressors with K_p, $(p \in \{1, \ldots, n\})$ and X, evaluated in $\mathcal{H}_{K_e}$, is reduced to

$$J_m = \frac{1}{n} \sum_{p=1}^{n} ||e_p(\cdot)||^2_{\mathcal{H}_{K_e}}. \tag{19}$$

On the other hand, the ensemble kernel regressor by the kernels in $\mathcal{K}$ is represented by

$$\hat{f}_e(\cdot) = \frac{1}{n} \sum_{p=1}^{n} P_{K_p,X} f(\cdot) = \frac{1}{n} \sum_{p=1}^{n} \hat{f}_p(\cdot), \tag{20}$$

and its generalization error, evaluated in $\mathcal{H}_{K_e}$, is reduced to

$$J_e = \|f(\cdot) - \hat{f}_e(\cdot)\|^2_{\mathcal{H}_{K_e}} = \left\| \frac{1}{n} \sum_{p=1}^{n} \left(f(\cdot) - \hat{f}_p(\cdot) \right) \right\|^2_{\mathcal{H}_{K_e}} = \left\| \frac{1}{n} \sum_{p=1}^{n} e_p(\cdot) \right\|^2_{\mathcal{H}_{K_e}}. \tag{21}$$

Note that the adopting the norm of $\mathcal{H}_{K_e}$ in the evaluation J_e is valid since $f(\cdot),\ \hat{f}_e(\cdot) \in \mathcal{H}_{K_e}$ obviously holds.

The following theorem is the main result of this paper.

Theorem 2

$$J_m - J_e = \frac{1}{2n^2} \sum_{p,q=1}^{n} \left\| \hat{f}_p(\cdot) - \hat{f}_q(\cdot) \right\|^2_{\mathcal{H}_{K_e}}. \tag{22}$$

Proof. Since $e_p(\cdot) - e_q(\cdot) = \hat{f}_q(\cdot) - \hat{f}_p(\cdot)$ holds, we have

$$\begin{aligned}
&J_m - \frac{1}{2n^2} \sum_{p,q=1}^{n} \left\| \hat{f}_p(\cdot) - \hat{f}_q(\cdot) \right\|^2_{\mathcal{H}_{K_e}} \\
&= \frac{1}{n} \sum_{p=1}^{n} \|e_p(\cdot)\|^2_{\mathcal{H}_{K_e}} - \frac{1}{2n^2} \sum_{p,q=1}^{n} \|e_p(\cdot) - e_q(\cdot)\|^2_{\mathcal{H}_{K_e}} \\
&= \frac{1}{n} \sum_{p=1}^{n} \|e_p(\cdot)\|^2_{\mathcal{H}_{K_e}} - \frac{1}{2n^2} \sum_{p,q=1}^{n} \left(\|e_p(\cdot)\|^2_{\mathcal{H}_{K_e}} + \|e_q(\cdot)\|^2_{\mathcal{H}_{K_e}} - 2\langle e_p(\cdot), e_q(\cdot) \rangle_{\mathcal{H}_{K_e}} \right) \\
&= \frac{1}{n^2} \sum_{p,q=1}^{n} \langle e_p(\cdot), e_q(\cdot) \rangle_{\mathcal{H}_{K_e}} = \left\| \frac{1}{n} \sum_{p=1}^{n} e_p(\cdot) \right\|^2_{\mathcal{H}_{K_e}} = J_e,
\end{aligned}$$

which concludes the proof. □

According to Theorem 2, it is concluded that the generalization error of the ensemble kernel regressor is smaller than or equal to the averaged generalization error of the kernel regressors by each kernel since the right-hand side of Eq.(22) is trivially non-negative, which implies that the ensemble kernel regressor gives a better performance than the kernel regressor by each kernel in terms of the averaged performance. Moreover, we can observe from Eq.(22) that when the learning results by the kernel regressors with each kernel are quite similar, the ensemble kernel regressor loses its advantage since the right-hand side of Eq.(22) becomes smaller. On the contrary, when the estimation error by the kernel regressors with each kernel, that is $e_p(\cdot)$, tend to be orthogonal each other, the generalization error of the ensemble kernel regressor J_e tends to be J_m/n from Eq.(21) and the Pythagorean theorem.

6 Example

In this section, we give a toy example confirming Theorem 2. Let $\mathcal{K} = \{K_1, K_2\}$ with

$$K_1(x, y) = 1 + xy + x^2y^2, \tag{23}$$
$$K_2(x, y) = 1 + xy + x^3y^3 \tag{24}$$

be a class of kernels. Note that a basis of $\mathcal{H}_{K_1}$ is $\{1, x, x^2\}$ and that of $\mathcal{H}_{K_2}$ is $\{1, x, x^3\}$. Therefore, we have

$$S = \overline{\mathcal{H}_{K_1} \cap \mathcal{H}_{K_2}} = \overline{\text{span}\{1, x\}}. \tag{25}$$

We adopt $f(x) = 1 + 2x \in S$ as an unknown true function and $X = \{0, 1\}$ as a set of input points. Therefore, we have

$$\hat{f}_1(\cdot) = P_{K_1,X} f(\cdot) = 1 + x + x^2, \tag{26}$$
$$\hat{f}_2(\cdot) = P_{K_2,X} f(\cdot) = 1 + x + x^3 \tag{27}$$

from Eq.(14) and

$$\hat{e}_1(\cdot) = f(x) - \hat{f}_1(x) = (1 + 2x) - (1 + x + x^2) = x - x^2, \tag{28}$$
$$\hat{e}_2(\cdot) = f(x) - \hat{f}_2(x) = (1 + 2x) - (1 + x + x^3) = x - x^3. \tag{29}$$

Let $K_e(x, y) = K_1(x, y) + K_2(x, y) = 2 + 2xy + x^2y^2 + x^3y^3$, then it is trivial that $\dim\mathcal{H}_{K_e} = 4$. Since $e_1(x), e_2(x) \in \mathcal{H}_{K_e}$, they can be expressed by linear combinations of four linearly independent functions in $\mathcal{H}_{K_e}$, such as

$$K_e(x, 0) = 2,$$
$$K_e(x, 1) = 2 + 2x + x^2 + x^3,$$
$$K_e(x, 2) = 2 + 4x + 4x^2 + 8x^3,$$
$$K_e(x, 3) = 2 + 6x + 9x^2 + 27x^3.$$

In fact, $e_1(x)$ and $e_2(x)$ are represented by

$$e_1(x) = \frac{1}{12}\left(-23K_e(x, 0) + 48K_e(x, 1) - 33K_e(x, 2) + 8K_e(x, 3)\right), \tag{30}$$
$$e_2(x) = \frac{1}{4}\left(-3K_e(x, 0) + 4K_e(x, 1) - 1K_e(x, 2)\right). \tag{31}$$

By the well known property $\langle K_e(\cdot, x), K_e(\cdot, y)\rangle_{\mathcal{H}_{K_e}} = K_e(x, y)$, we have

$$||e_1(\cdot)||^2_{\mathcal{H}_{K_e}} = ||e_2(\cdot)||^2_{\mathcal{H}_{K_e}} = \frac{3}{2}, \tag{32}$$

which immediately yields

$$J_m = \frac{1}{2}\left(||e_1(\cdot)||^2_{\mathcal{H}_{K_e}} + ||e_2(\cdot)||^2_{\mathcal{H}_{K_e}}\right) = \frac{3}{2}. \tag{33}$$

On the other hand, the learning result by the ensemble kernel regressor by K_1 and K_2 is reduced to

$$\hat{f}_e(x) = \frac{1}{2}(\hat{f}_1(x) + \hat{f}_2(x)) = 1 + x + \frac{x^2 + x^3}{2} \tag{34}$$

and we have

$$f(x) - \hat{f}_e(x) = x - \frac{x^2 + x^3}{2}, \tag{35}$$

which can be represented by

$$f(x) - \hat{f}_e(x) = \frac{1}{6}\left(-8K_e(x,0) + 15K_e(x,1) - 9K_e(x,2) + 2K_e(x,3)\right), \tag{36}$$

whose squared norm in $\mathcal{H}_{K_e}$ is

$$J_e = ||f(x) - \hat{f}_e(x)||^2_{\mathcal{H}_{K_e}} = 1. \tag{37}$$

Therefore, we have

$$J_m - J_e = \frac{1}{2}. \tag{38}$$

Accordingly, it is concluded that the ensemble kernel regressor gives a better result than the kernel regressor by the individual kernel K_1 or K_2 in these settings.

Finally, we evaluate the right-hand side of Eq.(22). Since

$$\begin{aligned} &\hat{f}_1(x) - \hat{f}_2(x) = x^2 - x^3 \\ &= \frac{1}{6}\left(7K_e(x,0) - 18K_e(x,1) + 15K_e(x,2) - 4K_e(x,3)\right) \end{aligned} \tag{39}$$

and $||\hat{f}_1(x) - \hat{f}_2(x)||^2_{\mathcal{H}_{K_e}} = 2$, the right-hand side of Eq.(22) is reduced to

$$\begin{aligned} &\frac{1}{2 \cdot 2^2} \sum_{p,q=1}^{2} ||\hat{f}_p(x) - \hat{f}_q(x)||^2_{\mathcal{H}_{K_e}} \\ &= \frac{1}{8}(||\hat{f}_1(x) - \hat{f}_2(x)||^2_{\mathcal{H}_{K_e}} + ||\hat{f}_2(x) - \hat{f}_1(x)||^2_{\mathcal{H}_{K_e}}) \\ &= \frac{1}{2}, \end{aligned} \tag{40}$$

which agrees with Eq.(38). Note that calculation of right-hand side of Eq.(22) does not require the unknown true function $f(\cdot)$, while the left-hand side of Eq.(22) includes $f(\cdot)$, which implies that we can obtain the reduced squared error by the ensemble kernel regressor only from the learning results by kernel regressors with each kernel.

7 Conclusion

In this paper, we discussed the generalization error of the ensemble kernel regressor with a class of kernels; and proved that the generalization error of the ensemble kernel regressor is smaller than the averaged generalization error of the kernel regressors by each kernel in the class. Similar analysis for the noise is one of our future works that should be undertaken.

Acknowledgment. This work was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 24500001.

References

1. Muller, K., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B.: An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks 12, 181–201 (2001)
2. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1999)
3. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000)
4. Sugiyama, M., Ogawa, H.: Subspace Information Criterion for Model Selection. Neural Computation 13, 1863–1889 (2001)
5. Sugiyama, M., Kawanabe, M., Muller, K.: Trading variance reduction with unbiasedness: The regularized subspace information criterion for robust model selection in kernel regression. Neural Computation 16, 1077–1104 (2004)
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., Miyakoshi, M.: Theoretical Analyses for a Class of Kernels with an Invariant Metric. In: 2010 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2010), pp. 2074–2077 (2010)
8. Tanaka, A., Imai, H., Kudo, M., Miyakoshi, M.: Theoretical Analyses on a Class of Nested RKHS's. In: 2011 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2011), pp. 2072–2075 (2011)
9. Tanaka, A., Takigawa, I., Imai, H., Kudo, M.: Extended analyses for an optimal kernel in a class of kernels with an invariant metric. In: Gimel'farb, G., Hancock, E., Imiya, A., Kuijper, A., Kudo, M., Omachi, S., Windeatt, T., Yamada, K. (eds.) SSPR&SPR 2012. LNCS, vol. 7626, pp. 345–353. Springer, Heidelberg (2012)
10. Aronszajn, N.: Theory of Reproducing Kernels. Transactions of the American Mathematical Society 68, 337–404 (1950)
11. 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)
12. Schatten, R.: Norm Ideals of Completely Continuous Operators. Springer, Berlin (1960)
13. Ogawa, H.: Neural Networks and Generalization Ability. IEICE Technical Report NC95-8, 57–64 (1995)
14. Rao, C.R., Mitra, S.K.: Generalized Inverse of Matrices and its Applications. John Wiley & Sons (1971)

Structural Human Shape Analysis for Modeling and Recognition

Chutisant Kerdvibulvech[1] and Koichiro Yamauchi[2]

[1] Rangsit University, 52/347 Muang-Ake, Paholyothin Rd, Lak-Hok, Patum Thani 12000, Thailand
chutisant.k@rsu.ac.th
[2] Keio University, 3-14-1 Hiyoshi, Kohoku-ku 223-8522, Japan
yamauchi@hvrl.ics.keio.ac.jp

Abstract. Structural human shape analysis is not a trivial task. This paper presents a novel method for a structural human shape analysis for modeling and recognition using 3D gait signatures computed from 3D data. The 3D data are obtained from a triangulation-based projector-camera system. To begin with, 3D structural human shape data which are composed of representative poses that occur during the gait cycle of a walking human are acquired. By using interpolation of joint positions, static and dynamic gait features are obtained for modeling and recognition. Ultimately, structural human shape analysis is achieved. Representative results demonstrate that the proposed 3D gait signatures based biometrics provides valid results on real-world 3D data.

Keywords: Structural Human Shape Analysis, Human Walking, 3D Human Body Model, Model Fitting, Modeling, 3D Recognition.

1 Introduction

Structural human shape analysis is an interesting topic to explore for many computer scientists around the world. This is because recognition based on human gait and structural human shape analysis has several advantages such as the acquisition of data at a distance from a non-cooperative subject and a characteristic biometrics signature that cannot be faked for a long time [1]. If the habit of walking is changed consciously, the motion seems unnatural and sooner or later a subject returns to his/her natural way of walking. In addition, gait involves not only surface shape, called static features of body parts, but also continuous motion of joints, called dynamic features. Nixon et al. [2] [3] introduced the concept of a total walking cycle according to which the action of walking is similarly assumed as a periodic signal. The gait cycle is assumed as the time interval. Each leg of human has two phases: swing period, when the foot is off the door moving forward to the next step, and stance period, when the foot is in contact with the floor. In the medical field, Murray et al. [4] introduced standard movement patterns of healthy subjects compared to disabled subjects pathologically. For data collection, required markers are attached to anatomical landmarks of human body. They advocate that the pelvic and thorax rotations are essentially variable from one subject to another. Moreover, agent-based simulation of human movement in street networks was discussed in [5] by Jiang and

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 282–290, 2014.

Jia. They conclude interestingly that the moving agents of human movement in large street networks are different in their moving behaviour in the fundamental aspects. For this reason, structural human shape analysis is indeed not trivial.

Recently, biometrics modalities with depth information are an interesting resource. As they can apply to many applications, range scanners have obviously become popular increasing the measurement accuracy and speed. It may be true that there are various approaches for 2D and 3D biometrics. Here, Multi-Cam indicates a single or multi-camera system and Pro-Cam indicates a projector-camera system. While biometrics approaches using 3D face, finger, ear, and their multimodal data have been presented, gait recognition methods still utilized video sequences. Thus, we attempt to tackle human recognition using 3D gait biometrics where both the modeling and the test data are obtained in 3D.

Yamauchi et al.'s work discussed about the possibility of recognition of a walking humans in [6]. However, only the initial results were proposed. In this paper, we present a structural human shape analysis method for recognition using 3D gait biometrics from a projector-camera system. A technique for person identification based on 3D body shape and gait is introduced. 3D human body data consisting of representative poses over one gait cycle are captured. 3D human shape model is fitted to the body data using a bottom-up approach. Since the body data is dense and it is at a high resolution, we can interpolate the entire gait sequence to fill-in between gait acquisitions. Gait features are usually considered by both dynamic features and static features. By using gait features, the similarity measure is applied for recognition of a subject and his/her pose.

The rest of this paper is clearly structured as follows. Section 2 introduces the framework of structural human shape analysis in detail. After that, in section 3, the experimental results are examined to ensure that the proposed 3D gait signatures based biometrics provides valid results on real-world 3D data. Section 4 ultimately concludes the paper and points to potential future research.

2 Structural Human Shape Analysis

To focus on structural human shape analysis, it is important to understand fundamentally the human gait. The truth is that gait consists of two distinct periods. First is a swing phase. This is a phase when the foot does not touch the ground moving the leg forward. Second is a stance phase. This phase is when the foot touches the ground. The gait cycle is expressed by the swing phase and the stance phase. The cycle begins with foot touch which marks the start of the swing phase. The body weight is transferred onto the other leg and the leg swings forward to meet the ground in front of the other foot. The cycle ends with the foot touch. The start of stance phase is when the heel strikes the ground. The ankle flexes to bring the foot flat on the ground and the body weight transferred onto it. The end of stance phase is when the heel leaves the ground.

We use the assumption that there are four measured poses. The model of the human body we used comes from a kinematic tree. The tree consists of 12 segments. The body segment is approximated by a 3D tapered cylinder which has one free parameter: the cylinder length. It has two degrees of freedom in rotational joints, in the local coordinate system. Upper torso is the root segment, i.e. the parent of lower torso, right upper leg, and left upper leg. Similarly, other segments are linked to parent segments by the rotational joints. The bounding angles of rotational joints are also important. The constraints are enforced in the form of bounding values of the joint angles. Within these

constraints the model has enough range of movement to represent various poses. The whole body is rotated around three axes and the other segments are rotated around two axes. Here, neck is the fixed segment between head and upper torso, so the neck joint angles are not considered. The articulated structure of the human body has a total of 40 degrees of freedom (DOFs). The pose is described by a 6-D vector, p, representing global position and rotation, a 22-D vector, q, representing the joint angles, and a 12-D vector, r, representing the lengths of body part. We denote s as the combination of the representative four poses. Joint DOF values concatenated along the kinematic tree define the kinematic pose as a tuple. One of the reasons why we use only 2-DOF rotational joints in this paper is since the 3D tapered cylinder has rotational symmetry along the direction orthogonal to the radial direction.

Figure 1 shows the method overview of structural human shape analysis. To begin with, we measure 3D human body data by a triangulation-based projector-camera system with human in the walking postures. We build a temporal structural body model, and then extract static dynamic features. Following these data collection we separate the human body data into six regions, and then 3D human body model is fitted to the segmented body parts in a top-down hierarchy from head to legs. The body model is refined by the Iterative Closest Point algorithm during the optimization process. The output of structural human shape analysis is successfully obtained to compare it against a database.

The intuition behind the principal component analysis (PCA) [7] is to find a set of basis vectors, so that they explain the maximum amount of variance of the data. PCA is applied to determine the coronal axis (*X*-axis), vertical axis (*Y* -axis), sagittal axis (*Z*axis), and centroid (*O*) of a human body in the world coordinate system (*O*-*X*-*Y* -*Z*), as described in [8]. The constraints are enforced in the form of bounding values of the joint angles. The whole body is rotated around three axes and the other segments are rotated around two axes. Here, neck is the fixed segment between head and upper torso, so the neck joint angles are not considered.

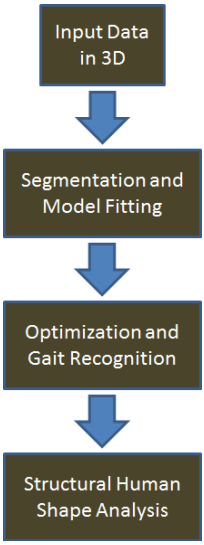


Fig. 1. Method overview of structural human shape analysis

The combination of the representative four poses is denoted by s. Joint DOF values concatenated along the kinematic tree define the kinematic pose, k, as a tuple, [p, q, r, s], where p ∈ R6, q ∈ R22, r ∈ R12, s = {s1, s2, s3, s4}. In the previous works, segments are linked to parent segments by either 1-DOF (hinge), 2-DOF (saddle) or 3-DOF (ball and socket) rotational joints [9]. We use only 2-DOF rotational joints, because the 3D tapered cylinder has rotational symmetry along the direction orthogonal to the radial direction. As a result, we eliminate the twist of body parts as an unnecessary variable.

In our proposed method, we first measure 3D structural human shape data by a triangulation-based projector-camera system with human in the walking postures. Following these data collection we separate the human body data into six regions, and then 3D human body model is fitted to the segmented body parts in a top-down hierarchy from head to legs. The body model is refined by the Iterative Closest Point algorithm during the optimization process. The human body data are segmented into six regions: head/neck, torso, right arm, left arm, right leg, and left leg. It can be written as $\mathbf{x}_0 = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}$. The subscript, reg, indicates the region number. Fig. 1 shows body axes, three segments, and six major regions which include a total of twelve body parts. Here, r. and l. indicate right and left, u. and l. indicate upper and lower (e.g. r. l. arm is right lower arm). In the following we present a fully automatic parts-based segmentation method.

To begin with, we measure 3D human body data by a triangulation-based projector-camera system with human in the walking postures. Following these data collection we separate the human body data into six regions, and then 3D human body model is fitted to the segmented body parts in a top-down hierarchy from head to legs. The body model is refined by the Iterative Closest Point algorithm during the optimization process. We first consider human body modeling. As discussed in [10], modeling methods usually fail when applied to real data. The real data captured by a projector-camera system have obviously some critical problems. For instance, the projector-camera system cannot cover well particular body parts. The groin region, axillary region, and the side of a human body are some examples. In this way, 3D points of the real data are dependently distributed as explained in [11]. Moreover, the body sways and deep color clothes have detrimental effects which appear as holes and gaps. To solve these mentioned problems, we present a modeling method for dealing with the problems occurring in real data. The proposed method to modeling a walking human incorporates two separate steps. First is model fitting, and second is optimization. The segmentation is useful for coarse registration, because it is unreasonable to fit a kinematic model to articulated objects without any prior knowledge. The prior knowledge allows automatic model fitting and the reduction in the computational cost. Therefore, we fit the model to body data by using the segmented regions. The distance between 3D data of a segmented region, $\mathbf{x}_{reg,j}$, and 3D model of the tapered cylinder, $\mathbf{y}_{i,j}$, is linearly minimized. The tapered cylinders can be fitted by determining two angles and one length in the order of levels 1, 2, 3 of the hierarchical structure. With regard to the head and neck, the parameters are estimated from the distribution of 3D points in the X-Y plane and Y-Z plane, respectively because the data for hair on the head and lower head region cannot be captured.

Next step is to get fine registration by minimizing the distance between the body data and kinematic model. The Iterative Closest Point algorithm [12] is utilized in our method. The key steps of the algorithm are: (a) Uniform sampling of points on both shapes. (b) Matching each selected point to the closest sample in the other shape. (c) Uniform weighting of point pairs. (d) Rejecting pairs with distances larger than some multiple of the standard deviation. (e) Point-to-point error metric. (f) Standard select-match minimizes iteration.

Obtaining the accurate results depends crucially on the type of gait features. Generally, gait features are divided into two types: (a) dynamic features and (b) static features. For example, the length of stride is one of the significant features of human gait. It can be computed by the leg length and its varying angles between poses. In addition, all the joint positions can be computed by using the same method. Therefore, both dynamic features and static features are used for recognition. We define the dynamic features as joint angles, qm, n, and static features as lengths of the body parts, rm, n. Here, m is the personal identification number, and n is the pose index. If dynamic features are only used, a feature vector is defined as Q[m, n]. Suppose that unknown feature vector, Qu, is one of M * N feature vectors, Q[m, n]. The minimum value of matching scores can be calculated as explained. The matching score is computed as distance. For unknown data, the personal identification number and pose index are recognized.

3 Experimental Results

The experiments were performed on the body data set collected by the human body measurement system. It contains twenty-four body data from the representative four poses of six subjects $X \in \{A,B,C,D,E, F\}$. The body data of representative poses are captured by a human body measurement system. The system consists of nine projector-camera pairs, which acquires nine range data in 2 seconds with 640×480 pixels, 3 mm depth resolution, and 3 mm measurement accuracy.

3D structural human shape data of walking humans of Subject A, Subject B, Subject C, Subject D, Subject E, and Subject F are depicted respectively. The number of measuring points is about 1/2 to one million depending on the subject and the pose. The full results of gait reconstruction are also experimented. It is defined that the one gait cycle is composed of twenty frames. The speed is given by dividing the stride length by the number of poses and the direction is not given automatically. The frame index 1, 6, 11, and 16 are representative poses, while the other frame indexes are calculated poses. The representative poses and their symmetric poses are used for the experiment. The poses are symmetric. They are synthesized by allocating right (or left) side parameters of representative poses to left (or right) side parameters of symmetrical poses. Figure 2 shows the results of human body modeling. Similarly, the six subjects of Subject A, Subject B, Subject C, Subject D, Subject E, and Subject F are used. The 3D human body model is fitted to four different walking poses. Thus, it contains a total of 24 different poses. The body model is suitably fitted to the captured body data. The joint angles are then obtained symmetrical poses, including the lengths of body parts.

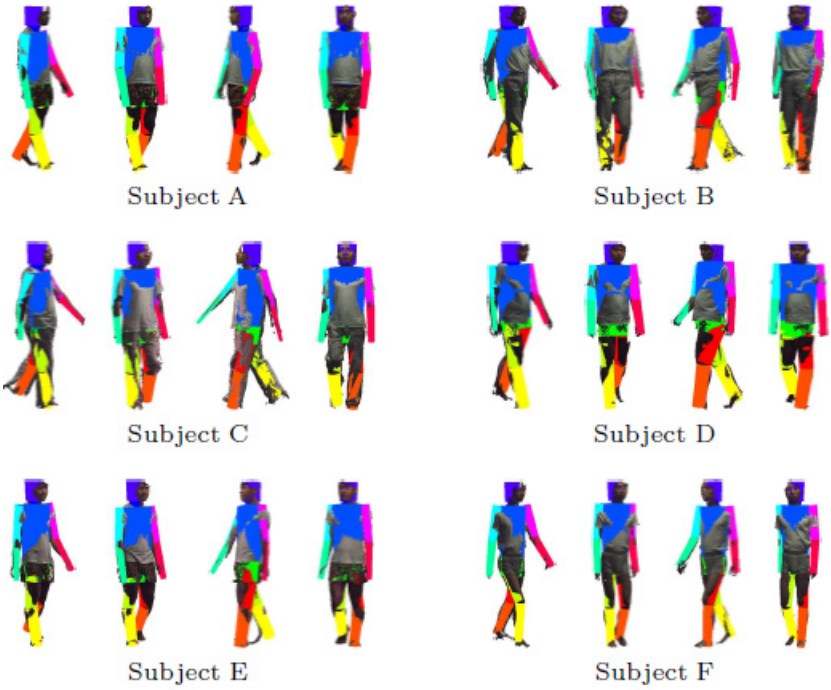


Fig. 2. 3D structural human shape data of walking humans fitted to four poses

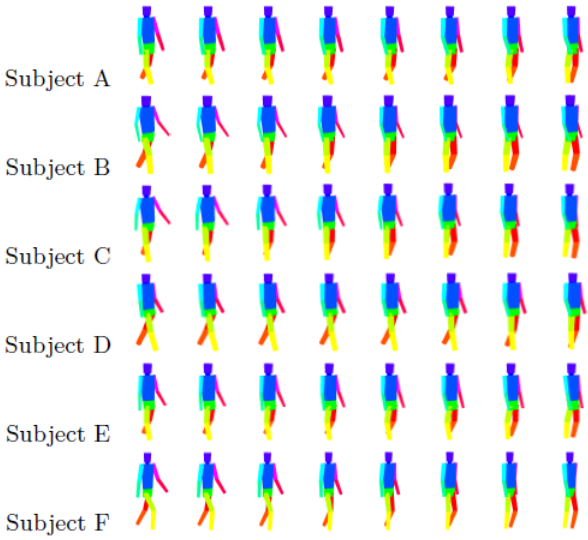


Fig. 3. Representative training data for six subjects

Figure 3 represents the training data of six subjects, i.e. Subject A, Subject B, Subject C, Subject D, Subject E, and Subject F. Each subject has 40 poses, so that training data contains a total of 240 kinematic poses. For the testing data, one gait sequence is recovered by representative poses = {s1, s2, s3, s4}. Average pose error is calculated in order to estimate the quantitative accuracy of the proposed method. The identification rate is obtained by dividing the number of recognized subjects by the number of testing data. The pose error is the frame difference between the estimated pose and the ideal pose. From our experiment, although bodies for these two subjects are different, their joint angles, i.e. leg and arm swings, are quite similar. For the average pose error we achieve 0.41 frame using dynamic features and 1.31 frames using both features. The experiment using dynamic features has better results, because it focuses on estimating poses, i.e. dynamic features do not consider lengths of body parts. However, by using both features, it definitely provides better results.

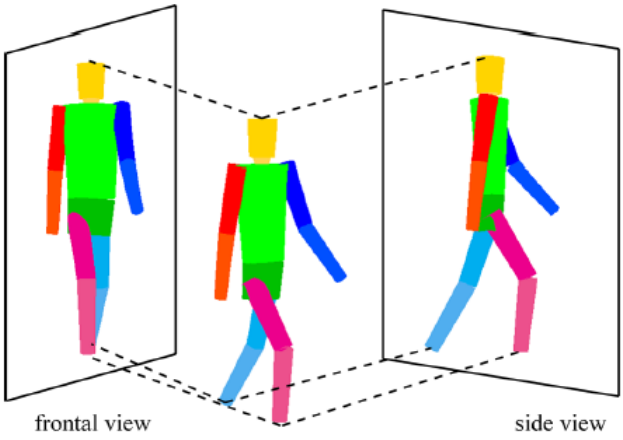


Fig. 4. Virtual front and side views

In fact, gait recognition methods usually use only a single view. This means that the information of 2D gait biometrics is solely used. Figure 4 shows how to compare 3D gait biometrics with 2D gait biometrics. Frontal view and side view are synthesized from our data by orthographic projection. We then use 2D gait biometrics for the experiment. From our experimental results, we achieve 95.83 percent using dynamic feature and 100 percent using both dynamic and static features for the identification rate. Furthermore, we achieve 0.57 using dynamic feature and 1.29 using both features for the average pose error. When only dynamic feature is used, the method fails to recognize testing data Subject B with pose 4, Subject D with pose 7, Subject D with pose 8, and Subject D with pose 14 who should not be recognized as the training data for Subject C with pose 4, Subject A with pose 7, Subject A with pose 8, and Subject B with pose 13.

Table 1. Average pose error for frontal view and side view

	Frontal View (%)	Side View (%)
Lower Average Pose Error for 3D Gait Biometrics	29 percent	72 percent

Furthermore, our experiments show that 3D gait biometrics provide lower average pose error. As shown in Table 1, it is 29 percent of the frontal view lower than that result of 2D gait biometrics. The number is also 72 percent of the side view lower than that of 2D gait biometrics. We believe that these numbers are suitable enough to make this proposed method newly useful for structural human shape analysis for modeling and recognition.

4 Conclusions

In this paper, an approach for a structural human shape modeling based on 3D gait biometrics and recognition where the input data are obtained from a triangulation-based projector-camera system is presented. By separating the human body into six regions, a bottom-up approach is utilized to fit the 3D human body. After that, the entire gait sequence is recovered. Finally, both static and dynamic gait features are obtained for recognition task. Our database we used consists of a total of twenty-four body data. In each data, it is comprised of four poses of six subjects. The size of database, we believe, is still quite small. We intend to expand our database to collect a huge number of subjects for possible better results. After that, we intend to apply the proposed method to use in related-applications. In fact, it is applicable to various kinds of computer science fields. We plan to refine this problem in the future.

References

1. Bhanu, B., Han, J.: Human Recognition at a Distance in Video. Springer, London (2011)
2. Nixon, M.S., Tan, T., Chellappa, R.: Human identification based on gait. Springer, New York (2005)
3. Nixon, M.S., Carter, J.N.: Automatic recognition by gait. Proc. of the IEEE 94(11), 2013–2024 (2006)
4. Murray, M.P., Drought, A.B., Kory, R.C.: Walking patterns of normal men. Journal of Bone and Joint Surgery 46A(2), 335–360 (1964)
5. Jiang, B., Jia, T.: Agent-based Simulation of Human Movement Shaped by the Underlying Street Structure. International Journal of Geographical Information Science 25(1) (2011)
6. Yamauchi, K., Bhanu, B., Saito, H.: Recognition of walking humans in 3D: Initial results. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), June 20-25, pp. 45–52 (2009)
7. Jolliffe, I.T.: Principal Component Analysis. Springer, New York (2002)

8. Kerdvibulvech, C., Saito, H.: Real-Time Guitar Chord Recognition System Using Stereo Cameras for Supporting Guitarists. Transactions on Electrical Engineering, Electronics, and Communications (ECTI) 5(2), 147–157 (2007)
9. Vondrak, M., Signal, L., Jenkins, O.C.: Physical simulation for probabilistic motion tracking. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)
10. Yu, H., Qin, S., Wight, D.K., Kang, J.: Generation of 3D human models with different levels of detail through point-based simplification. In: Proc. of the International Conference on "Computer as a Tool", pp. 1982–1986 (2007)
11. Werghi, N., Rahayem, M., Kjellander, J.: An ordered topological representation of 3D triangular mesh facial surface: concept and applications. EURASIP Journal on Advances in Signal Processing 1, 1–20 (2012)
12. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: Proc. of the 3-D Digital Imaging and Modeling, pp. 145–152 (2001)

On Cross-Validation for MLP Model Evaluation

Tommi Kärkkäinen

University of Jyväskylä,
Department of Mathematical Information Technology
tommi.karkkainen@jyu.fi

Abstract. Cross-validation is a popular technique for model selection and evaluation. The purpose is to provide an estimate of generalization error using mean error over test folds. Typical recommendation is to use ten-fold stratified cross-validation in classification problems. In this paper, we perform a set of experiments to explore the characteristics of cross-validation, when dealing with model evaluation of Multilayer Perceptron neural network. We test two variants of stratification, where the nonstandard one takes into account classwise data density in addition to pure class frequency. Based on computational experiments, many common beliefs are challenged and some interesting conclusions drawn.

Keywords: Cross-validation, Multilayer Perceptron, Model Selection.

1 Introduction

Cross-validation (CV) is a popular technique for model selection and evaluation, whose roots go back, at least, to 1960's [1]. The purpose of dividing data into independent training and test sets is to enable estimation of the generalization error of a trained model. This is especially important for the so-called universal approximators, of which Multilayer Perceptron neural network (MLP) is an example [2]. More precisely, with a real data set of input-output samples, a model that can represent unknown functions very accurately is prone to overlearning. Hence, its complexity should be determined using generalization as primary focus instead of training accuracy.

Kohavi [3] is probably the most well-known reference of CV in machine learning. Using six classification benchmarks with over 500 instances from UCI repository, two classification algorithms, C4.5 and naive Bayes, and amount of misclassifications in percentages as error indicator, stratified 10-CV (i.e., with ten folds) was concluded as the recommended approach. This approach has then established itself as a kind of community practice. For example, in [4] it is stated at page 153 (in relation to the amount of folds in CV): "Why 10? Extensive tests on numerous different datasets, with different learning techniques, have shown that 10 is about the right number of folds to get the best estimate of error, and there is also some theoretical evidence that backs this up." No references are given. At page 2980 in [5], which deals with nonlinear regression, it is stated that "Many simulation and empirical studies have verified that a reliable

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 291–300, 2014.

estimate of [generalization] Err can be obtained with $k = 10$ for $N > 100$ as recommended by Davison and Hinkley (1997)." The precise argument by Davison and Hinkley [6] at page 294, in the context of linear regression, is to take $k = \min\{N^{1/2}, 10\}$ due to *practical experience*: "taking $k > 10$ may be too computationally intensive... while taking groups of size at least $N^{\frac{1}{2}}$ should perturb the data sufficiently to give small variance of the estimate." No computational experiments are performed to support the claim.

Actually it has been observed in many articles that use of CV is not straightforward. In [7] it is shown, in the least-squares-estimation context, that for an unstable procedure the predictive loss, i.e., difference between "crystal ball" and cross-validation based selection, is large. The difficulties of using (10-)CV for the MLP model selection were already addressed in [8]: With fourteen UCI benchmark data sets the experiments showed that CV is only slightly better than random selection of MLP architecture and that the smallest size of the hidden layer tested provided almost the same generalization performance than the repeated folding. Based on experiments with the same UCI data sets as Kohavi [3], for ID3 and Info-Fuzzy Network classifiers with 2-CV, [9] ended up with "CV uncertainty principle": "the more accurate is a model induced from a small amount of real-world data, the less reliable are the values of simultaneously measured cross-validation estimates." Finally, in [10] large and general review of CV is given. Among the overall conclusions it is stated that i) usually CV overestimates generalization error compared to training error, ii) CV method with minimal variance [of generalization error estimate] seems strongly framework-dependent, and iii) the issue of "optimal" amount of folds in CV is not straightforward.

Hence, the purpose of this paper is to perform a set of experiments to explore the characteristics of cross-validation, when dealing with model evaluation of MLP. We test two variants of stratification, where the new approach takes into account classwise data densities [11] in addition to pure class frequency. To simplify the analysis, we restrict to ten folds. The contents are as follows: in Section 2 we summarize the model, the learning problem, and the actual algorithms. Then, in Section 3, a sequence of computational experiments with observations and subconclusions is presented. Finally, general conclusions are summarized in Section 4.

2 Methods and Algorithms

2.1 MLP

Action of MLP in a layerwise form, with given input vector $\mathbf{x} \in R^{n_0}$, can be formalized as follows [12]: $\mathbf{o}^0 = \mathbf{x}, \quad \mathbf{o}^l = \mathcal{F}(\mathbf{W}^l \tilde{\mathbf{o}}^{(l-1)})$ for $l = 1, \ldots, L$. Here the layer number has been placed as an upper index and by $\tilde{}$ we indicate the vector enlargement to include bias. This places these nodes in a layer as first column of the layer's weight matrix which then has the factorization $\mathbf{W}^l = \left[\mathbf{W}_0^l \ \mathbf{W}_1^l\right]$. $\mathcal{F}(\cdot)$ denotes the application of activation functions. We restrict to networks with one hidden layer so that the two unknown weight matrices are $\mathbf{W}^1 \in \mathbb{R}^{n_1 \times (n_0+1)}$

Algorithm 1. Determination of neural network model using cross-validation

Input: Data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$.
Output: Feedforward neural network.
Define $\boldsymbol{\beta}$, $n1max$, $nfolds$, and $nits$
for $n_1 \leftarrow 2$ **to** $n1max$ **do**
  for $regs \leftarrow 1$ **to** $|\boldsymbol{\beta}|$ **do**
    Create $nfolds$ using cross-validation
    for $k \leftarrow 1$ **to** $nfolds$ **do**
      for $i \leftarrow 1$ **to** $nits$ **do**
        Initialize $(\mathbf{W}^1, \mathbf{W}^2)$ from $\mathcal{U}([-1, 1])$
        Minimize (1) with current n_1 and $\boldsymbol{\beta}(regs)$ over kth training set
        Store network for smallest training set error
      Compute test_error over kth test set for the stored network
    Store network for the smallest mean{test_error}

and $\mathbf{W}^2 \in \mathbb{R}^{n_2 \times (n_1+1)}$. Using the given learning data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathbb{R}^{n_0}$ and $\mathbf{y}_i \in \mathbb{R}^{n_2}$, determination of weights is realized by minimizing the cost functional

$$\mathcal{J}(\mathbf{W}^1, \mathbf{W}^2) = \frac{1}{2N} \sum_{i=1}^{N} \|\mathbf{e}_i\|^2 + \frac{\beta}{2n_1} \sum_{i,j} \left(|\mathbf{W}^1_{i,j}|^2 + |(\mathbf{W}^2_1)_{i,j}|^2 \right) \tag{1}$$

for $\mathbf{e}_i = \mathbf{W}^2 \tilde{\mathcal{F}}(\mathbf{W}^1 \tilde{\mathbf{x}}_i) - \mathbf{y}_i$ and $\beta \geq 0$. The linear second layer and the special form of regularization omitting the bias-column $\mathbf{W}^2_0$ are due to Corollary 1 in [12]: For every locally optimal MLP-network with the cost functional (1), satisfying $\nabla_{\mathbf{W}^2} \mathcal{J} = \mathbf{0}$, the average error $\frac{1}{N} \sum_{i=1}^N \mathbf{e}_i$ is zero. Hence, every locally $\mathbf{W}^2$-optimal network provides un unbiased nonlinear estimator for the learning data, independently on the regularization coefficient β.

The actual determination of MLP is documented in Algorithm 1. The main point is to realize systematic grid search over the complexity landscape, determined by n_1 (size of network) and β (size of weights; the larger β the closer to zero). Hence, $n1max$ determines largest size of the hidden layer and predefined values in vector $\boldsymbol{\beta} = \{\beta_r\}$ define possible regularization coefficients. Due to preliminary testing, we use here $\beta_r = 10^{-r}, r = 2, \ldots, 6$. Moreover, with the fixed parameters we always create new folds to sample the CV approaches below. The parameter $nits$ determines the amount of local restarts with random generation/initialization of weights from the uniform distribution. This is the simplest globalization strategy when minimization of (1) will be done locally.

2.2 Two Folding Approaches for CV

We apply two stratification strategies for folding: the standard random creation where class frequencies of the whole training data are approximated in folds (SCV). As the second approach, DOB-SCV (Distribution Optimally Balanced Standard CV) as proposed in [11] was implemented, see Algorithm 2. In this

Algorithm 2. Distribution Optimally Balanced Standard CV (DOB-SCV)

Input: Data $(\mathbf{X}, C) = \{\mathbf{x}_i, c_i\}_{i=1}^{N}$ of inputs and class labels and amount of folds k.
Output: k non-disjoint folds $F_l, l = 1, \ldots, k$, such that $\mathbf{X} = \cup_{l=1}^{k} F_l$.

```
for each class j and input data X_j = {x_i | c_i = j} do
    while |X_j| ≥ k do
        Let x_1 be random observation from X_j
        Let x_2, ..., x_k be k − 1 closest neighbors of x_1 from X_j
        Let F_l = F_l ∪ {x_l} and X_j = X_j \ {x_l}, l = 1, ..., k
    Place the remaining observations from X_j into different folds F_l, l = 1, ..., |X_j|
```

approach, using the division of a random observation from class j and its $k-1$ nearest class neighbors to different folds, classwise densities in addition to frequencies are approximated in all the folds. We remind that in [11,13] the extensive experimentation on various data sets and classifiers did not include MLP as classifier, not to mention the particular optimization problem (1) that we solve here.

3 Computational Experiments

All methods described in the previous section were implemented and tested on MATLAB (R2013b running on 64-bit Windows 7). For SCV, *cvpartition* routine is used. Minimization of (1) is based on MATLAB's unconstrained minimization routine *fminunc*, using layerwise sensitivity calculus from [12] for computing gradients. Standard sigmoid $s(x) = 1/(1 + \exp(-x))$ is used as the activation function. All input variables are preprocessed into the range $[0, 1]$ of $s(x)$ to balance the overall scaling of unknowns [12]. Class encoding is realized in the well-known manner by using standard basis in $\mathbf{R}^{n_2}$: the lth unit vector is used as target output for an input $\mathbf{x}_i$ from class C_l.

As benchmark data we use "Segmentation" from UCI repository, which is multiclass ($n_2 = 7$ classes) and many-input ($n_0 = 17$ input variables when two nearly constant ones are omitted) data set with small training set "Sgm (Train)" and large, separate validation set "Sgm (Test)". These sets are documented in Table 1. In what follows, we use the term Training error, TrE, for the mean error which is computed over the training sets, i.e. the subsets of "Sgm(Train)" without the test folds. Similarly, Test error TsE refers to mean error over test folds. With Generalization error GeE, we refer to the error which is computed using the validation set "Sgm(Test)".

Table 1. UCI classification data sets for CV experiments

Name	N	Class frequencies	Comments
Sgm (Train)	210	[30 30 30 30 30 30 30]	Features 3–4 removed
Sgm (Test)	2100	[300 300 300 300 300 300 300]	Features 3–4 removed

Table 2. 10-CV results for misclassification rate in percentages as error measure

10-SCV	10-SCV	3x10-SCV
7/1e-5/3.5/12.4(6.4)	12/1e-3/6.0/11.9(7.9)	11/1e-3/6.0/13.3(8.3)
10-DOB-SCV	10-DOB-SCV	3x10-DOB-SCV
7/1e-5/4.2/13.8(4.2)	9/1e-3/6.7/12.4(4.0)	6/1e-6/4.7/12.9(5.8)

3.1 Misclassification Rate in Percentages as Error Measure

In Table 2 first set of results using SCV and DOB-SCV with Algorithm 1 are given. We use $nits = 2$ and $n1max = 12$. In the results, n_1^* and β^* for the smallest TsE, its standard deviation Std, and the corresponding TrE are given. The actual result format is thus n_1^*/β^*/TrE/TsE(Std). For both folding approaches the algorithm is first tested two times separately, and then three times repeated folding for fixed n_1 and β is performed so that the errors are then computed over 30 training and test set errors. As error measure the misclassification rate in percentages is used.

From Table 2 one notices very high unstability of the results. Training and Test errors are very different, best parameters between tests vary a lot, standard deviations are large (round 30%–65% of means) and they do not decrease when folding is reiterated. For DOB-SCV, Stds are typically smaller compared to SCV, but there is no real difference in TsEs. Altogether one ends up with high uncertainty with these results, especially when the relationship between Training and Test errors, visualized using scatter plots in Fig. 1, is taken into account. The quantized form of the discrete error measure does not allow accurate evaluation of MLP models with different complexity, which is reflected as high variability in parameter choices.

3.2 Testing Predictive Error Measures

Next we test whether the discrete approximation of classification error could be one reason for difficulties with CV. Instead of misclassification rate in percentages, we test two error measures which are typical for estimating the actual prediction error:

$$e_{\mathrm{MR}} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\sum_{j=1}^{n_2}(\mathcal{N}(\mathbf{x}_i)-\mathbf{y}_i)_j^2} \quad \text{(Mean-Root-Squared-Error)},$$

$$e_{\mathrm{RM}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n_2}(\mathcal{N}(\mathbf{x}_i)-\mathbf{y}_i)_j^2} \quad \text{(Root-Mean-Squared-Error)}.$$

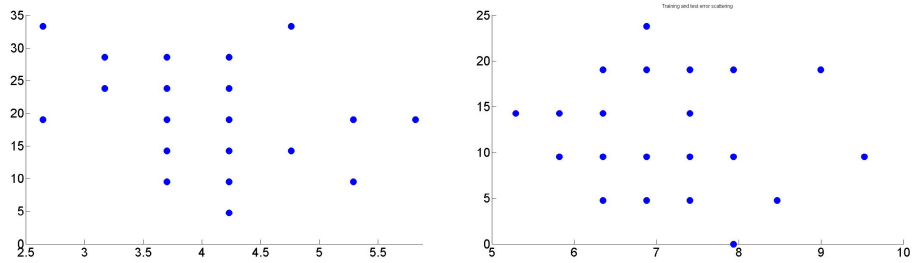


Fig. 1. Training/Test error scatterings for 3xSCV with $n_1 = 7$ and $\beta = 10^{-5}$ (left) and for 3xDOB-SCV with $n_1 = 9$ and $\beta = 10^{-3}$ (right)

Even if the definitions are very close to each other, the amount of observations weights the rest of the error measure differently ($1/N$ in e_{MR} compared to $1/\sqrt{N}$ in e_{RM}) and we want to find out how this affects comparisons of Training, Test, and Generalization errors which are computed with data sets of different sizes.

In Fig. 2 scatter plots of training and test set errors for SCV are given for all locally optimal MLPs obtained with Algorithm 1 for $n_1 = 5, \ldots, 8$ and $nits = 2$. It is visually clear that e_{MR} reflects the positive correlation between the two errors in a better way than e_{RM}. The visual appearance and the conclusion are precisely the same for DOB-SCV.

Using e_{MR} as CV error measure in MLP model evaluation in Algorithm 1, we obtain the following choices of parameters using the same grid search as in Table 2: for SCV, $n_1 = 5$ and $\beta = 10^{-5}$ and for DOB-SCV, $n_1 = 7$ and $\beta = 10^{-5}$. We then fix these and reiterate the two folding approaches three times with $nits = 5$. The individual results and their grand mean over different foldings are documented in Table 3.

From Table 3 we conclude that Training error underestimates and Test error overestimates Generalization error. For different foldings, SCV results are this time more stable than those of DOB-SCV. However, there is one remarkable difference in the characteristics of the results.

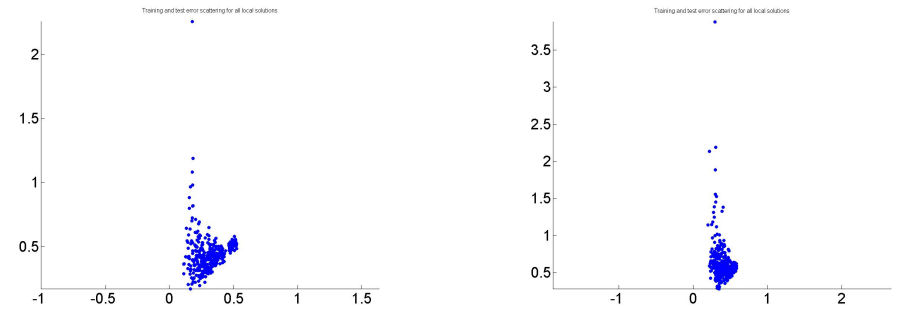


Fig. 2. Training/test set error scatterings with SCV: e_{MR} (left) and e_{RM} (right)

Table 3. Repeated CV with e_{MR} as error measure

	3xSCV			3xDOB-SCV		
Fold	TrE	TsE	GeE	TrE	TeE	GeE
1st	0.3065	0.3532	0.3443	0.1908	0.3421	0.3439
2nd	0.3063	0.3618	0.3480	0.1976	0.4478	0.3497
3rd	0.3080	0.3552	0.3497	0.1906	0.4852	0.3816
Grand	0.3069	0.3567	0.3473	0.1930	0.4250	0.3584

Namely, for one particular folding, one observation from the original training data belongs to exactly one test set due to disjoint division. Hence, for one observation the maximum amount of false test classifications over the three foldings is precisely three. Next, for 3xSCV with $n_1 = 5$ and $\beta = 10^{-5}$ and for 3xDOB-SCV with $n_1 = 7$ and $\beta = 10^{-5}$ we checked their classwise behavior in this respect, i.e. report the amount of indices per class where this maximum of three false test classifications per one observation is reached:

SCV: [1 0 4 29 3 1 0] = 38 cases,
DOB-SCV: [1 0 4 7 3 3 0] = 18 cases.

Hence, for SCV the pure inside-class randomness can produce high variability between classwise test accuracies (because the test folds can be very different from each other) whereas DOB-SCV compensates this dramatically better, through and due to distributional balancing. Notice that in the mean accuracy estimates without separating the classes, such behavior is completely hidden, as witnessed in Tables 2 and 3. We also conclude that class 4 is the most difficult one, so that to improve the classification performance more observations from that class should be contained in the training data.

3.3 Predictive CV with Modified Data Sets

To this end, we remove 30 random observations of class 4 from the original "Sgm(Test)" and add them to "Sgm(Train)". The previous stepwise experimentation is repeated as follows: i) With three repetitions apply the grid search for n_1 and β with $nits = 2$ using e_{MR} as error measure (cf. Table 2), ii) Compute mean errors with the chosen parameters with three repetitions of foldings (cf. Table 3) and visually assess an error scattering plot, and iii) check the classwise error rates from the test folds.

Result of Step i) is given in Table 4. We have obtained much higher stability in the parameter choice for both folding approaches. Also standard deviations are smaller compared to mean errors (around 20%–25% of means). Still, Training errors deviate from Test errors.

Table 4. Best parameters and errors for 10-CV with modified data sets

10-SCV	10-SCV	3x10-SCV
6/1e-6/0.21/0.36(0.07)	6/1e-6/0.25/0.37(0.10)	6/1e-5/0.28/0.37(0.07)
10-DOB-SCV	10-DOB-SCV	3x10-DOB-SCV
7/1e-6/0.19/0.34(0.09)	6/1e-5/0.28/0.37(0.08)	6/1e-5/0.27/0.36(0.08)

As for Step ii), we fix the parameters according to Table 4 as $n_1^* = 6$ and $\beta^* = 5 \cdot 10^{-6}$ for SCV and $n_1^* = 6$ and $\beta^* = 10^{-5}$ for DOB-SCV. The result with these choices for repeated foldings are given in Table 5.

We conclude from Table 5, especially compared to Table 3, that the increase of the size of training set from 210 to 240, that yielded to increase of the size of the hidden layer by one for SCV, then increased the differences between Training and Test errors. The common trend of Generalization error underestimation by Training error and overestimation by Test error remains. With the choices of parameters, slightly smaller Generalization errors are this time obtained with DOB-SCV compared to SCV. Now, for TrE's and TsE's the behavior of two folding approaches is similar.

Scatter plots for training and validation set errors with the two approaches are depicted in Figure 3. We see that both folding approaches yield to positive correlation between these errors, with DOB-SCV capturing such a desired behavior slightly better.

To this end, for the three repetitions in Table 5 and taking into account only those cases where an observation was always wrongly classified in a test set, we obtained the following amount of misclassifications per class:
SCV: [1 0 3 8 5 6 0] = 23 cases,
DOB-SCV: [1 0 4 6 7 8 0] = = 26 cases.
We conclude that the modifications of training and validation sets paid off, especially for SCV, by means of improved classwise balance of the classification accuracy and significantly smaller overall misclassification rate. For DOB-SCV, the amount of complete misclassifications increased significantly, from 18 into 26, because the emphasis on class 4 had negative effect on accuracies in classes 5 and 6. The smaller amount obtained by SCV does not imply superiority over DOB-SCV but just the fact that the result was obtained with more flexible model, i.e. with slightly smaller β^*.

Table 5. Repeated CV with modified sets

	3xSCV			3xDOB-SCV		
Fold	TrE	TsE	GeE	TrE	TeE	GeE
1st	0.2724	0.3754	0.3570	0.2728	0.4008	0.3597
2nd	0.2688	0.4080	0.3723	0.2795	0.3804	0.3570
3rd	0.2712	0.3840	0.3658	0.2743	0.3728	0.3402
Grand	0.2708	0.3891	0.3650	0.2755	0.3847	0.3532

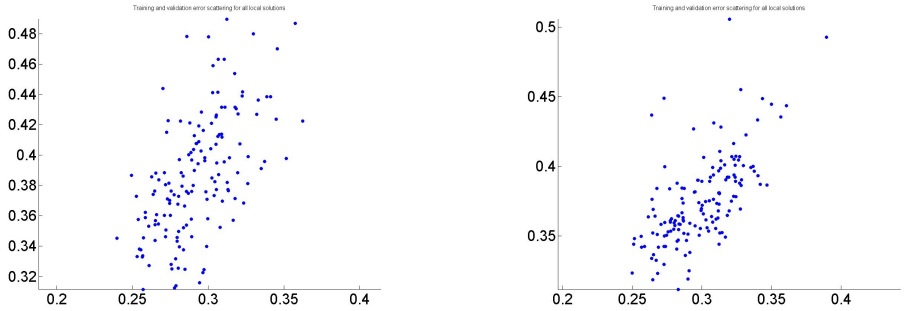


Fig. 3. Scatter plots of training/validation set errors for 3xSCV (left) and 3xDOB-SCV (right) with modified data sets

4 Conclusions

The performed set of experiments illustrate some difficulties related to model assessment with cross-validation. The general statistical assumptions on fixed input and input-output conditional distributions are not necessarily valid with real data sets. The amount of folds and the actual folding strategy have an effect on the behavior of CV. The error measure used with different data sets (training, test, validation) affects error computations and, hence, the form of obtained relationships underlying model selection. Especially when a universal prediction model, like MLP, is used in classification with typical output encoding, a discrete and quantized error measure suppress the precious information reflecting the quality of the model. In any case, estimation of the generalization error through test folds is only an approximation, and with all the experiments and techniques used here, we always ended up to overestimate the true generalization error using mean over ten test folds. Similarly, the standard deviation of generalization error estimate can remain large and does not necessarily decrease with repeated folding. We conclude, by comparing the Std estimates obtained with different parameters (typically with simpler network - with smaller size of hidden layer or larger regularization coefficient - we end up with smaller variance), that this estimate reflects more the variability of the model itself instead of one model's actual classification performance. Such an observation might be valid for universal approximators in general. We also illustrated that the quality of data has an effect on cross-validation results, especially when using the standard, stratified CV.

Through all the computational experiments performed we found that DOB-SCV folding approach could be better suited for real data sets, because it potentially provides better differentiation of a classifier's true performance through more homogenous test folds. This conclusion coincides with the results in [11] that were obtained with other classifiers and for larger set of folding approaches. Moreover, if classwise deviations in accuracy are revealed, one can, with sample data sets, augment the training data set accordingly or, in real applications,

launch a new data collection campaign to improve the overall classification performance. The findings here are obtained with only one benchmark data set with $k = 10$ folds, so further experiments with larger sample of real data sets and different amount of folds should be carried out in the future.

References

1. Elisseeff, A., Pontil, M.: Leave-one-out error and stability of learning algorithms with applications. NATO Science Series, Sub Series III: Computer and Systems Sciences 190, 111–130 (2003)
2. Pinkus, A.: Approximation theory of the MLP model in neural networks. Acta Numerica, 143–195 (1999)
3. Kohavi, R.: Study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 1995), pp. 1137–1143 (1995)
4. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann, Burlington (2011)
5. Borra, S., Ciaccio, A.D.: Measuring the prediction error. a comparison of cross-validation, bootstrap and covariance penalty methods. Computational Statistics and Data Analysis 54, 2976–2989 (2010)
6. Davison, A.C., Hinkley, D.V.: Bootstrap Methods and their Applications. Cambridge Series on Statistical and Probabilistic Mathematics. Cambridge University Press (1997)
7. Breiman, L.: Heuristics of instability and stabilization in model selection. The Annals of Statistics 24(6), 2350–2383 (1996)
8. Andersen, T., Martinez, T.: Cross validation and MLP architecture selection. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN 1999), pp. 1614–1619 (1999)
9. Last, M.: The uncertainty principle of cross-validation. In: Proceedings of the IEEE International Conference on Granular Computing (GrC 2006), pp. 275–280 (2006)
10. Arlot, S., Celisse, A.: A survey of cross-validation procedures for model selection. Statistics Surveys 4, 40–79 (2010)
11. Moreno-Torres, J.G., Sáez, J.A., Herrera, F.: Study on the impact of partition-induced dataset shift on k-fold cross-validation. IEEE Transactions on Neural Networks and Learning Systems 23(8), 1304–1312 (2012)
12. Kärkkäinen, T.: MLP in layer-wise form with applications to weight decay. Neural Computation 14, 1451–1480 (2002)
13. López, V., Fernández, A., Herrera, F.: On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed. Information Sciences 257, 1–13 (2014)

Weighted Mean Assignment of a Pair of Correspondences Using Optimisation Functions*

Carlos Francisco Moreno-García and Francesc Serratosa

Universitat Rovira I Virgili,
Departament d'Enginyeria Informàtica I Matemàtiques, Spain
carlosfrancisco.moreno@estudiants.urv.cat
francesc.serratosa@urv.cat

Abstract. Consensus strategies have been recently studied to help machine learning ensure better results. Likewise, optimisation in graph matching has been explored to accelerate and improve pattern recognition systems. In this paper, we present a fast and simple consensus method which, given two correspondences of sets generated by separate entities, enounces a final consensus correspondence. It is based on an optimisation method that minimises the cost of the correspondence while forcing it (to the most) to be a weighted mean. We tested our strategy comparing ourselves with the classical minimum cost matching system, using a palmprint database, with each palmprint is represented by an average of 1000 minutiae.

Keywords: Consensus strategy, Hamming Distance, Weighted Mean, Assignment Problem, Optimisation.

1 Introduction

When two subjects decide to solve the assignment problem, differences on the points' mapping may occur. These differences appear due to several factors. Between them, we could cite the following. One of the subjects gives more importance to some of the point attributes and the other subject believes other ones are more important. For instance, if the sets of points represent regions of segmented images, one subject may think the area is more important than the colour, and the other one can think it is the opposite. If the assignment problem is solved by an artificial system, the fact of "believing" the area is more important than the colour is gauged by some weights. Another factor could be that the assignment problem is computed in a suboptimal algorithm, and different non-exact assignments can appear. In these scenarios, a system can intervene as a third party to decide the final assignment as a consensus of both assignments since some discrepancies will appear, especially as the number of involved points increase.

This paper presents a method to find the consensus assignment between two sets given two different assignments between those sets. We model the consensus

* This research is supported by the Spanish CICYT project DPI2013-42458-P and Consejo Nacional de Ciencia y Tecnologías (CONACyT Mexico).

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 301–311, 2014.

assignment as the weighted mean assignment. This method is inspired in the one presented in [1] that obtains a clustering consensus given a set of clusterings. Other methods to perform this task are [2], where a final cluster is obtained based on a similarity graph and [3], where the least square algorithm is used. Our method, and also the one in [1], does not restrict the consensus assignment to be a strict mean but a weighted mean. This occurs because these methods aim to find an assignment (or clustering in [1]) that it is as closer as possible to both assignments (clusterings in [1]), but have to minimise the assignment cost (or the clustering cost in [1]). These methods are closely related to the unsupervised machine learning methods [4].

The drawback of this approach resides in the large number of possible solutions. One of the most well-known and practical options to reduce the complexity of a combinatorial calculation is combinatorial optimisation. The concept of optimisation is related to the selection of the "best" configuration or set of parameters to achieve a certain goal [5]. Functions involved in an optimisation problem can be either conformed by continuous values or discrete values, often called "Combinatorial Scenarios". These second scenarios have been largely studied and applied for graph matching problems, particularly in the case of the Hungarian Algorithm [6]. This method converts a combinatorial problem into an assignment problem, which will eventually derive in an optimal configuration for a cost-based labelling. Many recent researches have used graph theory and optimisation to solve diverse problems. Examples can be found in [7], where a graph representation and an optimisation method helped to design a gas drainage system for a coal mine. On [8], a research group used graph representation of newspaper articles to optimize the arrangement of each article within the page. Energy reduction in machinery [9] and most recently, biomedical compounds represented as labelled graphs [10] have been classified by using optimisation methods.

2 Basic Definitions

Given a set of elements $\mathrm{G}^1 = \{\mathrm{g}_1^1, \mathrm{g}_2^1, \ldots, \mathrm{g}_\mathrm{u}^1\}$, where the elements posses $\mathrm{g}_\mathrm{i}^1 = (\mathrm{m}_\mathrm{i}^1, \mathrm{a}_\mathrm{i}^1)$, being $\mathrm{m}_\mathrm{i}^1 \in \Sigma$ (where Σ is a unique number of the elements) and $\mathrm{a}_\mathrm{i}^1 \in \mathrm{T}$ (where T is the domain of the attribute of the elements), a labelling function f can be established between G^1 and another set of elements with similar characteristics G^2. This labelling function f is understood as a bijective function that proposes a match $f: \Sigma \rightarrow \Sigma$ from one element of G^1 to one element of G^2, where G^1 and G^2 have similar cardinality u.

We define the cost of a labelling $Cost(G^1, G^2, f)$ as the addition of individual element costs in a similar way as in the Graph Edit Distance [11],

$$Cost(f) = \sum_{i=1}^{u} c\left(a_i^1, a_j^2\right) \quad being\ j\ such\ that\ f(m_i^1) = m_j^2 \tag{1}$$

where c is defined as a distance function over the domain of attributes T and is application dependent [11].

The distance between sets $d_S(\cdot)$, which also delivers the minimum cost of all the labellings, is a function defined as

$$d_S(G^1, G^2) = min\{Cost(G^1, G^2, f)\}\ \forall\ f: \Sigma^1 \rightarrow\ \Sigma^2 \tag{2}$$

The labelling that obtains this distance is known as the optimal labelling f^*, and it is defined as

$$f^* = argmin_{\forall f: \Sigma^1 \rightarrow \Sigma^2}\{Cost(G^1, G^2, f)\} \tag{3}$$

We convert this linear minimisation problem into an assignment problem [6], for which any labelling f is related with a combination. With the calculation of a cost matrix $\boldsymbol{C}[i,j] = Cost(\mathrm{a}_\mathrm{i}^2, \mathrm{a}_\mathrm{j}^2)$, we can convert equation 3 into

$$f^* = argmin_{\forall f: \Sigma^1 \rightarrow \Sigma^2}\{\mathbf{C}_f\} \tag{4}$$

where $\mathbf{C}_f$ is the cost of the combination$f$ (or labelling in the set domain) applied to matrix $\boldsymbol{C}$. That is

$$\mathbf{C}_f = \sum_{i=1}^{\mathrm{u}} \boldsymbol{C}[i,k]\ where\ f(m_i^1) = m_k^2 \tag{5}$$

Assume f^a and f^b are two labelling functions between sets $\mathrm{G}^1 = \{\mathrm{g}_1^1, \mathrm{g}_2^1, \ldots, \mathrm{g}_\mathrm{u}^1\}$ and $\mathrm{G}^2 = \{\mathrm{g}_1^2, \mathrm{g}_2^2, \ldots, \mathrm{g}_\mathrm{u}^2\}$. We then define the Hamming Distance $\mathrm{d}_\mathrm{H}(\cdot)$ between the labellings $\mathrm{f}^\mathrm{a}$and f^b as

$$d_H(f^a, f^b) = \sum_{i=1}^{u}\left(1 - \partial\left(m_x^2, m_y^2\right)\right) \tag{6}$$

being x and y such that $\mathrm{f}^\mathrm{a}(\mathrm{m}_\mathrm{i}^1) = \mathrm{m}_\mathrm{x}^2$ and $\mathrm{f}^\mathrm{b}(\mathrm{m}_\mathrm{i}^1) = \mathrm{m}_\mathrm{y}^2$. Function ∂ is the well-known function known as the Kronecker Delta.

$$\partial(\mathrm{a},\mathrm{b}) = \begin{cases} 0 & \text{if } \mathrm{a} \neq \mathrm{b} \\ 1 & \text{if } \mathrm{a} = \mathrm{b} \end{cases} \tag{7}$$

In its more general form, the mean of two elements e^a and e^b has been defined as an element $\bar{\mathrm{e}}$ such that $\mathrm{d}(\mathrm{e}^\mathrm{a}, \bar{\mathrm{e}}) = \mathrm{d}\left(\mathrm{e}^\mathrm{b}, \bar{\mathrm{e}}\right)$ and $\mathrm{d}\left(\mathrm{e}^\mathrm{a}, \mathrm{e}^\mathrm{b}\right) = \mathrm{d}(\mathrm{e}^\mathrm{a}, \bar{\mathrm{e}}) + \mathrm{d}\left(\bar{\mathrm{e}}, \mathrm{e}^\mathrm{b}\right)$, being d any distance measure defined on the domain of these elements. Moreover, the weighted mean is sometimes used to gauge the importance or the contribution of the involved elements. In this case, the most general definition is $\mathrm{d}(\mathrm{e}^\mathrm{a}, \bar{\mathrm{e}}) = \alpha$ and $\mathrm{d}\left(\mathrm{e}^\mathrm{a}, \mathrm{e}^\mathrm{b}\right) = \alpha + \mathrm{d}\left(\bar{\mathrm{e}}, \mathrm{e}^\mathrm{b}\right)$ where α is a constant that controls the contribution of the elements and holds $0 \leq \alpha \leq \mathrm{d}\left(\mathrm{e}^\mathrm{a}, \mathrm{e}^\mathrm{b}\right)$. Finally, if we do not need to introduce the weighting factor α in our model, we have that any element $\bar{\mathrm{e}}$ is a weighted mean of two elements e^a and e^b if it holds that $\mathrm{d}\left(\mathrm{e}^\mathrm{a}, \mathrm{e}^\mathrm{b}\right) = \mathrm{d}(\mathrm{e}^\mathrm{a}, \bar{\mathrm{e}}) + \mathrm{d}\left(\bar{\mathrm{e}}, \mathrm{e}^\mathrm{b}\right)$. Note that elements e^a and e^b hold this condition so, they are also weighted means of themselves.

The most appropriate form to model a consensus scenario given two different options is the one done by [4], which is defined through the weighted mean of these two options. The aim of [4] is to find the consensus clustering of a set of elements given two different clustering proposals applied to this set of elements. If we want to translate this model to our problem, we should find the weighted mean labelling $\bar{\mathrm{f}}$ given two different labelling f^{a} and f^{b}. As commented in the previous paragraph, if we want $\bar{\mathrm{f}}$ to be defined as a weighted mean labelling of f^{a} and f^{b} the following restriction has to hold,

$$d_H(f^a, f^b) = d_H(f^a, \bar{f}) + d_H(\bar{f}, f^b) \tag{8}$$

Several labellings $\bar{f}$ hold this condition, and amongst them are f^a and f^b. To select one, an option would be a brute force method that obtains all possible combinations and selects the best one from the application point of view. Another option is a standard minimisation approach, to reduce the computational time.

Standard minimisation approaches aim to find an optimal element e^* that globally minimises a specific function. Usually this function is composed of an empirical risk $\nabla(\mathrm{e})$ plus a regularization term $\Omega(\mathrm{e})$ weighted by a parameter λ [5]. The empirical risk is the function to be minimised per se and the regularisation term is a mathematical mechanism to impose some restrictions. Parameter λ weights how much these restrictions have to be imposed.

$$e^* = \mathrm{argmin}_{\forall e} \{\nabla(e) + \lambda \cdot \Omega(e)\} \tag{9}$$

The aim of this paper is to present a method to find an approximation of the weighted mean labelling given two labellings. Therefore, we want to find $\bar{f}^*$ such that the following equation holds,

$$\bar{f}^* = \mathrm{argmin}_{\forall f: \Sigma^1 \to \Sigma^2} \{\lambda_C \cdot \nabla(f) + \lambda_H \cdot \Omega(f)\} \tag{10}$$

On the next section, we explain functions $\nabla(f)$ and $\Omega(f)$. Although it is not strictly necessary, we present this general equation with parameters λ_C and λ_H, instead of only one parameter λ as in equation 9, to simplify some explanations and examples.

3 Method

Our method defines the optimal labelling $\bar{f}^*$ through equation 10 in which the Loss function and the Regularisation term are

$$\nabla(f) = Cost(G^1, G^2, f) \text{ and } \Omega(f) = d_H(f^a, f) + d_H(f, f^b) - d_H(f^a, f^b) \tag{11}$$

That is, we want to minimise the labelling cost (equation 1) of the obtained labelling but restricted to be a weighted mean (equation 8). The degree of restriction depends on weights λ_C and λ_H. Note that by definition of a distance, $d_H(f^a, f) + d_H(f, f^b) - d_H(f^a, f^b) \geq 0$.

The aim of our method is to decide the labelling closer to both human's labellings. Therefore it seems logical that our strategy only seeks for the partial labelling where both of the specialists disagree. The other partial labelling, which is the one that both specialist has decided the same point mapping, is directly assigned as the mappings of these specialists. For this reason we split labellings f^a and f^b in two disjoint partial labellings such that $f^a = f'^a \cup f''^a$ and $f^b = f'^b \cup f''^b$, where f'^a and f'^b are the partial labellings where $f^a(m_i^1) = f^b(m_i^1)$, and f''^a and f''^b are the other partial ones where $f^a(m_i^1) \neq f^b(m_i^1)$. This also means that the cost of both labellings is $\mathrm{Cost}(G^1, G^2, f^a) = \mathrm{Cost}(G^1, G^2, f'^a) + \mathrm{Cost}(G^1, G^2, f''^a)$ and $\mathrm{Cost}(G^1, G^2, f^b) = \mathrm{Cost}(G^1, G^2, f'^b) + \mathrm{Cost}(G^1, G^2, f''^b)$. We define $\Sigma = \Sigma' \cup \Sigma''$. The set of nodes Σ' in G^1 is composed of the nodes such that $f^a(m_i^1) = f^b(m_i^1)$ and the set of nodes Σ'' in G^1 is composed of the nodes such that $f^a(m_i^1) \neq f^b(m_i^1)$.

Thus, we define the weighted mean labelling $\bar{f}^*$ we want to obtain as a union of two partial labellings, $\bar{f}^* = \bar{f}'^* \cup \bar{f}''^*$ where $\bar{f}'^* = f'^a$ (which is the same than $\bar{f}'^* = f'^b$) and $\bar{f}''^*$ is the one defined in the following equation,

$$\bar{f}''^*_{\lambda_C,\lambda_H} = \mathrm{argmin}_{\forall f'':\, \Sigma''^1 \to \Sigma''^2} \{ \lambda_C \cdot \mathrm{Cost}(G^1, G^2, f'') + \lambda_H \cdot \Big(d_H\big(f''^a, f''\big) + d_H\big(f'', f''^b\big) - d_H\big(f''^a, f''^b\big) \Big) \} \quad (12)$$

To solve equation 12, we translate the linear minimisation problem to an assignment problem [5] as we have shown in equation 4, but instead of the cost matrix $\boldsymbol{C}$, our method minimises matrix H_{λ_C,λ_H} defined as follows,

$$H_{\lambda_C,\lambda_H} = \lambda_C \cdot C'' + \lambda_H \cdot [\mathbf{1} - F''^{a,b}] \quad (13)$$

where $C''[i,j] = c(m_i^1, m_j^2)$ being $m_i^1 \in \Sigma''^1$ and $m_j^2 \in \Sigma''^2$. Besides, $F''^{a,b} = F''^a + F''^b$, where F''^a and F''^b are the labelling matrices corresponding to f''^a and f''^b, respectively. Aditionally, $\mathbf{1}$ is a matrix of all ones. Note that the number of rows and columns of matrices C'', F''^a and F''^b is lower or equal than C. As more similar node mappings of f^a and f^b are, the smaller the number nodes in Σ''^1 and Σ''^2 is, and so, the dimensions of C'', F''^a and F''^b. This fact affects directly on the computational cost. In a practical application, if both specialists are good enough, they discern in few node mappings and therefore the computational time of finding the agreement labelling is very low. Considering equation 13, we obtain the following expression,

$$\bar{f}''^*_{\lambda_C,\lambda_H} = \mathrm{argmin}_{\forall f''} \left\{ \left(H_{\lambda_C,\lambda_H} \right)_{f''} \right\} \quad (14)$$

Several algorithms can be used to minimise equation 14, for instance the Hungarian algorithm [5]. Finally, the cost of the obtained weighted mean becomes,

$$C_{\bar{f}^*_{\lambda_C,\lambda_H}} = C''_{\bar{f}''^*_{\lambda_C,\lambda_H}} + Cost\big(G^1, G^2, f'^a\big) \quad (15)$$

On section 3.1 we demonstrate equations 12 and 14 minimise at the same approximation of the weighted mean labelling $\bar{f}''^*_{\lambda_C,\lambda_H}$ for all weights λ_C and λ_H and

pair of graphs G^1 and G^2. Then on section 3.2 we demonstrate the cases in which the obtained labelling is an exact weighted mean labelling and not an approximated weighted mean labelling.

3.1 Reasoning about Optimality

If we want to use equation 14 to solve our problem instead of equation 12, we must now demonstrate that functional $\left\{\lambda_C \cdot \text{Cost}(G^1, G^2, f'') + \lambda_H \cdot \left(d_H(f''^a, f'') + d_H(f'', f''^b) - d_H(f''^a, f''^b)\right)\right\}$ extracted from equation 12 minimises the same partial labelling than $\left\{\left[\lambda_C \cdot C'' + \lambda_H \cdot [\mathbf{1} - F''^{a,b}]\right]_{f''}\right\}$ extracted from equation 14. Notice that, by definition, $Cost(G^1, G^2, f'') = C''_{f''}$ and for this reason, we have to demonstrate the following equation

$$[\mathbf{1} - F''^{a,b}]_{f''} = d_H(f''^a, f'') + d_H(f'', f''^b) - d_H(f''^a, f''^b); \forall f'': \Sigma''^1 \rightarrow \Sigma''^2 \quad (16)$$

If equation 16 holds, then we can confirm that it is valid to use equation 14 to solve our problem. Suppose the cardinality of $\Sigma''^1$and Σ''^2 is n. Therefore, by definition of these sets, $d_H(f''^a, f''^b) = n$. Given the involved labellings f'', f''^a and f''^b, we can define the three following natural numbers n_p, n_q and n_t:

1) n_p: number of nodes in Σ''^1that hold $f''(m_i^1) \neq f''^a(m_i^1)$ and $f''(m_i^1) \neq f''^b(m_i^1)$.
2) n_q: number of nodes in Σ''^1that hold $f''(m_i^1) = f''^a(m_i^1)$ and $f''(m_i^1) \neq f''^b(m_i^1)$.
3) n_t: number of nodes in Σ''^1that hold $f''(m_i^1) \neq f''^a(m_i^1)$ and $f''(m_i^1) = f''^b(m_i^1)$.

Again, by definition of these sets, there is not any m_i^1 such that $f''(m_i^1) = f''^a(m_i^1)$ and $f''(m_i^1) = f''^b(m_i^1)$. Therefore, $n = n_p + n_q + n_t$. By simplicity of notation, we order the nodes in Σ''^1 such that m_1^1 to $m_{n_p}^1$ hold the first condition, $m_{n_p+1}^1$ to $m_{n_p+n_q}^1$ hold the second condition and $m_{n_p+n_q+1}^1$ to m_n^1 hold the third condition.

To demonstrate that equation 16 holds, we first demonstrate that $[\mathbf{1} - F''^{a,b}]_{f''} = n_p$ and we second demonstrate that $d_H(f''^a, f'') + d_H(f'', f''^b) - d_H(f''^a, f''^b) = n_p$.

1) Demonstration of $[\mathbf{1} - F''^{a,b}]_{f''} = n_p$: Suppose that $f''(m_i^1) = m_k^2$ then $[\mathbf{1} - F''^{a,b}]_{f''} = \sum_{i=1}^{n}(\mathbf{1} - F''^{a,b})[i,k] = \sum_{i=1}^{n_p} 1 + \sum_{i=n_p+1}^{n} 0 = n_p$.
2) Demonstration of $d_H(f''^a, f'') + d_H(f'', f''^b) - d_H(f''^a, f''^b) = n_p$: $d_H(f''^a, f'') + d_H(f'', f''^b) - n = \sum_{i=1}^{n_p}\left(2 - \partial\left(f''^a(m_i^1), f''(m_i^1)\right) - \partial\left(f''(m_i^1), f''^b(m_i^1)\right)\right) - n = \left(\sum_{i=1}^{n_p} 0 + \sum_{i=n_p+1}^{n_p+n_q} 1 + \sum_{i=n_p+n_q+1}^{n} 1\right) - n = n_q + n_t - n = n_p$.

■

3.2 Exact Weighted Mean Labelling

In some cases, it is interesting to know if we have obtained an exact weighted mean labelling or an approximated one. First of all, we have to realise that whether both f' and f'' are exact partial weighted mean labellings, then the union $f = f' \cup f''$ is an exact weighted mean labelling, assuming that that $\mathrm{f}' \cap \mathrm{f}'' = 0$. It is clear that if equation 8 holds for both partial labellings then it holds for the complete one. Moreover, by definition of our partial labelling f', it is always defined as weighted mean labelling. Therefore, we conclude that the obtained labelling $\bar{\mathrm{f}}^*$is an exact weighted mean labelling if $\bar{\mathrm{f}}''^*$ is also an exact weighted mean labelling. These cases are the ones that that $\bar{\mathrm{f}}''^*$ holds equation 8. Due to we have demonstrated that $d_H(f''^a, f'') + d_H(f'', f''^b) - d_H(f''^a, f''^b) = \mathrm{n_p}$, then $\mathrm{n_p}$ has to be 0. By definition of $\mathrm{n_p}$, these labelling are the ones that $\bar{f}''^*(m_i^1) = f''^a(m_i^1)$ or $\bar{f}''^*(m_i^1) = f''^b(m_i^1)$. Therefore, we conclude the following expression,

$$\bar{\mathrm{f}}^*_{\lambda_C,\lambda_H} \text{is a weighted mean labelling if:}$$
$$\bar{\mathrm{f}}''^*_{\lambda_C,\lambda_H}(\mathrm{m_i^1}) = \mathrm{f}''^{\mathrm{a}}(\mathrm{m_i^1}) \text{ or } \bar{\mathrm{f}}''^*_{\lambda_C,\lambda_H}(\mathrm{m_i^1}) = \mathrm{f}''^{\mathrm{b}}(\mathrm{m_i^1}); \forall \mathrm{m_i^1} \in \Sigma''^1 \qquad (17)$$

The cost of testing if the labelling obtained is a weighted mean is linear on the number of discordances between labellings f^a and f^b.

Note that if $\lambda_C = 0$ and $\lambda_H > 0$ and we use the Hungarian method [6] to solve equation 14, then the method always obtains an exact weighted mean labelling. This is because equation 16 has been demonstrated, and also because the optimality of the Hungarian method has been demonstrated in equation 17.

4 Experimentation

We used images contained in the Tsinghua 500 PPI Palmprint Database [12]. It is a public high-resolution palmprint database composed of 500 palmprint images of a resolution of 2040 x2040 pixels. From each person, 8 palmprints are enrolled. We extracted the minutiae set of the 8 palmprints of the first 10 subjects of the database using the algorithm presented in [13], [14] and [15] and we obtained an average of 1000 minutiae per palmprint. The attributes of minutiae are their position and angle, which means $\mathrm{a_i^s} = \{\theta_\mathrm{i}^\mathrm{s}, (\mathrm{x}, \mathrm{y})_\mathrm{i}^\mathrm{s}\}$. Therefore, our core database is composed by 80 sets of minutiae classified in 10 subjects. Nevertheless, we wish to have a database of several registers and each register composed of four elements: 2 minutiae sets G^1 and G^2 extracted from the same palm and two different labellings f^a and f^b between these minutiae sets. To do so, we matched each of the 8 minutiae sets of the same subject obtaining 64 correspondences per subject. Therefore, we defined an initial database of 8x8x10=640 registers composed of 2 minutiae sets G^1 and G^2 extracted from the same subject and a correspondence $\check{\mathrm{f}}$ between them. Correspondences were

computed through the Hungarian method [16] and a greedy method to select the matches from the resulting matrix. The distance between minutiae has been defined to be

$$c\left(a_i^1, a_j^2\right) = 0.5 \cdot ad\left(\theta_i^1, \theta_j^2\right) + 0.5 \cdot dd\left((x,y)_i^1, (x,y)_j^2\right) \quad (18)$$

being *ad* the angular distance and *dd* the Euclidean distance.

Given each of the 640 registers, we need to generate labellings f^a and f^b from the initial labellings $\check{f}$. Nevertheless, to perform our experiments, we need to control the distance between these labellings. Therefore, we introduce parameter α that decides the Hamming distance between them, $d_H(f^a, f^b) = 2\alpha$. $f^a$and f^b are randomly generated such that $d_H\left(f^a, \check{f}\right) = \alpha$ and $d_H\left(f^b, \check{f}\right) = \alpha$. As we will see later, parameter α is the horizontal axis of the figures presented in this section and for each $\alpha \in \{10,11,\dots,212\}$we have a dataset of 640 registers, so, the values presented in these figures are the average of 640 times we computed $\bar{f}^*_{\lambda_C,\lambda_H}$.

The aim of our method is to find the consensus labelling with the minimum cost and close to both labellings. For this reason, we have performed several tests using different configurations of λ_C and λ_H and parameter α. The aim of these tests is threefold. First, we want to know the cost of the obtained labellings $\bar{f}^*_{\lambda_C,\lambda_H}$, that is $Cost\left(\bar{f}^*_{\lambda_C,\lambda_H}\right)$. Second, we want to analyse if the obtained labellings appear to be "in the middle" of both labellings. In this case, we propose the following measure,

$$\text{Middle}\left(f^a, f^b, \bar{f}^*_{\lambda_C,\lambda_H}\right) = \frac{\left|d_H\left(f^a, \bar{f}^*_{\lambda_C,\lambda_H}\right) - d_H\left(f^b, \bar{f}^*_{\lambda_C,\lambda_H}\right)\right|}{d_H(f^a, f^b)} \quad (19)$$

and third, we want to check if the obtained labellings are really weighted mean labellings.

For this experimentation we chose three different configurations for λ_H and λ_C. First, when $\lambda_H = 0$ and $\lambda_C = 1$ the labellings are not being considered, thus basing the decision only on the minimum cost. Therefore, this configuration will reproduce a classical minimum-cost method (red in Figures 1 to 3). Second, when $\lambda_H = 1$ and $\lambda_C = 1$ there is a contribution both of the cost and the labelling. Therefore, this approach would represent our method (green in Figures 1 to 3). Finally, when $\lambda_H = 1$ and $\lambda_C = 0$ only the labellings are being considered but no cost is used. Therefore this approach would be considered a pure consensus of the correspondences done by $f^a$and f^b(violet in Figures 1 to 3).

Figure 1 shows the cost of $\bar{f}^*_{\lambda_C,\lambda_H}$as the number of mistakes increases. Notice that the y-axis represents the cost (equation 1, where the cost between nodes is equation 18), being an application-dependent metric. However, it is clearly noticeable that the classical method (red) performs just as good as our method in terms of minimising the cost. It must be pointed out that a minimum cost not necessarily translates in a better result

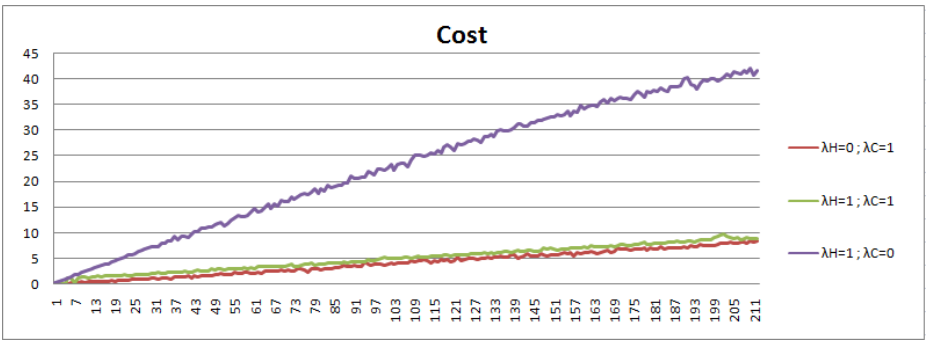


Fig. 1. Comparison for the cost of $\bar{f}^*_{\lambda_C,\lambda_H}$ for the three configurations

Figure 2 shows the Middle measure (equation 19) which measures how far in terms of Hamming Distance is the consensus labelling with respect of the median of f^a and f^b. We can see once again that the classical method (red) performs slightly worse than our approach, however as mistakes increase, the distance "Middle" decreases and eventually stabilizes for every configuration.

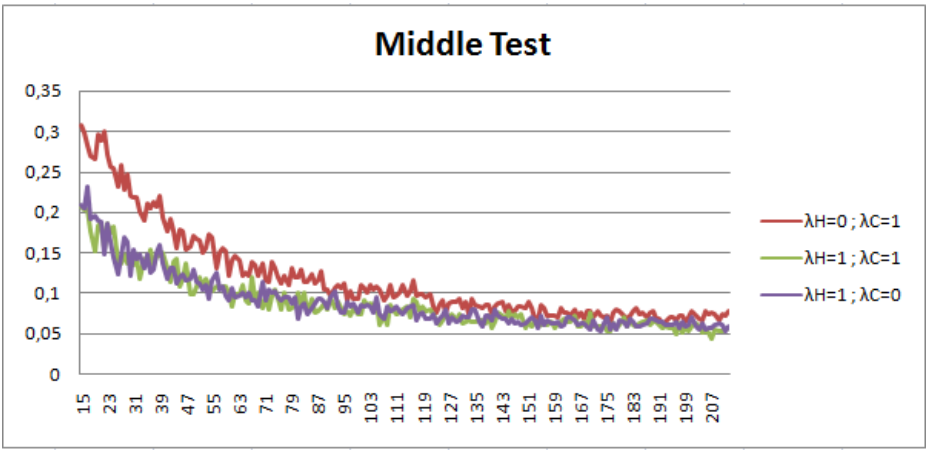


Fig. 2. Comparison for the position of $\bar{f}^*_{\lambda_C,\lambda_H}$ for the three configurations

Figure 3 shows the percentage of experiments in which $\bar{f}^*_{\lambda_C,\lambda_H}$ is really a weighted mean of f^a and f^b. We can notice that the classical method (red) does not always give weighted means starting from 25 mistakes (which means that it stops delivering results that are consensus). As we deducted from equation 17, the labelling-only configuration (violet) will always result in weighted means, whereas the approach that equally considers both terms (green) will slightly decrease in successful weighted mean results as α increase.

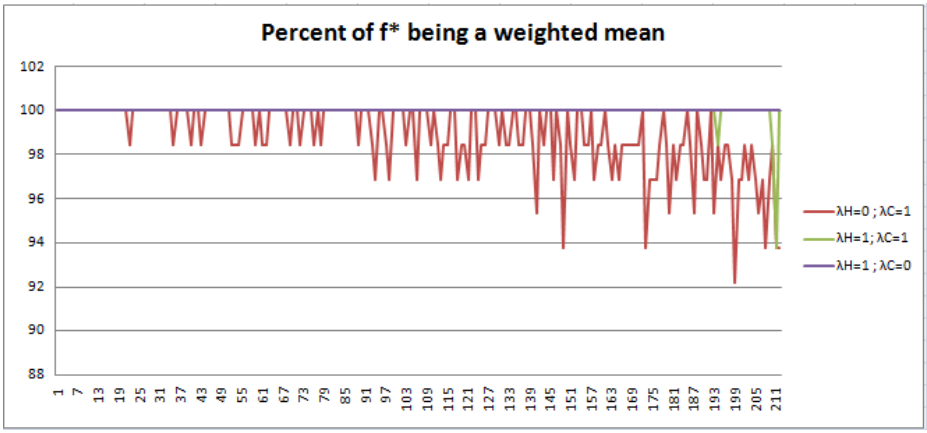


Fig. 3. Comparison for the percent of $\bar{f}^*_{\lambda_C,\lambda_H}$ being a weighted mean for the three configurations

5 Conclusions and Further Work

We present a fast and efficient method to perform a consensus decision based on a regularization term consisting in two labellings developed by two separate entities, and a loss function consisting on a cost relation which is application dependant. We also demonstrate that an optimisation process can be applied to reduce the computational cost of calculating the multiple possibilities and that different configurations on the loss function and the regularisation term can be produced to obtain different results. As a further work, we would like to continue studying the effects of consensus techniques in with multiple inputs and multiple metrics [17].

References

1. Franek, F., Jiang, X., He, C.: Weighted Mean of a Pair of Clusterings. Pattern Analysis Applications. Springer (2012)
2. Mimaroglu, S., Erdil, E.: Combining multiple clusterings using similarity graphs. Pattern Recognition 44, 694–703 (2010)
3. Murino, L., Angelini, C., De Feis, I., Raiconi, G., Tagliaferri, R.: Beyond classical consensus clustering: The least squares approach to multiple solutions. Pattern Recognition Letters 32, 1604–1612 (2011)
4. Ghahramani, Z.: Unsupervised Learning. Advanced Lectures on Machine Learning, pp. 72–112. Springer (2004)
5. Papadimitriou, C., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Dover Publications (July 1998)
6. Kuhn, H.W.: The Hungarian method for the assignment problem Export. Naval Research Logistics Quarterly 2(1-2), 83–97 (1955)
7. Qing, Y., Haizhen, W., Zhenzhen, J., Yan, P.: Graph Theory and its application in optimization of gas drainage system in coal mine. Procedia Engineering 45, 339–344 (2012)

8. Gao, L., Wang, Y., Tang, Z., Lin, X.: Newspaper article reconstruction using ant colony optimization and bipartite graph. Applied Soft Computing 13, 3033–3046 (2013)
9. Eberspächer, P., Verl, A.: Realizing energy reduction of machine tools through a control-integrated consumption graph-based optimization method. Procedia CIRP 7, 640–645 (2013)
10. Livi, L., Rizzi, A., Sadeghian, A.: Optimized dissimilarity space embedding for labeled graphs
11. Solé, A., Serratosa, F., Sanfeliu, A.: On the Graph Edit Distance cost: Properties and Applications. International Journal of Pattern Recognition and Artificial Intelligence. IJPRAI 26(5) (2012)
12. Jain, A.K., Feng, J.: Latent Palmprint Matching. IEEE Trans. on PAMI (2009)
13. Funada, J., et al.: Feature Extraction Method for Palmprint Considering Elimination of Creases. In: Proc. 14th Int. Conf. Pattern Recognition, pp. 1849–1854 (1998)
14. Dai, J., Zhou, J.: Multifeature-Based High-Resolution Palmprint Recognition. IEEE Trans. Pattern Analysis and Machine Intelligence 33(5), 945–957 (2011)
15. Dai, J., Feng, J., Zhou, J.: Robust and Efficient Ridge Based Palmprint Matching. IEEE Transactions on Pattern Analysis and Matching Intelligence 34(8) (2012)
16. Kuhn, H.W.: The Hungarian Method for the assignment problem. Naval Research Logistics Quarterly 2, 83–97 (1955)
17. Franek, L., Jiang, X.: Ensemble clustering by means of clustering embedding in vector space. Pattern Recognition 47(2), 833–842 (2014)

Chemical Symbol Feature Set for Handwritten Chemical Symbol Recognition

Peng Tang, Siu Cheung Hui, and Chi-Wing Fu

School of Computer Engineering,
Nanyang Technological University, Nanyang Avenue, Singapore 639798
{ptang1,asschui,cwfu}@ntu.edu.sg

Abstract. There are two main types of approaches for handwritten chemical symbol recognition: image-based approaches and trajectory-based approaches. The current image-based approaches consider mainly the geometrical and statistical information from the captured images of users' handwritten strokes, while the current trajectory-based recognition approaches only extract temporal symbol features on users' writing styles. To recognize chemical symbols accurately, however, it is important to identify an effective set of important chemical features by considering the writer dependent features, writer independent features as well as context environment features. In this paper, we propose a novel CF44 chemical feature set based on the trajectory-based recognition approach. The performance of the proposed chemical features is also evaluated with promising results using a chemical formula recognition system.

1 Introduction

There are two main types of approaches for handwritten chemical symbol recognition, namely *image-based approaches* and *trajectory-based approaches*. The image-based approaches aim at recognizing chemical symbols based on the image input of the pen strokes. The trajectory-based recognition approaches recognize chemical symbols when they are written with a pen-based device, where a sensor picks up the pen-tip movements as well as pen-up/pen-down switching. The actuarial pen trajectory data is known as digital ink and it is captured for the recognition process.

For the past decade, different techniques such as Support Vector Machines (SVM) [3], Hidden Markov Models (HMM) [8], hybrid SVM-HMM [7] and Support Vector Machine-Elastic Matching (SVM-EM) [6] have been proposed for chemical symbol recognition. The current image-based approaches [3,5,4] mainly considered the geometrical and statistical information from the captured images of users' handwriting strokes. As such, users' writing styles during the writing process are not captured. On the other hand, the current trajectory-based recognition approaches [8,7] only considered extracting temporal symbol features based on stroke points. These techniques only focused on capturing features about the writing process which are dependent on users' writing styles.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 312–322, 2014.

To recognize chemical symbols accurately, it is important to identify an effective set of important chemical features from users' handwritten input data. In this paper, we propose a novel CF44 chemical feature set which consists of writer dependent features, writer independent features and context environment features. We limit the feature set to a total of 44 important features in order not to over influence the processing speed of the feature extraction process. The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 presents the proposed chemical symbol features. Experimental results are then presented in Section 4. Finally, Section 5 concludes the paper.

2 Related Work

Different kinds of features have been proposed based on the image-based symbol recognition approaches. In [5], geometrical features based on lines and polygons were extracted and recognized with text recognition. In [3], Ouyang and Davis used both the geometrical and statistical features including number of strokes, bounding-box dimensions, stroke ink density, inter-stroke distance and inter-stroke orientation for SVM symbol recognition. In [4], Ouyang and Davis further proposed to extract additional geometrical features from the stroke segments including segment length, segment count, stroke diagonal, symbol diagonal and symbol ink density for symbol recognition using the Conditional Random Field (CRF). However, the current image-based approaches can only capture the visual features about the handwritten chemical symbols.

For trajectory-based symbol recognition approaches, different types of chemical features have also been extracted. In [8,7], Zhang et al. proposed 11 types of chemical features including normalized stroke point coordinates, normalized first derivatives and second derivatives of the stroke points, curvature, writing direction, aspect, curliness and linearity for symbol recognition in HMM and SVM-HMM. In [6], Tang et al. used some simple chemical features including number of strokes, stroke point coordinates, horizontal angle and turning angle for the SVM-EM symbol recognition. As the chemical features extracted by the current trajectory-based approaches are quite limited, in this paper we propose an effective set of chemical features based on the trajectory-based approach for chemical symbol recognition.

3 Chemical Symbol Features

Chemical symbols consist of digits, alphabetic characters, bonds, and operators. For the purpose of handwritten chemical symbol and formula recognition, we propose a set of 44 chemical symbol features to capture the various aspects of handwritten chemical symbols. They are classified into 11 writer dependent features, 31 writer independent features and 2 context environment features.

First, we define the following fundamental concepts on *stroke*, *chemical symbol pattern* and *bounding box*:

Definition 1 (Stroke). *A stroke s is a sequence of m two-dimensional points $(p_1, p_2, ..., p_m)$:*

$$s = (p_1, p_2, ..., p_m),$$

where $p_i = (x_i, y_i), 1 \leq i \leq m$, p_1 is the pen-down point, p_m is the pen-up point and $p_2, ..., p_{m-1}$ are the pen-move points.

Definition 2 (Chemical Symbol Pattern). *A chemical symbol pattern S is a valid sequence of k strokes which are recognized together as a chemical symbol. Therefore, it is also a sequence of n two-dimensional points $(p_1, p_2, ..., p_n)$:*

$$S = (s_1, s_2, ..., s_k) = (p_1, p_2, ..., p_n),$$

p_1 and p_n are the starting point and ending point of the chemical symbol pattern.

Definition 3 (Bounding Box). *The bounding box $B = (w, h)$ of a symbol S is the smallest rectangle which encloses S. The edges of the rectangle are in parallel with the coordinate axes. w and h are the width and height of the bounding box. And the center of the bounding box is defined as the center of S.*

3.1 Writer Dependent Features

The writer dependent features are dynamic features including the order of points, writing direction, and the number and order of strokes which model the writing process of handwritten chemical symbols.

Pattern Starting and Ending Point Features. The positions of the starting point and ending point of the chemical symbol pattern are important features for the recognition of chemical symbols since users often write the same symbol with similar positions on its starting point and ending point. However, if a handwritten chemical symbol has very small width or height such as chemical bond symbols, we will extract the pattern starting point and ending point features by standardizing them with respect to a bounding square. The bounding square is defined as follows:

Definition 4 (Bounding Square). *The bounding square of a symbol pattern S is the smallest virtual square which encloses S. The center of the bounding square $c = (x_c, y_c)$ is located at the center of the chemical symbol pattern. The edges of the bounding square are in parallel with the coordinate axes. The length l of the bounding square is the larger of the width w and height h of the bounding box.*

For a chemical symbol pattern $S = (p_1, ..., p_n)$ with starting point $p_1 = (x_1, y_1)$ and ending point $p_n = (x_n, y_n)$, the pattern starting point features (f_1 and f_2) and ending point features (f_3 and f_4) are formulated as follows:

$$\textit{Starting Point:} \quad f_1 = \frac{x_1 - x_c}{l} + \frac{1}{2}, \quad f_2 = \frac{y_1 - y_c}{l} + \frac{1}{2}, \tag{1}$$

$$\textit{Ending Point :} \quad f_3 = \frac{x_n - x_c}{l} + \frac{1}{2}, \quad f_4 = \frac{y_n - y_c}{l} + \frac{1}{2}. \tag{2}$$

where l is the length of the bounding square.

Pattern Writing Direction Features. The pattern writing direction features model users' general writing direction when users write a chemical symbol.

Definition 5 (Pattern Writing Direction). *For a chemical symbol pattern* $S = (p_1, p_2, ..., p_n)$*, the pattern writing direction of* S *is defined as a vector from the starting point* p_1 *to the ending point* p_n*.*

The writing direction features are important since users often write the same symbol with the same direction. The writing length feature (f_5) and direction features (f_6 and f_7) are formulated as follows:

$$\textit{Writing Length:} \quad f_5 = \|\mathbf{v}\| = \|p_1 p_n\|, \tag{3}$$

$$\textit{Writing Direction:} \quad f_6 = \frac{\mathbf{v_x} \cdot \mathbf{u_x}}{\|\mathbf{v}\|}, \quad f_7 = \frac{\mathbf{v_y} \cdot \mathbf{u_y}}{\|\mathbf{v}\|}, \tag{4}$$

where $\mathbf{v}$ is the starting point to ending point vector $\overrightarrow{p_1 p_n}$ which models the writing direction. $\mathbf{u_x}$ and $\mathbf{u_y}$ denote the unit vector of the horizontal and vertical axes respectively.

The two direction features f_6 and f_7 become unstable when the writing length feature f_5 is too small. To tackle this problem, the values of the two direction features will be set to zero when the writing length feature is less than a minimal distance threshold which is set empirically to $\max(w, h)/4$.

Pattern Closure Feature. It is used to distinguish the difference between chemical symbols with closed or circular writing trajectory pattern (such as 'O') and those with straight line writing trajectory pattern (such as 'I' and chemical bond symbols). This feature is defined based on the writing length feature f_5.

Definition 6 (Pattern Closure). *The pattern closure of a symbol pattern is defined as the ratio between the writing length and the trajectory length of the handwritten chemical symbol pattern.*

For a chemical symbol pattern $S = (p_1, p_2, ..., p_n)$, the pattern closure feature (f_8) is formulated as follows:

$$\textit{Closure:} \quad f_8 = \frac{\|\mathbf{v}\|}{L}, \tag{5}$$

where $L = pathdist(p_1, p_n) = \sum_{i=1}^{k} \sum_{j} \|\overrightarrow{p_j p_{j+1}}\|$, L is the path distance between the pattern starting point p_1 and ending point p_n. And it is calculated by summing up the length of each stroke segment $\overrightarrow{p_j p_{j+1}}$ in S. It indicates the length of the writing trajectory by the user.

Pattern Inflection Features. The pattern inflection features are used to distinguish the convex and concave curvature patterns shown in the handwritten chemical symbol.

Definition 7 (Pattern Inflection). *The pattern inflection of a chemical symbol pattern* S *is a measure of the relative positioning of the middle-path point* p_{mid} *of* S *with respect to the middle point* p_m *of the straight line* $\|\overrightarrow{p_1 p_n}\|$*.*

The pattern inflection features (f_9 and f_{10}) are formulated as follows:

$$\textit{X and Y-Inflection:} \quad f_9 = \frac{1}{w}(x_{mid} - x_m), \quad f_{10} = \frac{1}{h}(y_{mid} - y_m), \tag{6}$$

where $p_m = (x_m, y_m) = (\frac{x_1+x_n}{2}, \frac{y_1+y_n}{2}) = \frac{p_1+p_n}{2}$.

The pattern inflection features f_9 and f_{10} are normalized by the width w and height h of the bounding box of the chemical symbol pattern respectively.

Total Stroke Number Feature. It is an important feature to indicate the writing complexity of the handwritten chemical symbol. The total stoke number feature (f_{11}) accounts for the number of strokes k in the handwritten chemical symbol pattern as follows:

$$\textit{Total Stroke Number:} \qquad f_{11} = k. \tag{7}$$

3.2 Writer Independent Features

The writer independent features are visual features which model the appearance of the handwritten chemical symbols. These features will not be affected by changes in users' stroke order or writing direction.

Bounding Box Features. The bounding box features model the basic information about the shape of the handwritten chemical symbol. The diagonal angle feature (f_{13}) and diagonal ratio feature (f_{14}) are formulated as follows:

$$\textit{Diagonal Angle and Ratio:} \qquad f_{13} = arctan\frac{h}{w}, \qquad f_{14} = \frac{h+w}{L}. \tag{8}$$

The diagonal angle feature measures the angle between the bounding box diagonals with respect to the horizontal axis. And the diagonal ratio is a relative measure of the bounding box size which characterizes the complexity when writing the chemical symbol.

Deviation Feature. The deviation feature is used to measure the dispersion of stroke points in the handwritten chemical symbol. For a chemical symbol pattern $S = (p_1, p_2, ..., p_n)$, the deviation feature (f_{15}) is formulated as follows:

$$\textit{Pattern Deviation:} \qquad f_{15} = \frac{1}{n}\sum_{i=1}^{n} \|p_i\mu\|, \tag{9}$$

where $\mu = \sum_{i=1}^{n} p_i/n$, and p_i is the i-th stroke point in the handwritten chemical symbol. μ is the center of gravity which is calculated as the mean of all the stroke points in the chemical symbol pattern. And $\|p_i\mu\|$ is the Euclidean distance between the i-th stroke point p_i and the center of gravity μ.

Average Direction Feature. The average direction feature is computed by averaging the writing directions of all the stroke segments defined in the trajectory of the handwritten chemical symbol. For a chemical symbol pattern $S = (p_1, p_2, ..., p_n)$, the average direction feature (f_{16}) is formulated as follows:

$$\textit{Average Direction:} \qquad f_{16} = \frac{1}{n}\sum_{i=1}^{n-1} arctan(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}). \tag{10}$$

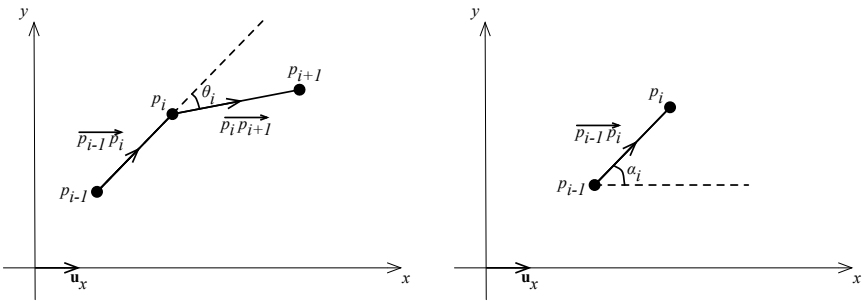
Curvature Feature. The curvature feature measures the curvature of the handwritten chemical symbol. Here, we introduce *stroke turning angle* which is used to measure the local curvature for two consecutive stroke segments.

Definition 8 (Stroke Turning Angle). *The stroke turning angle is the turning angle between two consecutive stroke segments within the same stroke. For a chemical symbol pattern $S = (p_1, p_2, ..., p_n)$, the stroke turning angle between the stroke segments $\overrightarrow{p_{i-1}p_i}$ and $\overrightarrow{p_ip_{i+1}}$ is defined as follows:*

$$\theta_i = arccos(\frac{\overrightarrow{p_{i-1}p_i} \cdot \overrightarrow{p_ip_{i+1}}}{\|\overrightarrow{p_{i-1}p_i}\| \|\overrightarrow{p_ip_{i+1}}\|}).$$

An example on stroke turning angle is shown in Figure 1a. The curvature feature (f_{17}) is then formulated by summing up all the local curvatures in S:

$$\textit{Pattern Curvature:} \qquad f_{17} = \sum_{i=2}^{n-1} \theta_i. \tag{11}$$



(a) Stroke Turning Angle (b) Stroke Segment Orientation

Fig. 1. Stroke Turning Angle and Stroke Segment Orientation

Therefore, symbols which consist of straight lines will have low curvature, whereas symbols which consist of curved strokes will have high curvature. For example, the symbol 'H' will have lower curvature measure than the symbol 'O'.

Perpendicularity Features. The perpendicularity features are important features to identify the abrupt changes in the stroke trajectory, especially for symbols with sharp turning strokes such as 'A' and 'N'. For a chemical symbol pattern $S = (p_1, p_2, ..., p_n)$, the perpendicularity features (f_{18} and f_{19}) are formulated as follows:

$$\textit{Perpendicularity:} \qquad f_{18} = \sum_{i=2}^{n-1} sin^2\theta_i, \tag{12}$$

$$\textit{2-th Perpendicularity:} \qquad f_{19} = \sum_{i=2}^{n-1} sin^2(\theta_i^2), \tag{13}$$

where 2nd-turning angle $\theta_i^2 = arccos\left\{\frac{\overrightarrow{p_{i-2}p_i}\cdot\overrightarrow{p_ip_{i+2}}}{\|\overrightarrow{p_{i-2}p_i}\|\|\overrightarrow{p_ip_{i+2}}\|}\right\}$.

Directional Histogram Features. The directional histogram features help to capture information about writing direction of each stroke segment of the chemical symbol pattern. First, we define *stroke segment orientation* as follows:

Definition 9 (Stroke Segment Orientation). *The stroke segment orientation is a measure for the writing direction of that stroke segment. For a chemical symbol pattern* $S = (p_1, p_2, ..., p_n)$*, the stroke segment orientation for a stroke segment* $\overrightarrow{p_ip_{i+1}}$ *is calculated as:*

$$\alpha_i = arccos\left\{\frac{\overrightarrow{p_ip_{i+1}}\cdot\mathbf{u_x}}{\|\overrightarrow{p_ip_{i+1}}\|}\right\}. \tag{14}$$

Figure 1b illustrates the definition of stroke segment orientation.

Since a chemical bond symbol can have 12 possible directions in chemical formulas, the stroke segment range is divided into 12 directional histogram bins ($sh_1 - sh_{12}$) and each directional histogram bin indicates one of the 12 possible directions for stroke segment orientations. The range between two consecutive directional histogram bins is 30°. A fuzzy quantization approach is used to quantize the stroke segment orientation into histogram bins. A stroke segment orientation α_i contributes to two histogram bins with two different weights:

$$w_{(m,i)} = 1 - \frac{6}{\pi}\cdot arccos\{\mathbf{u}_{\alpha_i}\cdot\mathbf{v_m}\}, \quad w_{(m+1,i)} = \frac{6}{\pi}\cdot arccos\{\mathbf{u}_{\alpha_i}\cdot\mathbf{v_{m+1}}\}, \tag{15}$$

where $\mathbf{u}_{\alpha_i}$ is the unit vector which has the same orientation as stroke segment orientation α_i and $\mathbf{v_m}$ is the unit vector for the orientation of directional histogram bin sh_m. Each histogram bin is then calculated as the sum of weighted contributions from each stroke segment:

$$sh_m = \sum_{i=1}^{n-k} w_{(m,i)}. \tag{16}$$

We define 6 directional histogram features based on the stroke segment orientation by inspecting the number of stroke segments oriented in the 12 directional histogram bins. The 6 directional histogram features (f_{20} to f_{25}) for the stroke segment orientation are formulated as follows:

$$\textit{Stroke Directional Histogram}: \quad f_{20} = \frac{sh_1 + sh_7}{n-k}, ..., f_{25} = \frac{sh_6 + sh_{12}}{n-k}. \quad (17)$$

Similarly, we define 6 directional histogram features (f_{26} to f_{31}) for the stroke turning angle, which provides additional curvature information about the writing trajectory of the symbol pattern, as follows:

$$\textit{Turning Directional Histogram}: \quad f_{26} = \frac{th_1 + th_7}{n - 2 \times k}, ..., f_{31} = \frac{th_6 + th_{12}}{n - 2 \times k}. \quad (18)$$

where th_1 to th_{12} are the 12 directional histogram bins for stroke turning angle.

2-Dimensional (2D) Histogram Features. The 2D histogram features measure which area of the symbol pattern is denser or has more stroke points. To extract the features, the bounding box of a symbol is divided into 3×3 2D histogram bins. Figure 2 shows the 3×3 2D histogram bins for a handwritten chemical symbol 'C'. Then, each stroke point will be quantized into 4 nearest histogram bins with the respective weightings. The weighted contributions to the 4 histogram bins are determined by the distance from the stroke point to the center of the 4 respective histogram bins.

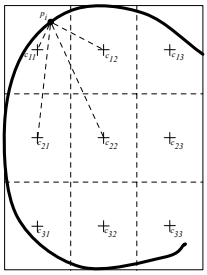


Fig. 2. 2D Histogram Bins

Nine 2D histogram features are extracted and each feature is calculated by inspecting all the contributions to each 2D histogram bin from the stroke points in the handwritten chemical symbol pattern. The 2D histogram features (f_{32} to f_{40}) are formulated as follows:

$$\textit{2D Histogram}: \quad f_{32} = \frac{h_{11}}{n} = \frac{1}{n}\sum_{i=1}^{n} w_{11}(p_i), ..., f_{40} = \frac{h_{33}}{n} = \frac{1}{n}\sum_{i=1}^{n} w_{33}(p_i) \quad (19)$$

Convex Hull Features. The convex hull of the chemical symbol pattern is a set of points which can be used to construct the smallest convex shape to enclose the chemical symbol pattern. Therefore, the convex hull features are a perfect measure to model the geometric information of the handwritten chemical symbol pattern. In this research, the convex hull of the chemical symbol pattern is computed using the Graham algorithm [2]. The two convex hull features (f_{40} and f_{41}) are formulated as follows:

$$\textit{Convex Hull Box Ratio:} \quad f_{41} = \frac{A_H}{w \times h}, \tag{20}$$

$$\textit{Convex Hull Trajectory Ratio:} \quad f_{42} = \frac{L^2}{A_H}, \tag{21}$$

where the convex hull H is a sequence of points $(v_1, ..., v_i, ..., v_m)$ with $v_i = (v_{i,x}, v_{i,y})$ and the convex hull area $A_H = \frac{1}{2}\left|\sum_{i=1}^{p-1}(v_{i,x} \times v_{i+1,y} - v_{i+1,x} \times v_{i,y})\right|$.

3.3 Context Environment Features

The context environment features capture the related context information on the handwritten chemical symbols when users write the chemical formulas. The context environment features are only possible after the first symbol of a handwritten chemical formula is recognized. The context environment features are extremely useful to distinguish symbols with the same shape and writing direction, for example, capital letters and its non-capital. There are two context environment features: previous symbol type (f_{43}) and relative symbol height (f_{44}). They are formulated as follows:

$$\textit{Previous Symbol Type:} \quad f_{43} = \begin{cases} 0 \text{ if } S_P \text{ is a digit} \\ 1 \text{ if } S_P \text{ is a character} \\ 2 \text{ if } S_P \text{ is an operator} \\ 3 \text{ if } S_P \text{ is a bond} \end{cases}, \tag{22}$$

$$\textit{Relative Symbol Height:} \quad f_{44} = \frac{h}{h_P}, \tag{23}$$

where S_P is the previous recognized symbol of the current unknown symbol in the chemical formula. h_P is the height of the previous symbol. f_{43} models the symbol type of the previous symbol and f_{44} captures the relative height information between the previous symbol and the current unknown symbol.

4 Performance Evaluation

In this research, we have developed a handwritten chemical formula recognition system on the iOS platform, called iDrawChem. In the system, LibSVM [1] is used for symbol recognition with the proposed chemical features. The performance evaluation was conducted as follows. First, we introduced the iDrawChem system to 10 users. Then, the users spent about 10 minutes to familiarize themselves with the system and experimented a few simple expressions which were different from those used for testing. We prepared 468 valid simple chemical expressions and the previous symbol of the chemical symbol was also contained in each chemical expression. In the experiment, each user was required to write all the chemical expressions. Then, the writer dependent features, writer independent features and context environment features of the chemical symbols were recorded. As a result, a total of 4680 valid symbol expressions were collected

Table 1. Performance Results based on Symbol Categories

Category	**Precision@1**	**Precision@3**	**Precision@5**
Digits	95.67%	98.33%	99.33%
Alphabetical Characters	92.56%	96.22%	98.85%
Bonds & Operators	91.9%	97.50%	98.96 %
Average	93.80%	96.75%	98.93 %

Table 2. Performance Results based on Feature Types

Features	**Precision@1**	**Precision@3**	**Precision@5**
WD	69.49%	75.56%	83.50%
WI	86.97%	90.47%	94.15%
WD+WI	92.82%	95.13%	97.61%
WD+WI+CE	93.80%	96.75%	98.93%

from the 10 users. Among them, 2340 of the symbol expressions were used for training the SVM classifier and the rest were used for testing.

Table 1 shows the performance results based on different symbol categories. As our symbol recognition is able to return a ranked list of candidate symbols for user selection, we use precision@1, precision@3 and precision@5 for performance evaluation. Precision@n reports the recognition accuracy that the correct symbol is in the top-n candidate symbols. As shown in the table, the performances at precision@1, precision@3 and precision@5 are 93.80%, 96.75% and 98.93% respectively which are quite promising. The performance of digits is better than that of alphabetical characters, bonds and operators, as digits generally have more distinguishing features.

Table 2 gives the performance results based on different types of chemical features: Writer Dependent (WD) features, Writer Independent (WI) features, Context Environment (CE) features. As shown in the table, the performance with WI features (86.97% at precision@1) is much higher than that with WD features (69.49% at precision@1), as writer independent features which capture the visual appearance of the symbols may help recognize the symbols more effectively than writer dependent features. In addition, using WD features and WI features together, the performance has improved to 92.82% at precision@1. Furthermore, the CE features can further improve the performance to 93.80% at precision@1 since they can help distinguish symbols with the same visual shape and writing direction.

5 Conclusion

In this paper, we have proposed a set of 44 chemical symbol features which can effectively model the user writing process, visual appearance and context environment of each handwritten chemical symbol. The proposed CF44 chemical symbol features are designed by considering both writer dependent and writer independent features as well as the context environment features. The proposed features have been evaluated using our chemical formula recognition system called iDrawChem. The performance evaluation has shown promising results with high accuracy up to 98.93% at precision@5. Therefore, the proposed chemical features are effective to be used for the recognition of handwritten chemical symbols for chemical formula recognition.

References

1. Chang, C., Lin, C.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3), 27 (2011)
2. Graham, R.L.: An efficient algorith for determining the convex hull of a finite planar set. Information Processing Letters 1(4), 132–133 (1972)
3. Ouyang, T., Davis, R.: Recognition of hand drawn chemical diagrams. In: Proceedings of the National Conference on Artificial Intelligence, vol. 22, p. 846. AAAI Press; MIT Press, Menlo Park; Cambridge (2007)
4. Ouyang, T., Davis, R.: Chemink: A natural real-time recognition system for chemical drawings. In: Proceedings of the 16th International Conference on Intelligent User Interfaces, pp. 267–276. ACM (2011)
5. Ramel, J., Boissier, G., Emptoz, H.: Automatic reading of handwritten chemical formulas from a structural representation of the image. In: Proceedings of the Fifth International Conference on Document Analysis and Recognition, pp. 83–86. IEEE (1999)
6. Tang, P., Hui, S., Fu, C.: Online chemical symbol recognition for handwritten chemical expression recognition. In: Proceedings of the 12th International Conference on Computer and Information Science (2013)
7. Zhang, Y., Shi, G., Wang, K.: A svm-hmm based online classifier for handwritten chemical symbols. In: Proceeding of International Conference on Pattern Recognition 2010, pp. 1888–1891. IEEE (2010)
8. Zhang, Y., Shi, G., Yang, J.: Hmm-based online recognition of handwritten chemical symbols. In: Proceedings of 10th International Conference on Document Analysis and Recognition. pp. 1255–1259. IEEE (2009)

About Combining Metric Learning and Prototype Generation*

Adrian Perez-Suay, Francesc J. Ferri,
Miguel Arevalillo-Herráez, and Jesús V. Albert

Dept. Informàtica, Universitat de València. Spain
{Adrian.Perez,Francesc.Ferri,Miguel.Arevalillo,Jesus.V.Albert}@uv.es

Abstract. Distance metric learning has been a major research topic in recent times. Usually, the problem is formulated as finding a Mahalanobis-like metric matrix that satisfies a set of constraints as much as possible. Different ways to introduce these constraints and to effectively formulate and solve the optimization problem have been proposed. In this work, we start with one of these formulations that leads to a convex optimization problem and generalize it in order to increase the efficiency by appropriately selecting the set of constraints. Moreover, the original criterion is expressed in terms of a reduced set of representatives that is learnt together with the metric. This leads to further improvements not only in efficiency but also in the quality of the obtained metrics.

1 Introduction

Classifying and/or conveniently representing high dimensional data has always been a very important goal in many different domains across the pattern recognition and image analysis fields. When the objects under study correspond to large collections of images or any other kind of visual information, this issue becomes even more critical due to the huge sizes usually involved. The classical approach for dealing with such high dimensional data is to apply some kind of dimensionality reduction in order to look for either numerical stability, performance improvement or simply to be able to get results in a reasonable amount of time [1, 2].

Dimensionality reduction has been largely studied from different points of view. In particular, linear methods such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are very well-known and commonly used in practice [1, 3]. Particular implementations and extensions of these have been proposed in particular domains such as face recognition [4–7].

The vast majority of approaches that propose using either linear or non linear dimensionality reduction to map the original problem into a (usually simplified) representation space, end up using a straightforward distance-based classification

* This work has been partially funded by FEDER and Spanish MEC through projects TIN2009-14205-C04-03, TIN2011-29221-C03-02 and Consolider Ingenio 2010 CSD2007-00018.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 323–332, 2014.

method in this space. The combination of the mapping and distance function can be seen as a composite (and possibly complex) metric in the original space. This puts forward the close relation that exists between dimensionality reduction and metric learning. Metric learning has received recent interest and has been tackled from very different viewpoints [8–11] using rather different methodologies to learn a convenient metric for a particular problem.

Basically, all methods that directly look for a (usually parameterized) distance function follow to some extent the same rationale that guides most (discriminant) dimensionality reduction approaches. This consists of increasing the effective distances between objects from different classes while decreasing the distances among objects of the same class. To this end, different approaches explicitly use distances either to define criteria or introduce constraints in the formulation along with different kinds of regularizers [8, 12, 13].

Regardless of the particular way of formulating the problem, one can distinguish between the criterion (and how exactly it relates to the ultimate goal of obtaining an appropriate metric), and the particular training information that is given to the algorithm (usually as sets of similar and dissimilar pairs). For example, different particular methods use different strategies to select this training information ranging from using all possible pairs [12] to pairs in the near vicinity of particular points [14]. More recently, it has been proposed to learn the best training pairs along with the metrics [15].

In this paper, we also propose a way of progressively adapting the training pairs along with the metric. Starting from the convex formulation used for MCML (Maximally Collapsing Metric Learning, [12]), we generalize it by introducing a reduced set of representative prototypes. With this generalization it is possible to obtain Mahalanobis like metrics that improve the results of the original algorithm, using a smaller amount of (selected) training pairs. Experimentation using several publicly available databases has been carried out to empirically validate the benefits of the proposed approach.

2 Metric Learning and Collapsing Classes

Given a collection of objects in a multidimensional vector space, $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$, let us consider distances parametrized by a positive semi-definite (PSD) matrix, A, as $d^A(x_i, x_j) = (x_i - x_j)^T A(x_i - x_j)$.

This quadratic distance (also referred to as Mahalanobis distance by analogy) is at the root of much recent work on metric learning in which the goal consists of appropriately estimating these matrices. As any PSD matrix can be decomposed as $A = W^T W$, using the above distance is equivalent to mapping the objects using W and then using the Euclidean distance on them.

The MCML algorithm [12], works by looking for a matrix A whose corresponding mapping makes all classes collapse into a single target point per class (which means null distances), which are arbitrarily far away from each other.

To construct a criterion that measures goodness with regard to the above idealized mapping, the following probability of x_i being *similar* (i.e. from the

same class in the context of the present paper) to any other x_j is introduced as follows.

$$p^A(j|i) = \frac{1}{Z_i} e^{-d_{ij}^A} = \frac{e^{-d_{ij}^A}}{\sum_{k \neq i} e^{-d_{ik}^A}}$$

where $d_{ij}^A = d^A(x_i, x_j)$. For each i, this is a discrete probability density function ranging for all j such that $i \neq j$.

The more classes collapse into a single point and are far away from each other, the closer these probabilities will be to the following target probability:

$$p_0(j|i) \propto \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ belong to the same class} \\ 0 & \text{otherwise} \end{cases}$$

The Kullback-Leibler divergence can be used as an objective measure of how far we are from the goal of having all classes maximally collapsed. The criterion to be minimized is the above mentioned divergence averaged for all objects i, which can be written as [12]:

$$J_M(A, X) = \frac{1}{n} \sum_{i=1}^{n} p_0(j|i) \log \frac{p_0(j|i)}{p^A(j|i)}$$

This criterion can be changed to an equivalent one after obviating constant terms:

$$J(A, X) = \frac{1}{n} \sum_{i,j=1}^{n} p_0(j|i) d_{ij}^A + \frac{1}{n} \sum_{i=1}^{n} \log \sum_{k \neq i} e^{-d_{ik}^A}$$

When minimizing this criterion with regard to A, the problem becomes convex. This adds some guarantees for applying optimization methods based on the gradient, which is given by:

$$\nabla_A J(A, X) = \frac{1}{n} \sum_{i,j=1}^{n} (p_0(j|i) - p^A(j|i)) \cdot \nabla_A d_{ij}^A$$

where $\nabla_A d_{ij}^A = (x_i - x_j)(x_i - x_j)^T$.

The corresponding gradient descent algorithm is guaranteed to converge to a global optimum but is extremely inefficient in practice as it needs to perform $O(n^2)$ operations involving $O(d^2)$ matrices. Moreover, the PSD constraint on A needs to be enforced at each iteration which implies a further $O(d^3)$ computational burden per iteration.

In the rest of the paper we will restrict ourselves to moderate dimensional problems and will concentrate only in reducing the $O(n^2)$ cost as much as possible at the same time that the quality of the learned metric gets improved.

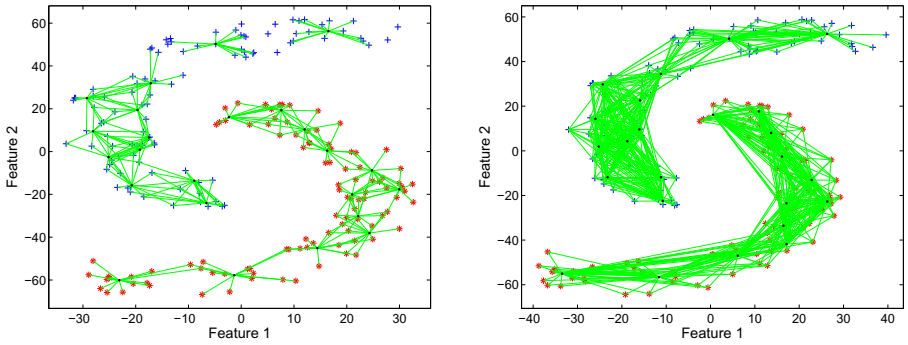


Fig. 1. Illustrative example displaying same-class neighborhood sets, $S_\beta(y_i)$, for a synthetic banana-shaped set for values $\beta = 0.15$(left) and $\beta = 0.4$(right). Different-class sets, $D_\beta(y_i)$ are not shown.

3 Maximally Collapsing Clusters around Representative Prototypes

One of the problems of MCML is related to the computational cost per iteration. A relatively straightforward way of alleviating this problem while maintaining the rationale of the method consists of considering a convenient set of *landmark* or *anchor* points to which distances are measured instead of the whole training set. The same idea has been largely used in the literature in different contexts [16, 17] and specifically for this very same problem [18].

The above expressions need to be rewritten in terms of the given training set, X, and a (reduced) landmark set $Y = \{y_i\}_{i=1}^m$. In both criterion and gradient expressions all terms $(x_i - x_j)$ must be substituted by $(y_i - x_j)$ and the i index must range now over the set Y. If the set Y is small but representative, the new criterion obtained is a good approximation of the original one.

Regardless of the way in which landmark points are obtained, the corresponding algorithm will be referred here to as MCMLA(α), where the A suffix refers to the use of anchors and the proportion $\alpha = \frac{m}{n}$ is the only parameter that controls the size of Y while maintaining the relative sizes of the classes as in X. In the particular case of $X = Y$, we have MCMLA(1) which matches the original MCML algorithm. For high proportion values, the behavior of the algorithm is very similar to MCML. On the other hand, the smaller its value, the more efficient the algorithm will be. Below a particular value of α which is problem dependent, the MCMLA algorithm usually deteriorates due to the poor representativity of the landmarks used with regard to the whole set X.

The first step of the new proposal consists of restricting the probability functions only to objects, x_j, in the close neighborhood of y_i. In fact, for each (fixed) landmark, y_i, we define the $(\beta \cdot n)$-nearest same-class neighbors, $S_\beta(y_i)$, and the

$(\beta \cdot n)$-nearest different-class neighbors, $D_\beta(y_i)$, and then redefine the probabilities as

$$p^A(j|i) = \frac{1}{Z_i} e^{-d^A(y_i, x_j)}, \quad \forall x_j \in N_\beta(y_i)$$

where Z_i must be redefined accordingly and $N_\beta(y_i) = S_\beta(y_i) \cup D_\beta(y_i)$. The parameter $\beta \in (0, 1]$ is a proportion over the size of X that controls the size of the neighborhoods around each landmark. As far as the size of the neighborhoods is fixed, it is straightforward to redefine the above criteria and gradient which will be written now as $J_M(A, Y, X, N_\beta)$, $J(A, Y, X, N_\beta)$ and $\nabla_A J(A, Y, X, N_\beta)$, respectively.

Figure 1 illustrates the S_β sets for two different values of β (0.15 and 0.4) for a fixed number of landmarks generated using k-means clustering on a synthetic banana-shaped two dimensional set. Note that with the proposed modification, probabilities still represent true similarities according to class labels but only in a neighborhood of the landmarks. Note also that the optimization problem is mathematically equivalent but it will lead to a different solution. In addition, the effective size of the set of constraints (pairs of objects) taken into account has been reduced to a proportion which is $\alpha \cdot \beta$ of the original MCML one while the same reduction when using MCMLA is only α.

It is possible to generalize the problem further by considering the set Y as a variable and then try to learn it. To this end, we first write the corresponding gradient as

$$\nabla_Y J(A, Y, X, N_\beta) = \frac{1}{m} \sum_{\substack{i\,:\,y_i \in Y \\ j\,:\,x_j \in N_\beta(x_i)}} (p_0(j|i) - p^A(j|i)) \cdot \nabla_Y d_{ij}^A$$

with $\nabla_Y d_{ij}^A = 2A(y_i - x_j)$. This expression can be plugged into the same gradient based optimization algorithm along with $\nabla_A J(A, Y, X, N_\beta)$ in order to learn both metric and landmarks at the same time. It is important to note that the problem is no longer convex in general. Nevertheless, with reasonable initializations and in a wide range of experiments it is possible to obtain appropriately good results that approximate the MCML ones as both parameters approach one.

The new algorithm will be referred to as MCMLC(α, β) where the C suffix stands for changing anchors. In the following section, several experiments with different kinds of data are carried out in order to put forward the main benefits of the proposal with regard to previous algorithms.

4 Experiments and Results

Several different publicly available databases have been adopted in order to compare the different methods and extensions described in this work. Firstly, some small size databases from the UCI repository [19] as in previous works have been considered. Moreover, databases involving handwritten digits from the Multiple Features Database [20] and the well-known AR face database [21]

Table 1. Details of the databases used in the experimental validation

Name	Size	Dimension	Classes	Objects/class
Iris	150	4	3	50
Wine	178	13	3	48–71
Balance	625	4	3	49–288
Ionosphere	351	34	2	126–225
Mfeat-kar	1000	20	10	100
AR	532	30	38	14

have also been used. For the purposes of this work, the dimensionality of these two databases has been reduced to 20 and 30, respectively, by using PCA. Table 1 shows the details of the databases. In the particular case of the AR database, only 14 images (the ones without occlusions: scarf, glasses, etc.) per individual (20 men and 20 women) have been taken into account.

All data has been used to learn a metric matrix which has been evaluated by computing the leaving one out error of the nearest neighbor classifier in the corresponding mapped space. Although this is well known to be an optimistic measure of (classification) performance, we have found it well suited to make relative comparisons about the quality of the different metrics and mappings. All the presented results correspond to the average of 5 independent runs.

Landmark points for MCMLA(α) have been selected by running a standard k-means algorithm with $k = \alpha \cdot n$ (the number of desired landmark points given by the proportion α). The initial set of prototypes for MCMLC(α, β) has been computed in exactly the same way using α. Moreover, the size of the subsampled set of neighbors for each prototype has been selected as a proportion, β, of the total size of the available training set.

For the experiments in the present work, typical α values in $\{\frac{1}{8}, \frac{1}{4}, \frac{1}{2}\}$ for the MCMLA have ben set [18]. The proportion of prototypes for MCMLC has been set to smaller values, $\alpha \in \{\frac{1}{20}, \frac{1}{10}, \frac{1}{5}\}$ while the proportion of neighbors per prototype, β, has been set to 5 equally spaced values between $\frac{1}{10}$ and $\frac{1}{2}$. In the particular case of the AR database in which the number of objects per class is only 14, the 3 values of α and 5 values of β have been set as equally spaced between $\frac{1}{5}$ and $\frac{2}{5}$, and $\frac{1}{10}$ and $\frac{1}{2}$, respectively.

The methods considered in this work and corresponding extensions have been implemented using the toolbox drtools [22]. All the other parameters of the different methods have been tuned as in the above toolbox and taking into account appropriate ranges. All databases were centered and normalized (to a fixed and common domain) prior to using the algorithms.

For illustration purposes, some measures on the MCMLC algorithm as it iterates using the small database Wine are shown in Figure 2. These are representative of the behavior of the algorithm in all the databases considered in this work. In particular, the value of the modified criterion, $J_M(A, Y, X, N_\beta)$, along with the corresponding original MCML criterion using all pairs, $J_M(A, X)$, are shown for two different settings for (α, β) on the left hand side of this figure.

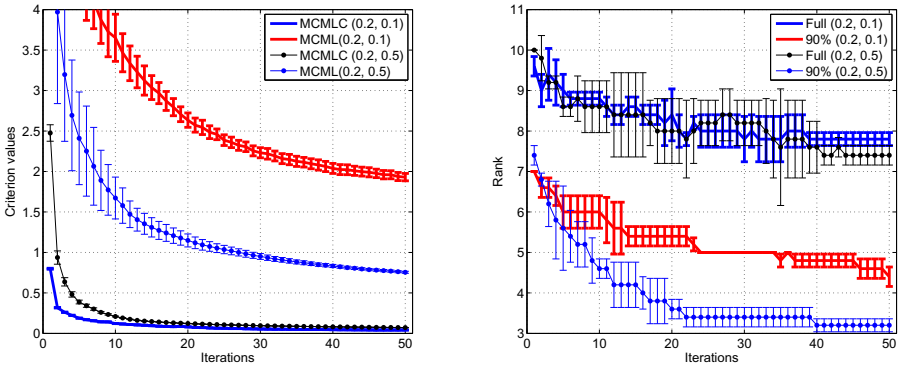


Fig. 2. Criterion (left) and rank (right) values obtained at each iteration when using algorithm MCMLC on the wine dataset. The original MCML criterion is shown together with the MCMLC one for two different (α, β) settings. Strict (or full) rank is shown together with the rank when keeping only the largest eigenvalues up to 90% of the total.

Even noting that absolute values are not directly comparable, we see that the new criteria closely follow the behavior of the original one although it does not necessarily optimize it. As expected, we also see that higher values of β lead to (significantly) smaller values of the original criterion. A subproduct of the new proposal is a reduced variability and consequently the possibility of faster convergence. However, this advantage has not been fully exploited in the present work.

On the right hand side of Figure 2, the values of the ranks of the (PSD version of the) metric matrix with iterations is plotted. The same rank after keeping only the directions that correspond to 90% of the eigenvalues is also shown. As was previously put forward for the MCML algorithm [12, 18], the (full) ranks slowly decrease to arrive at the optimum. Moreover, this decrease does not depend much on the parameter β. On the other hand, a larger sparseness of the metric matrices is observed for greater values of β. If we restrict the ranks in the same way, we observe a significant decrease for higher values of β.

Comparative experiments using MCMLA and MCMLC with the different settings mentioned have also been carried out. The corresponding performance measures on four of the databases are shown in Figure 3. In these plots, the averaged leaving one out error rate estimate corresponding to the nearest neighbor classifier is displayed with regard to the relative number of constraints effectively processed by each algorithm at each iteration (in a logarithmic scale). That is, α for MCMLA(α) and the product $\alpha \cdot \beta$ for MCMLC(α, β). This relative number is an accurate estimate of the computational burden of each algorithm.

The results for the original MCML algorithm are not shown but they are in all cases indistinguishable from the ones obtained for MCMLA($\frac{1}{2}$). Looking at the curves in Figure 3, we see that it is possible to reproduce the behavior of

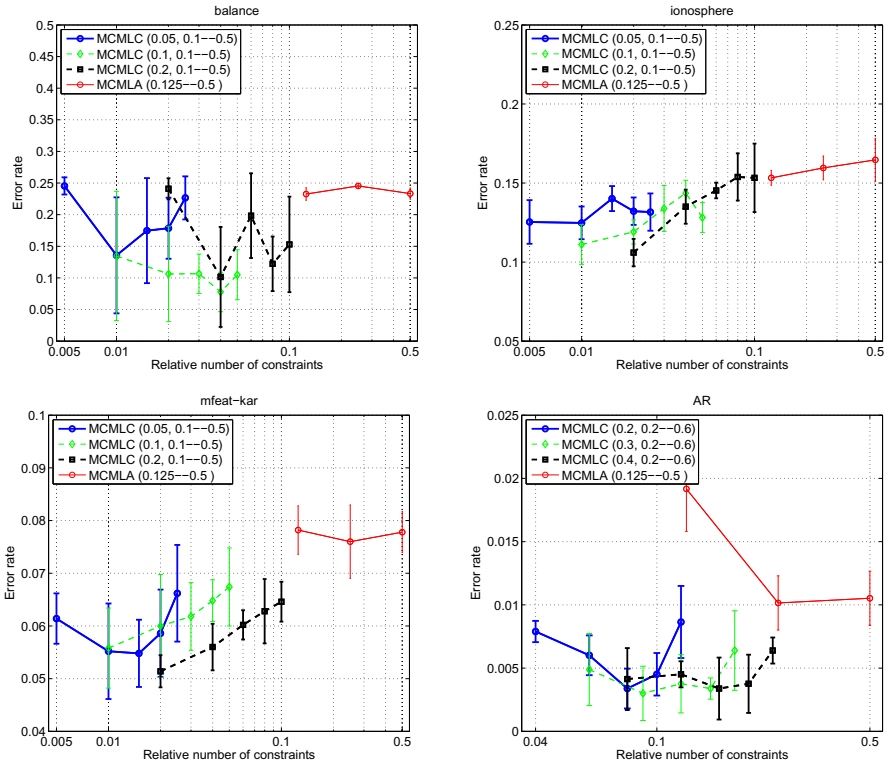


Fig. 3. Performance obtained for the metric learning algorithm with different number of anchors, MCMLA(α), and the proposed generalization with different number of prototypes, MCMLC (α, β). The leaving one out estimate of the 1-NN classifier is shown against the relative number of constraints each method uses (that is, α and $\alpha \cdot \beta$, respectively).

the original MCML algorithm at 50 and 25% of its cost. When using $\alpha = \frac{1}{c}$ (12% in computational cost), the behavior of MCMLA begins to deteriorate in the two larger databases and very significantly in the case of AR. On the other hand, MCMLC with different settings is not only able to reproduce the original behavior but also to improve it. It can be seen that in most of the cases there is a tradeoff between reducing the two parameters and improving the result. In general, we have observed very good results when using from 1% to 5% of the original constraints (around 10% in the case of AR).

5 Concluding Remarks

An empirical evaluation of an extension of a metric learning algorithm that includes prototype generation and adaptation has been considered. The proposed

approach is able to improve both the quality of the metrics obtained and the computational efficiency of the method by significantly reducing the effective number of constraints effectively taken into account at each gradient step.

Some interesting facts and also some critical points have been discovered in this work. Amongst the bad news, the tuning of these methods is not trivial and it is not easy to automate. In fact, more experimentation is needed prior to establishing whether an optimal tradeoff between the two parameters introduced can be found. On the other hand, we have observed that the proposed method leads to good results for a relatively wide range of its parameters.

Apart from consolidating some of the findings of the present work, efforts are also being currently directed towards improving the behavior of the algorithm by forcing and maintaining the sparseness of the metric matrix. A more ambitious line of research tries to formulate some of the ideas in the present work in a more general way. This would make possible to apply them to other metric learning algorithms.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley and Sons (2001)
2. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: A review. IEEE Trans. Pattern Anal. Mach. Intell. 22, 4–37 (2000)
3. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press (1990)
4. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 3, 71–86 (1991)
5. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. 19, 711–720 (1997)
6. Chen, L., Liao, H.M., Ko, M., Lin, J., Yu, G.: A new LDA-based face recognition system which can solve the small sample size problem. Pattern Recognition 33, 1713–1726 (2000)
7. Cevikalp, H., Neamtu, M., Wilkes, M., Barkana, A.: Discriminative common vectors for face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 4–13 (2005)
8. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.J.: Distance metric learning with application to clustering with side-information. In: NIPS, pp. 505–512 (2002)
9. Paredes, R., Vidal, E.: Learning weighted metrics to minimize nearest-neighbor classification error. IEEE Trans. Pattern Anal. Mach. Intell. 28, 1100–1110 (2006)
10. Yu, J., Amores, J., Sebe, N., Radeva, P., Tian, Q.: Distance learning for similarity estimation. IEEE Trans. Pattern Anal. Mach. Intell. 30, 451–462 (2008)
11. Kulis, B.: Metric learning: A survey. Foundations and Trends in Machine Learning 5, 287–364 (2013)
12. Globerson, A., Roweis, S.: Metric learning by collapsing classes. Neural Information Processing Systems (NIPS 2005) 18, 451–458 (2005)
13. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: ICML, pp. 209–216 (2007)

14. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research 10, 207–244 (2009)
15. Wang, J., Woznica, A., Kalousis, A.: Learning neighborhoods for metric learning. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) ECML PKDD 2012, Part I. LNCS, vol. 7523, pp. 223–236. Springer, Heidelberg (2012)
16. Micó, L., Oncina, J., Vidal, E.: A new version of the nearest-neighbour approximating and eliminating search algorithm (aesa) with linear preprocessing time and memory requirements. Pattern Recognition Letters 15, 9–17 (1994)
17. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
18. Perez-Suay, A., Ferri, F.: Scaling up a metric learning algorithm for image recognition and representation. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Porikli, F., Peters, J., Klosowski, J., Arns, L., Chun, Y.K., Rhyne, T.-M., Monroe, L. (eds.) ISVC 2008, Part II. LNCS, vol. 5359, pp. 592–601. Springer, Heidelberg (2008)
19. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007)
20. Duin, R.P.W.: Prtools - version 3.0 - a matlab toolbox for pattern recognition. In: Proc. of SPIE, p. 1331 (2000)
21. Martinez, A., Benavente, R.: The AR face database. Technical Report 24, Computer Vision Center, Barcelona (1998)
22. van der Maaten, L., Postma, E., , van den Herik, H.: Dimensionality reduction: A comparative review. Technical report, Tilburg University, TiCC-TR 2009-005 (2009)

Tracking System with Re-identification Using a RGB String Kernel

Amal Mahboubi[1], Luc Brun[1],
Donatello Conte[2], Pasquale Foggia[3], and Mario Vento[3]

[1] GREYC UMR CNRS 6072, Equipe Image ENSICAEN
6, boulevard Maréchal Juin F-14050 Caen, France
amal.mahboubi@unicaen.fr, luc.brun@ensicaen.fr
[2] Université François-Rabelais de Tours, LI EA 6300
64, Avenue Jean Portalis, F-37200, Tours, France
donatello.conte@univ-tours.fr
[3] Dipartimento di Ingegneria dell'Informazione,
Ingegneria Elettrica e Matematica Applicata
Università di Salerno, Via Ponte Don Melillo, 1 I-84084 Fisciano (SA), Italy
{pfoggia,mvento}@unisa.it

Abstract. People re-identification consists in identifying a person that comes back in a scene where it has been previously detected. This key problem in visual surveillance applications may concern single or multi camera systems. Features encoding each person should be rich enough to provide an efficient re-identification while being sufficiently robust to remain significant through the different phenomena which may alter the appearance of a person in a video. We propose in this paper a method that encodes people's appearance through a string of salient points. The similarity between two such strings is encoded by a kernel. This last kernel is combined with a tracking algorithm in order to associate a set of strings to each person and to measure similarities between persons entering into the scene and persons who left it.

Keywords: Re-identification, String kernel, Visual surveillance.

1 Introduction

The purpose of re-identification is to identify people coming back into the field of view of a camera. Several types of features including interest point [9,2], histograms [3,10,16,8], shape [6] and graph based representations [17,11,13,2], have been proposed in the literature. However, some features like histograms do not encode any spatial information while some others like interest point and graph based representations may induce a matching step that requires important execution times. Moreover complex features like bags or graphs [13] of interest points, RAG [2] may be sensitive to the evolution of the appearance of a person in a video due to his displacements or occlusions.

Independently of the type of features used to perform the re-identification step, re-identification methods may be split into two categories: methods of the

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 333–342, 2014.

first group [9] compute a unique signature for each object and perform the re-identification based on this single signature. Methods of the second group [3,18] delay the re-identification that is then performed on a set of signatures. Using a method of the latter category imposes to base the comparison between two objects on a comparison of two sets of signatures rather than between two single signatures. However such an approach can potentially better capture the variability of the appearance of a person over a video.

Our approach belongs to the second category and describes the appearance of a person by a set of RGB string descriptors (Section 2) computed over a sliding window. A kernel between two sets of strings (Section 3) is then applied in order to encode the similarity between two persons. The integration of this kernel into a tracking method is described in Section 4 while Section 5 reports several experiments that demonstrate the validity of our approach.

2 RGB String Descriptor Construction Scheme

One of the main challenge in people re-identification is to capture peoples' appearance properties. As mentioned in Section 1, several modelings have been developed. However, although complex models such as graph based representation offer the advantage of a precise modeling of an object, they usually require a complex matching step and important execution times. An alternative solution consists in using a string descriptor. Indeed, a string allows an effective comparison while preserving useful information of the region of interest. Although a string usually encodes less information than a graph, we expect a greater stability of this simpler structure over time.

The first step of our method consists in separating subjects from the background. To that end, we use binary object masks [2] defined by a foreground detection with shadow removals. Each moving person within a frame is thus associated to a mask and to a bounding box characterized using a salient string. Each character of this last string is defined by a couple of coordinates (x,y) and the associated RGB image's color. The construction of a salient string is outlined in Figure 1. This construction consists of the following 3 steps:

First, we apply a Deriche edge detector on each moving person according to its binary mask within a frame.

Then, for the second stage we build a discriminating curve of the object using contour points provided by the Deriche detector. Let us consider the bounding box $W \times H$ of an object obj_a whose top-left corner's coordinates are denoted (tl_x,tl_y). Thanks to the Deriche filter, obj_a, should be delineated by two main contours (Figure 1-step-1). For each value $h \in \{0, \ldots, H-1\}$, we consider the horizontal line segment defined as the intersection between the bounding box of obj_a and the line $y_h = h + tl_y$. The x coordinate of the central point of obj_a at height y_h is denoted $\overline{x}_h$ and is defined as the x coordinate of the weighted mean of all points along the line segment. More precisely, $\overline{x}_h$ is defined as:

$$\forall h \in \{0, \ldots, H-1\} \quad \bar{x}_h = \frac{\sum_{w=0}^{W} |\nabla I(x_w, y_h)|^2 . x_w}{\sum_{w=0}^{W} |\nabla I(x_w, y_h)|^2}, \tag{1}$$

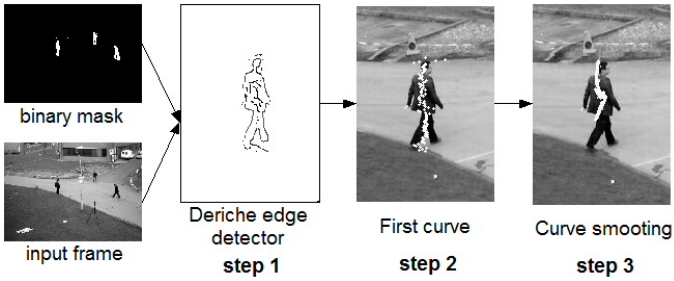


Fig. 1. RGB string construction steps

where $x_w = tl_x + w$, $I(x_w, y_h)$ denotes the pixel's value of (x_w, y_h) and $|\nabla I(x_w, y_h)|$ is the amplitude of its gradient.

The last and third step, enforces the quality of the resulting curve. Indeed, our resulting curve (Figure 1-step-2) is sensible to small perturbations of the gradient on each line and contains important discontinuities. This last point may alter the similarity of two curves of a same person taken on two different frames. We thus propose to regularize this curve through an energy minimization framework. Our energy function (equation 2) combines two terms: the former encodes the attachment to the initial curve $(\bar{x}_h, y_h)$. The latter is a regularization term, which enforces the continuity of the curve. Hence, we assume that, the energy functional of the curve $c = (x^*_h, y_h)_{h \in \{0,\dots,H-1\}}$ is defined as follows:

$$\jmath(c) = \sum_{h=1}^{H} (\bar{x}_h - x^*_h)^2 + \lambda (x^*_h - x^*_{h-1})^2 \,, \tag{2}$$

where λ is a tuning parameter. The average coordinate $\bar{x}_h$ is given by equation 1 and x^*_h is the corresponding final coordinate.

Minimization of equation 2 leads to search for the zeros of its gradient:

$$\frac{\partial \jmath}{\partial x^*_h} = 2(1 + 2\lambda) x^*_h - 2\lambda (x^*_{h-1} + x^*_{h+1}) - 2\bar{x}_h = 0 \,. \tag{3}$$

Equation 3 corresponds to the formulation of a tridiagonal system which can be solved in $\mathcal{O}(n)$. This last minimization step obtained using equation 3 provides the final curve $c = (x^*_h, y_h)_{h \in \{0,\dots,H-1\}}$, where each point is associated to its RGB value (Figure 1-step-3).

3 People Description

Curves encoding of people's appearance may be altered by the addition of erroneous extremities; encoding, for example, a part of the floor or a difference of sampling due to the variations of the distance between a person and the camera. In order to cope with such variations we consider each curve as a string

and encode the similarity between two strings using the global alignment kernel defined by [4]:

$$K_{GA}(s_1, s_2) = \sum_{\pi \in A(n,m)} e^{-D_{s_1,s_2}(\pi)}\,, \tag{4}$$

where n and m, denote the length of the first string s_1 and the second string s_2 respectively. An alignment is noted π and $A(n, m)$ represents the set of all alignments between s_1 and s_2. The symbol D denotes the Dynamic Time Warping distance. It measures the discrepancy between two strings s_1 and s_2 according to an alignment π. Function D is defined [4] as:

$$D_{s_1,s_2}(\pi) = \sum_{i=1}^{|\pi|} \varphi(x_{\pi_1(i)}, y_{\pi_2(i)})\,, \tag{5}$$

where $s_1 = (x_i)_{i\in\{1,..,n\}}$, $s_2 = (y_i)_{i\in\{1,...m\}}$ and function φ corresponds to a distance function defined [4] as follows:

$$\varphi(x, y) = \frac{1}{2\sigma^2}\|x - y\|^2 + \log(2 - e^{-\frac{\|x-y\|^2}{2\sigma^2}})\,, \tag{6}$$

where x and y denote the RGB values of the first object and the second object respectively. Symbol σ denotes a tuning parameter. The log term is added to the squared Euclidean distance $\|x-y\|^2$ in order to ensure the definite positiveness of K_{GA} (equation 4) [4]. Note that, using equation 5, equation 4 may be computed using a slightly modified version of the classical string edit distance algorithm. The computational complexity of equation 4 is thus bounded by $\mathcal{O}(nm)$ where n and m denote respectively the length of s_1 and s_2.

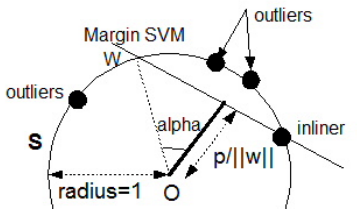


Fig. 2. Geometrical interpretation of equation 7

3.1 People's Kernel

As the appearance of a person evolves in a scene, due to slight changes of the pose, the use of a single string is inappropriate to identify a person. Assuming that, the appearance of a person is established in a set of successive frames, we thus describe each person by a set of salient strings. The temporal window over which this set is built is called the history tracking window (HTW).

Each person in the video is hence not defined by a single string but by a set of strings (on HTW). This set may include outlier strings, due to slight changes of the pose or occlusion. The construction of a representative string based on a simple average of all the strings of a set (in the Hilbert space defined by the kernel) may be sensible to such outliers. We thus suggest to enforce the robustness of our representative string through the use of an one class SVM classifier.

Let $\mathcal{H}$ denotes the Hilbert space defined by K_{GA} (equation 4). In order to get a robust model encoding the mean appearance of a person, we first use K_{GA} to project the mapping of all strings onto the unit-sphere of $\mathcal{H}$. This operation is performed by normalizing our kernel [5]. Following [5], we then apply a one class ν-SVM on each set of strings describing a person. From a geometrical point of view, this operation is equivalent to model the set of projected strings by a spherical cap defined by a weight vector w and an offset ρ both provided by the ν-SVM algorithm. These two parameters define the hyper plane whose intersection with the unit sphere defines the spherical cap. Strings whose projection on the unit sphere lies outside the spherical cap are considered as outliers. Each person is thus encoded by a triplet (w, ρ, S) where S corresponds to the set of strings and (w, ρ) are defined from a one class ν-SVM. Figure 2 gives the geometric interpretation of (w, ρ); The parameter w indicates the center of the spherical cap and may be intuitively understood as the vector encoding the mean appearance of a person over its HTW window. The parameter ρ influence the radius of the spherical cap and may be understood as the extend of the set of representative strings in S.

Let $P_A = (w_A, \rho_A, S_A)$ and $P_B = (w_B, \rho_B, S_B)$ denote two triplets encoding two persons A and B. The distance between A and B is defined from the angle between vectors w_A and w_B defined as follows [5]:

$$d_{sphere}(w_A, w_B) = arccos\left(\frac{w_A^T K_{A,B} w_B}{\|w_A\|\|w_B\|}\right) \tag{7}$$

where $\|w_A\|$ and $\|w_B\|$ denote the norms of w_A and w_B in $\mathcal{H}$ and $K_{A,B}$ is a $|S_A| \times |S_B|$ matrix defined by $K_{A,B} = (K_{norm}(t, t'))_{(t,t') \in S_A \times S_B}$, where K_{norm} denotes our normalized kernel.

Based on d_{sphere}, the kernel between A and B is defined as the following product of RBF kernels:

$$K_{change}(P_A, P_B) = e^{\frac{-d^2_{sphere}(w_A, w_B)}{2\sigma^2_{moy}}} e^{\frac{-(\rho_A - \rho_B)^2}{2\sigma^2_{origin}}}, \tag{8}$$

where σ_{moy} and σ_{origin} are tuning variables.

4 Re-identification

Our tracking algorithm is based on a previous work [13]. The tracking algorithm uses four labels 'new', 'get-out', 'unknown' and 'get-back' with the following

Algorithm 1. Algorithm for re-identification. Note: frame is a collection of object

```
 1: DB_o= φ ;
 2: while ( not at end of this video ) do
 3:     for each (object in pastFrame ) do
 4:         if ((object not in currentFrame) AND (object.duration ≥ HTW)) then
 5:             DB_o.Insert(video,object);
 6:         end if
 7:     end for
 8:     for each (object in currentFrame) do
 9:         object.computeString();                                  ▷ section 2
10:         if (object not in pastFrame) then
11:             object.label="unknown";
12:             object.duration=0;
13:             if (DB_o is empty) then
14:                 object.label="new";
15:                 object.duration=1;
16:             end if
17:         end if
18:         if (object.label=="unknown") then
19:             object.duration++;
20:             if (object.duration== HTW) then
21:                 var integer index=-1;
22:                 computeSimilarity(DB_o,object,index);
23:                 if (index ≥ 0) then
24:                     object.label="get-back";
25:                     DB_o.delete(index);
26:                 else
27:                     object.label="new";
28:                 end if
29:             end if
30:         end if
31:     end for
32: end while
```

meaning: '*new*' refers to an object classified as new, '*get-out*' represents an object leaving the scene, '*unknown*' describes a query object (an object recently appeared, not yet classified) and '*get-back*' refers to an object classified as an old one after a re-identification step. All masks detected in the first frame of a video are considered as new persons. The proposed re-identification approach is depicted in Algorithm 1:

- 'get-out' processing (lines 3 to 7): when an object leaves the scene its triplet $P = (w, \rho, S)$ (Section 3) computed over the last $|HTW|$ frames is stored in an output object data base noted DB_o.
- 'new' processing (lines 10 to 17): when an 'unknown' person is found and DB_o is empty we label this 'unknown' person as new.

– 'unknown' processing (lines 18 to 30) when an 'unknown' person is found and DB_o is not empty we should postpone the identification of this 'unknown' person; The 'unknown' person is tracked on $|HTW|$ frames in order to have its description by a triplet (w, ρ, S). Using this description we calculate the value of kernel K_{change} (equation 8) between this 'unknown' person and all 'get-out' persons present in our database (line 22). Similarities between the 'unknown' person and the 'get-out' persons are sorted in decreasing order so that the first 'get-out' person of this list corresponds to the best candidate for a re-identification. Our criterion to map an 'unknown' person to 'get-out', and thus to label it as 'get-back' is based on a threshold on the first two maximal similarity values max_{ker} and max_2 of the list of similarities ($max_2 \leq max_{Ker}$). This criterion called, SC is defined as $max_{ker} > th_1$ and $\frac{max_2}{max_{Ker}} < th_2$, where th_1 and th_2 are experimentally fixed thresholds.

Classically, any tracking algorithm has to cope with many phenomena such as occlusions. In this paper we limit the study to overlapping bounding boxes. When an overlap greater than an experimentally fixed threshold occurs between two bounding boxes, an occlusion is found. We assume two kinds of occlusions: partial occlusions; where the occluded object remains visible and severe occlusions; where the occluded object is completely hidden. If two or more objects (detected at time t) merge together (at time $t+1$) to form one new object, this object is deemed to be a group rather than an occlusion. Group cases are not considered in this work.

5 Experiments

The proposed algorithm has been tested on v01 and v05 video sequences of the PETS'09 S2L1 dataset[1]. Each sequence contains multiple persons and occlusion cases. We have to notice that all the tuning parameters used in this work are set by cross-validation. As the used dataset does not contains a training set, the tuning parameters are set using the test dataset.

In our first experiment we have evaluated how different values of the length of HTW affect the re-identification accuracy. Figure 3 shows the effects for HTW changes on the true positive measurement for each view. The obtained results show that, v01 performs at peak efficiency for HTW=30. Video v05 reach its optimum at HTW=20. These curves also show that the length of HTW is not a crucial parameter of our method.

In a second experiment we show the improvement of the proposed kernel with respect to a histogram based approach. Similarly to [15] where histograms are defined from the already extracted blob (segmented parts), we propose the following histograms construction scheme: color histograms are computed on HTW frames for both the query object and each 'get-out' persons contained in DB_o. Then, we try to map the query object with one of the 'get-out' objects already stored in DB_o using EMD distance [14] between histograms. If a map is found,

[1] Available at `http://www.cvg.rdg.ac.uk/PETS2009/a.html`

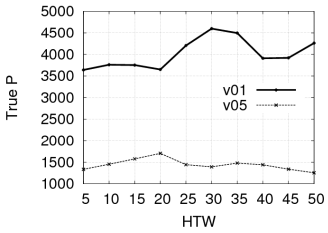


Fig. 3. HTW effects

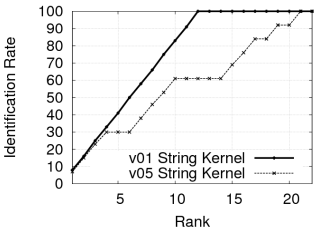


Fig. 4. CMC curves

the query object gets the label of the mapped 'get-out' object, and we update DB_o. Otherwise, we create a new label for the query object. The map criterion used here is similar to the above SC criterion (Section 4), nevertheless the best candidate corresponds to the minimum since we use distances. Table 1 reports the comparison results between histogram-based, kernel-based approaches and graph approach of [2]. As it can be seen from Table 1, the performance of the proposed kernel is superior to the histogram method as expected. Furthermore, the results of the proposed kernel give lower values than the graph approach regarding v05, while the results are clearly better for v01. We attribute this to the high detection accuracy in v01. Furthermore, v05 contains a lot of severe occlusions that are not specifically addressed in the proposed method. Indeed on such severe occlusions a large part of a person is usually hidden by another person.

To validate our method of re-identification we used the Cumulative Matching Characteristic (CMC) curves. The CMC curve represents the percentage of times the correct identity match is found in the first n matches. Figure 4 shows the CMC curves for the two views. We can see that the performance of v01 is much better than that of v05. This last result being due to the large number of occlusions occurring in v05.

In a third experiment, we assess the statistical performance of the proposed kernel. To this end, we iteratively shift the beginning of the video by a unit of 50 frames to obtain 15 subsets of the original video i.e., $t_0 \in [50, 750]$. We thus reported for each set the TrueP and the FalseP values. Finally we computed the mean (avg) and the standard deviation (σ) of the obtained results. As expected, the results are coherent with Table 1 (line 2) since the resulting couples (avg, σ) regarding TrueP are equal to $(0.90, 0.08)$ and $(0.66, 0.165)$ for v01 and v05 respectively.

Table 1. Comparison results

	view01		view05	
	TrueP	FalseP	TrueP	FalseP
histogram	51.46%	48.54%	49.91%	50.09%
kernel	100%	0%	62.61%	33.39%
graph of [2]	81%	7%	83%	13%

Table 2. Evaluation results

View	work of [1]	current work		
	MODA	MODA	MOTA	SFDA
v01	0.67	0.97	0.97	0.91
v05	0.72	0.60	0.60	0.81

We also used the exhaustive comparison of 13 methods defined in [7] in order to compare our results to the state of the art. The study [7] did a quantitative evaluation of the results submitted by contributing authors of the two PETS workshops in 2009 on PETS'09 S2.L1 dataset. We noticed that the submitted results of [1] outmatch all other methods using the MODA, MOTA, MODP, MOTP SODA and SFDA metrics described in [12]. Therefore, we only compare our results to that last method. Table 2 depicts the following: the left column shows the best results [1] obtained by methods described in [7] on each video. The second column of Table 2 shows that our method obtains a lower MODA index than [1] on v05 but clearly outperform this last method on v01. These results may again be explained by the high number of occlusions in v05 which are overcomed by [1] using multiple views of each person while the present method is restricted to a single view. These results indicate thus the relevance of the proposed re-identification method when objects are not severely occluded.

6 Conclusion

In this work, we addressed the people re-identification problem by proposing a new approach based on RGB string kernels. A benchmark public dataset was used to validate our method. Our results show that the proposed approach outperforms state-of-the art methods when few severe occlusions occur.

Our future research will focus on the investigation of occlusion and group problems still using a single camera. To handle these phenomena, we should for each object severely occluded or entering into a group, suspend the update of its curves for the frames where it is hidden. Indeed in such cases no reliable feature may be extracted to characterize hidden persons.

References

1. Berclaz, J., Shahrokni, A., Fleuret, F., Ferryman, J., Fua, P.: Evaluation of probabilistic occupancy map people detection for surveillance systems. In: Proceedings of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), pp. 55–62 (2009)
2. Brun, L., Conte, D., Foggia, P., Vento, M.: People re-identification by graph kernels methods. In: Jiang, X., Ferrer, M., Torsello, A. (eds.) GbRPR 2011. LNCS, vol. 6658, pp. 285–294. Springer, Heidelberg (2011)
3. Cong, D.N.T., Khoudour, L., Achard, C., Meurie, C., Lezoray, O.: People re-identification by spectral classification of silhouettes. Signal Processing 90(8), 2362–2374 (2010)
4. Cuturi, M.: Fast global alignment kernels. In: Getoor, L., Scheffer, T. (eds.) ICML, pp. 929–936. Omnipress (2011)
5. Desobry, F., Davy, M., Doncarli, C.: An online kernel change detection algorithm. IEEE Transactions on Signal Processing 53(8-2), 2961–2974 (2005)
6. Doretto, G., Sebastian, T., Tu, P., Rittscher, J.: Appearance-based person reidentification in camera networks: problem overview and current approaches. Journal of Ambient Intelligence and Humanized Computing 2, 127–151 (2011)

7. Ellis, A., Shahrokni, A., Ferryman, J.M.: Pets2009 and winter-pets 2009 results: a combined evaluation. In: 2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Snowbird, USA, Conference held December 7-9. IEEE (2009)
8. Farenzena, M., Bazzani, L., Perina, A., Murino, V., Cristani, M.: Person re-identification by symmetry-driven accumulation of local features. In: Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2010). IEEE Computer Society, San Francisco (2010)
9. Hamdoun, O., Moutarde, F., Stanciulescu, B., Steux, B.: Person re-identification in multi-camera system by signature based on interest point descriptors collected on short video sequences. In: ICDSC, pp. 1–6. IEEE (2008)
10. Ijiri, Y., Lao, S., Han, T.X., Murase, H.: Human re-identification through distance metric learning based on jensen-shannon kernel. In: VISAPP (1), pp. 603–612 (2012)
11. Iodice, S., Petrosino, A.: Person re-identification based on enriched symmetry salient features and graph matching. In: Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Rodríguez, J.S., di Baja, G.S. (eds.) MCPR 2012. LNCS, vol. 7914, pp. 155–164. Springer, Heidelberg (2013)
12. Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., Zhang, J.: Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. Pattern Analysis and Machine Intelligence 31(2), 319–336 (2009)
13. Mahboubi, A., Brun, L., Conte, D., Foggia, P., Vento, M.: Tracking system with re-identification using a graph kernels approach. In: Wilson, R., Hancock, E., Bors, A., Smith, W. (eds.) CAIP 2013, Part I. LNCS, vol. 8047, pp. 401–408. Springer, Heidelberg (2013)
14. 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), 99–122 (2000)
15. Satta, R., Fumera, G., Roli, F., Cristani, M., Murino, V.: A multiple component matching framework for person re-identification. In: Maino, G., Foresti, G.L. (eds.) ICIAP 2011, Part II. LNCS, vol. 6979, pp. 140–149. Springer, Heidelberg (2011)
16. Schwartz, W.R., Davis, L.S.: Learning Discriminative Appearance-Based Models Using Partial Least Squares. In: Brazilian Symposium on Computer Graphics and Image Processing, pp. 322–329 (2009)
17. Wang, X., Doretto, G., Sebastian, T., Rittscher, J., Tu, P.H.: Shape and appearance context modeling. In: ICCV, pp. 1–8. IEEE (2007)
18. Zhao, S., Precioso, F., Cord, M.: Spatio-temporal tube data representation and kernel design for svm-based video object retrieval system. Multimedia Tools Appl. 55(1), 105–125 (2011)

Towards Scalable Prototype Selection by Genetic Algorithms with Fast Criteria

Yenisel Plasencia-Calaña[1,2], Mauricio Orozco-Alzate[3], Heydi Méndez-Vázquez[1], Edel García-Reyes[1], and Robert P.W. Duin[2]

[1] Advanced Technologies Application Center, 7ma A ♯ 21406, Playa, Havana, Cuba
{yplasencia,egarcia,hmendez}@cenatav.co.cu
[2] Faculty of Electrical Engineering, Mathematics and Computer Sciences, Delft University of Technology, The Netherlands
r.duin@ieee.org
[3] Departamento de Informática y Computación, Universidad Nacional de Colombia - Sede Manizales. Kilómetro 7 vía al Aeropuerto, Campus La Nubia – Bloque Q, Piso 2, Manizales, Colombia
morozcoa@unal.edu.co

Abstract. How to select the prototypes for classification in the dissimilarity space remains an open and interesting problem. Especially, achieving scalability of the methods is desirable due to enormous amounts of information arising in many fields. In this paper we pose the question: are genetic algorithms good for scalable prototype selection? We propose two methods based on genetic algorithms, one supervised and the other unsupervised, whose analyses provide an answer to the question. Results on dissimilarity datasets show the effectiveness of the proposals.

Keywords: dissimilarity space, scalable prototype selection, genetic algorithm.

1 Introduction

The vector space representation is a common option to represent the data for learning tasks since many statistical techniques are applicable for this kind of representation. However, there is an increasing number of real-world problems which are not vectorial. Instead, the data are given in terms of pairwise dissimilarities which may be non-Euclidean and even non-metric. In [1] several approaches were presented to learn from dissimilarity data, where the dissimilarity space (DS) has several advantages over the other approaches. In the DS approach, the dissimilarities of the training objects to the representation set are interpreted as coordinates in the space. Class separability is a property that one wants to maintain after the mapping which can be accomplished by a careful selection of the prototypes. However, a random selection was found to perform well for large numbers of prototypes [2]. Since this is the fastest method, the selection of good prototypes by more dedicated methods is of interest only for small numbers of prototypes. A good method must be able to find a minimal set without a significant decrease in accuracy of classifiers in the DS.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 343–352, 2014.

Several methods have been proposed [2,3] to find small representation sets by supervised or unsupervised strategies. Supervised methods have the advantage of maintaining a high accuracy, however this is achieved at high computational costs; besides, they might suffer from overfitting. Unsupervised methods have the advantages of being fast, generalizing well and avoiding overfitting. However, as a disadvantage, they are not always good in maintaining class separability since class labels are not taken into account. When the purpose is to learn from large datasets, the scalability of the method must also be considered since it is known that the prototype selection problem is NP-complete. This has been overlooked so far and only some studies such as the one in [4] copes with the problem. Large datasets arise in several situations and being able to deal with them is of interest. Some of the issues that cause the existence of large datasets are: vast amounts of data due to dropping costs for capturing, transmitting, processing and storing; modalities that have millions of classes such as biometrics; and modalities that have hundreds of thousands samples such as brain tractography data in [4].

This paper is aimed to study if genetic algorithms (GAs) are good for scalable prototype selection and if a fast clustering replacing the random initialization can help to improve the results further. It is usually assumed in the literature that linear-time algorithms are acceptable for scaling up to large datasets [5]. We also adopt this assumption. The parameter that dominates the complexity in our problem is the number of samples, since we assume that the other parameter, the number of selected prototypes, will always be small and that the full dissimilarity matrix is already computed. Some strategies to cope with scalability include parallelism, techniques to work with data that do not fit into memory, stochastic methods, etc. However in this study we focus on the time complexity and, in case the full dissimilarity matrix do not fit into memory, the developed approaches are easily adaptable by loading the dissimilarities on demand.

In [6], a GA was proposed for prototype selection but it is not able to cope with scalability issues. Here, we develop two versions of a scalable GA that optimize two different criteria for prototype selection. The remaining part of the paper is organized as follows: Section 2 presents the dissimilarity representation and prototype selection, Section 3 presents the two proposed methods. Experimental results are reported and discussed in Section 4 and conclusions are drawn in Section 5.

2 Dissimilarity Space and Prototype Selection

The dissimilarity space was proposed by Pekalska and Duin [1]. It was postulated as a Euclidean vector space, which allows the use of several classifiers. Let X be the space of objects which may not be vectorial, let $R = \{r_1, r_2, ..., r_k\}$ be the set of prototypes such that $R \in X$, and let $d : X \times X \mapsto \mathbb{R}^+$ be a suitable dissimilarity measure for the problem. The prototypes may be chosen based on some criterion or even at random; however, the goal is that they have good representation capabilities specially when pursuing small representation sets. For a finite training set $T = \{x_1, x_2, ..., x_n\}$ such that $T \in X$, the dissimilarity space is created by the data dependent mapping $\phi_R^d : X \mapsto \mathbb{R}^k$ where:

$$\phi_R^d(x_i) = [d(x_i, r_1)\ d(x_i, r_2)\ \ldots\ d(x_i, r_k)]. \quad (1)$$

For dissimilarity representations, the adaptation of prototype selection techniques for improving the k-nearest neighbour (k-NN) classifier as well as the adaptation of clustering techniques have been investigated showing good results [2]. Other approaches use the geometry and the distribution of the objects to find the prototypes [7]. In [2], various techniques were compared such as Kcentres, mode seeking, forward selection (FS), linear programming, editing and condensing, and a mixture of Kcentres with linear programming. These techniques showed good performances. However, some of them are computationally expensive for very large datasets such as the FS, which runtime is quadratic in the number of samples.

3 Proposed Methods

We propose two different variants of GA with different scalable criteria for prototype selection. The two methods receive as parameter the desired number of prototypes. Finding an appropriate number of prototypes for each particular problem is out of the scope of this paper, however for this purpose some methods can be applied. For example, a good practice is to find the intrinsic dimensionality and select the number of prototypes accordingly. We assume we have a square dissimilarity matrix D among all training samples; the prototypes will be selected from this set of samples. Note that our goal is not to generate new prototypes as combinations of the original ones, but to select ones that already exist.

The GA is a biologically motivated search method which explores individuals (chromosomes or solutions) created after each generation by the best fitted ones. This property of GAs makes them much better scalable than using a full search. In our problem, each individual is a set of prototypes of fixed cardinality k codified in a $k - vector$ containing in each position the index of the potential prototype. For example the $5 - vector$ $(65, 30, 7, 19, 87)$ codifies an individual representing a set of 5 potential prototypes which can be accessed in some data structure by the indexes 65, 30 and so on. The GA starts the search in an initial population of randomly generated individuals.

Before executing the GA, we propose to perform a 1-step K-centres clustering in the space of candidates, where the number of clusters equals the desired number of prototypes. The candidates are clustered in order to guide the GA search in such a way that it has a faster convergence. The clustering runtime is $O(nk)$, where $n = |T|$, being T the training/validation set and $k = |R|$. The GA is slightly modified since its initial population is now generated by randomly sampling one potential prototype per cluster. Therefore, each element of an individual (also named gene) is linked to a particular cluster since only objects from that cluster are allowed in the corresponding position of the individual. In each generation, the best solution according to the fitness function is found and reproduced with each member of the population with a preset probability by

Algorithm 1. Genetic Algorithm

Input: D: dissimilarity matrix among samples and candidates to prototypes; k: desired number of prototypes, S: number of individuals in the population, rp: reproduction probability, mp: mutation probability, $iter$: number of generations
Output: bestindividual: set of prototypes indexes

```
// perform a 1-step Kcentres clustering in the candidates to
   prototypes to find k clusters
cluslabs ← Kcentres(D, k, 1);
// randomly generate the population ensuring that, in the j-th
   position of the individual, only objects belonging to the j-th
   cluster are allowed
P ← GenerateInitialPopulation(cluslabs, D, k, S);
while number of generations < iter do
    // find the best solution from the population and assign it to
       bestindividual
    foreach currentindividual in P do
        if Fitness(currentindividual, D) > Fitness(bestindividual, D)
        then
            bestindividual ← currentindividual;
        end
    end
    // Evolution cycle
    foreach currentindividual in P do
        // Reproduction, replace a gene of currentindividual with
           probability rp by a gene of the best
        Reproduce(bestindividual, currentindividual, rp);
        // Mutation, change a gene of currentindividual with
           probability mp
        Mutate(currentindividual, mp);
    end
end
```

gene using uniform reproduction or crossover. Elitist selection is performed since the best fitted individual is retained for the next generation without undergoing mutation; in addition, only the best fitted individual is selected as parent of the next population of individuals. The rest of the population undergoes gene mutation with a preset probability which is usually small but also with the constrain that the new index codified in a gene must belong to the related cluster. The pseudo-code is presented in Algorithm 1.

To achieve full scalability these methods should be able to handle: (1) large sets of candidates to prototypes, (2) large number of individuals in the search space of the GA and (3) large number of samples to be used (if needed) to compute the fitness function. In our proposal, the GA handles well (1) large sets of candidates for prototypes since we discarded the standard binary codification of individuals that demands vectors of length equal to n where $n \gg k$. Instead,

we resorted to vectors of length equal to the number of prototypes k since we codify only the indexes of the prototypes to be evaluated. Scalability in the number of individuals to analyze in the search space (2) is achieved since the stopping condition is a small predefined number of GA generations that does not depend on the number of individuals in the search space. A small number of generations is sufficient for GA's convergence thanks to its guided sampling since not all the possible combinations of prototypes are explored but only the best ones which arise after each generation. In addition, the initial clustering helps to avoid redundant prototypes in the same individual. Scalability in the number of samples to be used to compute the fitness (3) will be explained in the next subsections.

3.1 Minimum Spanning Tree Based Unsupervised Criterion

In the fitness function computation, the set of prototypes being evaluated and the training samples are usually involved. However, note that the proper number of prototypes k depends on the intrinsic dimension of the data which is usually small, thereby, $n \gg k$. For large datasets this implies that the dominant term for the fitness computation is the total number of samples n. To achieve scalability in the fitness function it must scale well to a large n. This highly depends on how the criterion to be optimized in the fitness function by the GA is conceived. Our first proposal for GA criterion is based on the minimum spanning tree (MST) of a set of prototypes. The prototypes are interpreted as nodes in a graph and the dissimilarity values between prototypes correspond to edge weights. The sum of edge weights (named tree weight) is used as criterion to be maximized, thereby improving the coverage over the DS. As we used the Prim's algorithm to find the MST and the graph is complete, the computation of this criterion has a runtime of $O(k^2 \log(k))$. Therefore, it is completely independent on the large number of samples n and as a consequence highly scalable for very large problems. The pseudo-code is presented in Algorithm 2.

The total runtime of the proposed GA with this criterion is as follows. For computing the initial clustering of the prototypes the runtime is $O(nk)$, for the fitness function $O(k^2 \log(k))$, and $O(k)$ for mutation and reproduction since each position of the vector representing an individual has to be analyzed. The dominant term in the whole procedure is $O(nk)$, as we assume that the desired number of prototypes is small and fixed, therefore the total runtime is $O(nk)$. However, if the initial clustering step is discarded, the complexity is only $O(k^2 \log(k))$, which in case of needing sub-linear (in n) methods is more appropriate.

3.2 Supervised Criterion Based on Counting Matching Labels

Our second criterion proposal is a linear-time supervised criterion that is different from previous expensive supervised ones [8] since it does not compute a classification error in the DS or an intra-class distance, which are usually quadratic. Our method, instead, considers each candidate to prototype as a representative of a cluster and every object in T is assigned to the cluster represented by its

Algorithm 2. Unsupervised Fitness function by MST

Input: w: vector of prototypes indexes; D: dissimilarity matrix
Output: res: fitness value
// Interpret the prototypes indexed in w as nodes and dissimilarities among them as edges weights of a complete graph G = (V,E)
// compute minimum spanning tree by Prim's algorithm
1 $V' \leftarrow V[1]$;
2 $k \leftarrow |V|$;
3 $E' \leftarrow \emptyset$;
4 **while** $|E'| < k - 1$ **do**
  // select an edge of minimum weight which connects one node in V' with a node which is not in V'
  // add the new edge to E', add the new node to V'
5 **end**
6 $T \leftarrow (V', E')$;
// sum all the weights of edges in E'
7 $res \leftarrow$ SumWeights(E');

Algorithm 3. Supervised Fitness function

Input: w: vector of prototypes indexes; D: dissimilarity matrix
Output: res: fitness value
// interpret the prototypes r_j indexed in w as centers of clusters
1 $res \leftarrow 0$;
2 **foreach** $x \in T$ **do**
  // find the nearest prototype of x
3 $r' \leftarrow$ argmin($D[x, r_j]$);
4 **if** getclasslabel(x)=getclasslabel(r') **then**
5 $res \leftarrow res + 1$;
6 **end**
7 **end**

nearest prototype. The proposed criterion counts the number of assigned objects whose labels match their representative label. The best solution is the one that maximizes this value. This has the smallest runtime for a supervised method that uses all the samples ($O(nk)$). The pseudo-code is presented in Algorithm 3.

The runtime of the whole supervised GA is as follows. For computing the clustering the runtime is $O(nk)$, for the fitness function $O(nk)$, and for mutation and reproduction $O(k)$. The total runtime is $O(nk)$. However, in practice, this is higher than the unsupervised procedure since it is multiplied by a high constant due to the cost for comparing the labels. In general these times are better than or comparable to other linear (in n) algorithms compared in our experiments such as the Kcentres which is $O(nk)$ and the farthest first transversal (FFT) which is also $O(nk)$.

4 Experiments

4.1 Datasets and Experimental Setup

Four different datasets of moderate size were used for the experiments: the Zongker data [9] computed by deformable template matching, Pendigits [10,11] computed by edit distances, XM2VTS [12] computed by chi square distances on LBP histograms, and Diabetes [13] computed using Euclidean distances on features. The characteristics of the datasets as well as the cardinality of the training sets used are summarized in Table 1.

Table 1. Characteristics of the datasets used in this study, the $|T|$ column refers to the training/validation set cardinality used for the experiments

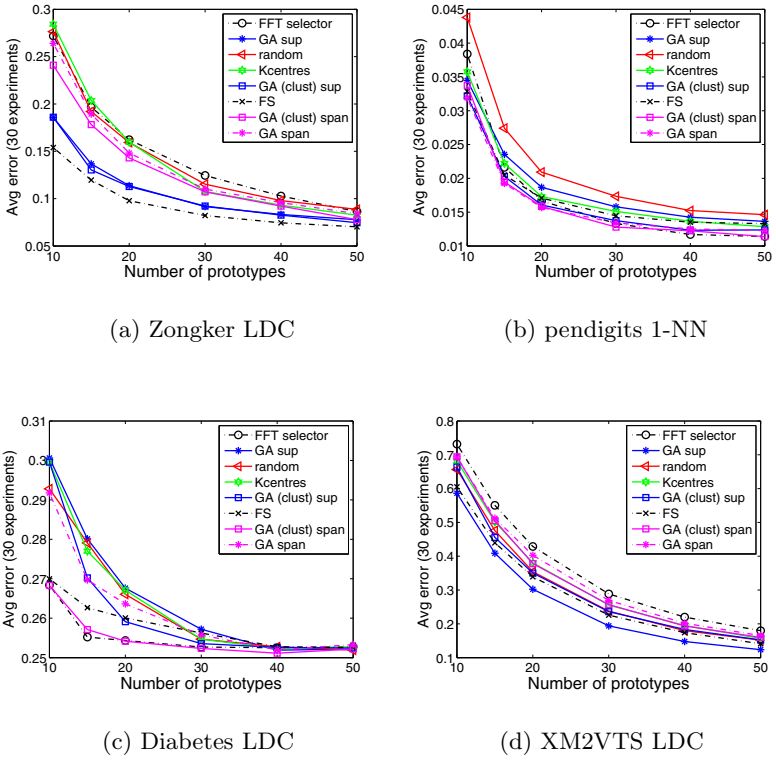
Datasets	# Classes	# Obj	Metric	$\|T\|$
Zongker	10	200×10	no	1000
Diabetes	2	500/268	yes	384
Pendigits	10	10992	no	5000
XM2VTS	295	12×295	yes	1770

The datasets were randomly divided 30 times into training/validation set and test set. The validation set is used to optimize the criteria. The candidates to prototypes also belong to the validation set. The best performing classifier per dataset between the linear discriminant classifier (LDC) and the 1-NN classifier was used to report the classification errors for the different prototype selection methods compared, which are: random selection, FS [2] optimizing the supervised criterion; FFT [4], Kcentres [2], GA in the space of clustered prototypes with the proposed unsupervised fitness function based on MST (GA (clust) span), GA with the proposed unsupervised fitness function based on MST without clustering the prototypes (GA span), GA in the space of clustered prototypes with the proposed supervised fitness function (GA (clust) sup), and GA with the proposed supervised fitness function (GA sup).

The parameters used for the GA are: 20 individuals for the initial population, 0.5 for reproduction probability per gene, 0.02 for mutation probability per gene, the stopping condition is 20 generations reached for the GA with initial clustering in the space of prototypes, and 25 for the GA without the clustering.

4.2 Results and Discussion

From test set errors, for an increasing number of GA generations, we observed that 20 generations provide a good compromise for acceptable classification error at acceptable runtime. We used this number in all the experiments. Figure 1 presents the average errors over 30 experiments for 10, 15, 20, 30, 40 and 50 prototypes. It can be seen in Fig. 1(a) that, for the Zongker dataset, the best results are obtained with the FS and the supervised criterion, also the proposed



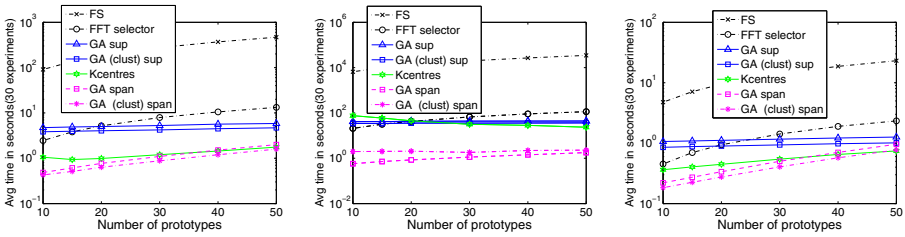
(a) Zongker LDC (b) pendigits 1-NN

(c) Diabetes LDC (d) XM2VTS LDC

Fig. 1. Average errors for different numbers of prototypes for the best performing classifier in each dataset

GA with this criterion outperforms the other unsupervised methods in accuracy with a comparable efficiency. The FS outperformed the GA in this dataset because it does not present a significant class overlap. However, when there is a significant overlap among the classes as in XM2VTS (see Fig. 1(d)), the GA outperforms the FS since it takes the relation among all the prototypes into account. One problem of the FS that causes its lower performance is that when an object is selected to be a prototype, it cannot be discarded afterwards.

From results on Diabetes dataset in Fig. 1(c), it can be seen that the proposed GA with the unsupervised criterion outperforms the other methods except for the FFT which has a similar performance but at a higher computational cost. From results for Pendigits in Fig. 1(b) we find again that this method is among the best performing ones both in speed and accuracy (see Fig. 2(b)); in addition, the computation of times for some of the datasets reported in Fig. 2 shows that the unsupervised GA is the fastest method. The supervised proposal is comparable to other unsupervised methods in execution times. This analysis, together with the computational complexity analysis, indicates that the presented GAs are able to scale well to large datasets when the final goal is the selection of



(a) Zongker (b) Pendigits (c) Diabetes

Fig. 2. Average times in seconds plotted in log scale

prototypes. However, the fitness function must also be designed scalable. As an extra bonus, GAs are embarrassingly parallelizable. Regarding our second question whether cluster analysis may be helpful, we see that results with initial clustering are usually equal to or better than those without clustering the prototypes before executing the GA, except for the XM2VTS complicated dataset that presents a significant class overlap.

In the XM2VTS we find that the best method is the proposed supervised GA but without the clustering. A clear explanation of why this happens was derived from the data exploration by multidimensional scaling (MDS) and the knowledge of the dataset characteristics. This is a dataset with strong illumination changes: frontal, right, and left illuminations of the faces. The three different illuminations create three large clusters of objects, so they are determined by noise instead of by the identities. Due to this, the unsupervised clustering before the GA does not find suitable clusters. However, our supervised criterion handles the class overlap well. Thereby, the supervised clustering that we evaluate in our criterion is not linked to the initial unsupervised clustering of the space of prototypes. We analyzed relations between performance of methods and data distribution by inspecting the MDS plots. We found that the supervised method handles well datasets with homogeneous distributions or with class overlap. In contrast, the unsupervised criterion copes better with inhomogeneous distributions where we can find inside the same class densely populated regions as well as sparse ones. In addition, the MST-based unsupervised method handles well elongated classes as in the Diabetes dataset.

5 Conclusions

The selection of prototypes is a crucial step for classification in the dissimilarity space. In this paper we proposed two different prototype selection methods by a GA and two different supervised and unsupervised criteria. Our work focuses on achieving low computational costs by finding approximate but sufficiently good solutions by powerful search heuristics, and maintaining low asymptotic complexities in the fitness function. Experimental results showed the validity of

the proposals for selecting good prototypes. The runtime analysis showed that the proposed methods are able to scale well to large datasets. Other general approaches include parallelism, stochastic methods etc. The proposed unsupervised method is the fastest since the evaluation of its criterion does not depend on the size of the dataset but on the number of prototypes. Besides, the linear time supervised criterion is also very fast compared to other supervised ones, which are generally quadratic and thereby do not scale well, and is comparable to unsupervised methods.

References

1. Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition: Foundations and Applications (Machine Perception and Artificial Intelligence). World Scientific Publishing Co., Inc., River Edge (2005)
2. Pekalska, E., Duin, R.P.W., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recogn. 39(2), 189–208 (2006)
3. Bunke, H., Riesen, K.: Graph classification based on dissimilarity space embedding. 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. 996–1007. Springer, Heidelberg (2008)
4. Olivetti, E., Nguyen, T.B., Garyfallidis, E.: The approximation of the dissimilarity projection. In: Second Workshop on Pattern Recognition in NeuroImaging, PRNI 2012, pp. 85–88. IEEE Computer Society, Washington, DC (2012)
5. García-Pedrajas, N., Haro-García, A.: Scaling up data mining algorithms: review and taxonomy. Progress in Artificial Intelligence 1(1), 71–87 (2012)
6. Plasencia-Calaña, Y., García-Reyes, E., Orozco-Alzate, M., Duin, R.P.W.: Prototype selection for dissimilarity representation by a genetic algorithm. In: Proceedings of the 20th International Conference on Pattern Recognition, ICPR 2010, pp. 177–180. IEEE Computer Society, Washington, DC (2010)
7. Lozano, M., Sotoca, J.M., Sánchez, J.S., Pla, F., Pekalska, E., Duin, R.P.W.: Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces. Pattern Recogn. 39(10), 1827–1838 (2006)
8. Zare Borzeshi, E., Piccardi, M., Riesen, K., Bunke, H.: Discriminative prototype selection methods for graph embedding. Pattern Recogn. 46(6), 1648–1657 (2013)
9. Jain, A.K., Zongker, D.: Representation and recognition of handwritten digits using deformable templates. IEEE Trans. Pattern Anal. Mach. Intell. 19, 1386–1391 (1997)
10. Spillmann, B., Neuhaus, M., Bunke, H., Pękalska, E., Duin, R.P.W.: Transforming strings to vector spaces using prototype selection. In: Yeung, D.-Y., Kwok, J.T., Fred, A., Roli, F., de Ridder, D. (eds.) SSPR 2006 and SPR 2006. LNCS, vol. 4109, pp. 287–296. Springer, Heidelberg (2006)
11. Alimoglu, F., Alpaydin, E.: Methods of combining multiple classifiers based on different representations for pen-based handwritten digit recognition. In: Fifth Turkish Artificial Intelligence and Artificial Neural Networks Symposium (1996)
12. Messer, K., Matas, J., Kittler, J., Jonsson, K.: XM2VTSDB: The extended M2VTS database. In: Second International Conference on Audio and Video-based Biometric Person Authentication, pp. 72–77 (1999)
13. Frank, A., Asuncion, A.: UCI machine learning repository (2010)

IOWA Operators and Its Application to Image Retrieval*

Esther de Ves, Pedro Zuccarello, Teresa Leon, and Guillermo Ayala

Universitat de València, Spain
Instituto de Microelectronica de Barcelona, IMB-CNM (CSIC), Spain

Abstract. This paper presents a relevance feedback procedure based on logistic regression analysis. Since, the dimension of the feature vector associated to each image is typically larger than the number of evaluated images by the user, different logistic regression models have to be fitted separately. Each fitted model provides us with a relevance probability and a confidence interval for that probability. In order to aggregate these set of probabilities and confidence intervals we use an IOWA operator. The results will show the success of our algorithm and that OWA operators are an efficient and natural way of dealing with this kind of fusion problems.

Keywords: Content-based image retrieval, logistic regression, IOWA.

1 Introduction

An aggregation operator is designed to reduce a set of objects into a unique representative one. The average or the weighted average are prototypes of numeric aggregation operators. Ordered Weighted Averaging (OWA) operators generalize the idea of the weighted average and in an additive form include the minimum and the maximum as particular cases. An important feature of these operators is the reordering step: the arguments are ordered by their value. Then a vector of weights is selected in order to model some aggregation imperative. Since their introduction in 1988 [8] OWA operators have been successfully used in a wide range of applications Yager and Filev introduced the class of Induced OWA (IOWA) operators in which the ordering of the arguments is induced by another variable called the inducing order variable. IOWA operators allow us to aggregate not only numerical quantities but also objects as intervals.

In a previous work we have made use of OWA operators [7] to fuse a collection of relevance probabilities into a single one in a Content Based Image Retrieval (CBIR) system. CBIR systems are one of the most promising techniques for retrieving multimedia information [6]. Visual features related to color, shape and texture are extracted in order to describe the image content. A general classification can be made: low level features (color, texture and shape) and high

* This work has been supported by project MCYT TEC2009-12980 from Spanish government.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 353–363, 2014.

level features (usually obtained by combining low level features in a reasonable predefined model). Since high level features have a strong dependency with the application domain, many research activities have been focused on the extraction of good low level descriptors [3] [2]. A query can be seen as an expression of an information need to be satisfied. Any CBIR system aims at finding images relevant to a query. The relationship between any image in the database and a particular query can be expressed by a relevance value. This relevance value relies on the user perceived satisfaction and can be interpreted as a relevance probability. In this work a relevance probability $\pi(\boldsymbol{x})$ is a quantity which reflects the estimate of the relevance of the image with low level feature vector $\boldsymbol{x}$ with respect to the user's information needs. Initially, every image in the database is equally likely, but as more information of the user's preferences is available, the probability concentrates on a subset of the database. The iterative algorithms which, in order to improve the result set from a query, require that the user enters his preferences in each iteration are called relevance feedback algorithms [9]. In [7] we presented a first version of the procedure (based on logistic regression analysis) that we improve in this work. In our algorithm, the image database is ranked by the output of the logistic regression model and shown to the user, who selects a few positive and negative samples, repeating the process in an iterative way until he/she is satisfied. The problem of the small sample size with respect to the number of features is solved by adjusting several partial generalized linear models and combining their relevance probabilities (and their confidence interval for those probabilities). Thus, we face to the question of how to combine them in order to rank the database. We have seen this question as an information fusion problem which is takled by using OWA and IOWA operators.

The major improvements introduced in this paper compared to [7] are the use of confidence intervals as the estimation for relevance probabilities, and the aggregation of them using an IOWA operator. Another novelty comes from the use of bootstrap sampling to compensate the size of the sample set that feeds the different logistic regression models.

Section 2 is a brief recall of the notation corresponding to OWA and IOWA operators. Section 3 summarize how logistic regression is applied to our CBIR system. In section 4 IOWA operator is explained. The bootstrap selection of sample images and the ranking procedure is detailed in section 5. In section 6 and 7 we present the experimental results and we extract some conclusions.

2 Notation and Preliminary Results

An OWA operator of dimension m is a mapping $f : \mathbb{R}^m \to \mathbb{R}$ with an associated weighting vector $W = (w_1, \ldots, w_m)$ such that $\sum_{j=1}^{m} w_j = 1$ and where $f(a_1, \ldots, a_m) = \sum_{j=1}^{m} w_j b_j$ being b_j the j-th largest element of the collection of aggregated objects $a_1, \ldots, a_m$. For $W = (1, 0, \ldots, 0)$ we obtain $f(a_1, \ldots, a_m) = \max_i a_i$, for $W = (0, 0, \ldots, 1)$, $f(a_1, \ldots, a_m) = \min_i a_i$ and for $W = (\frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m})$, we have that $f(a_1, \ldots, a_m) = \frac{1}{m} \sum_{j=i}^{m} a_i$.

As OWA operators are bounded by the Max and Min operators, Yager introduced a measure call *orness* to characterize the degree to which the aggregation is like an *or* (Max) operation: orness$(W) = \frac{1}{m-1}\sum_{i=1}^{m}(m-i)w_i$

This author also introduced the concept of *dispersion* or *entropy* associated with the weighting vector $Disp(W) = \sum_{i=1}^{m} w_i \ln w_i$. It reflects how much of the information in the arguments is used during an aggregation based on W.

Yager extended the OWA operator to the case where the arguments to be aggregated are in an interval $[a, b]$. Let $Q : [0, 1] \to [0, 1]$ be a function having the properties of $Q(0) = 0$, $Q(1) = 1$ and $Q(x) \geq Q(y)$ if $x > y$, then Q is a basic unit-interval monotonic (BUM) function. This function provides the weights of the values in the interval. And the corresponding continuous interval argument OWA (COWA) operator F_Q is defined as :

$$F_Q([a, b]) = \int_0^1 \frac{dQ(y)}{dy}(b - y(b - a))dy = a + (b - a)\int_0^1 Q(y)dy. \qquad (1)$$

If we denote $\lambda = \int_0^1 Q(y)dy$ we have that $F_Q([a, b]) = (1 - \lambda)a + \lambda b$, and λ is called the attitudinal character of the BUM function Q and its interpretation is similar to the orness.

An Induced Ordered Weighted Average (IOWA) operator of dimension m is a function $\Phi_W : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}$ with an associated weighting vector $W = (w_1, \ldots, w_m)$ such that $\sum_{j=1}^{m} w_j = 1$, and it is defined to aggregate the set of second arguments of a list of m 2-tuples $\{(v_1, d_1), \ldots, (v_m, d_m)\}$ according to the expression $\Phi_W((v_1, d_1), \ldots, (v_m, d_m)) = \sum_{i=1}^{m} w_i d_{\sigma(i)}$ where σ is a permutation of $\{1, \ldots, m\}$ such that $v_{\sigma(i)} \geq v_{\sigma(i+1)}, \forall i \in 1, \ldots, m-1$, i.e. $(v_{\sigma(i)}, d_{\sigma(i)})$ is the 2-tuple with $v_{\sigma(i)}$ the i-th highest value in the set $\{v_1, \ldots, v_m\}$. Yager and Filev [4] call the set of values $\{v_i\}_{i=1}^{m}$ the values of an inducing order variable and $\{d_i\}_{i=1}^{m}$ the values of the argument variable.

3 A Relevance Feedback Mechanism

Previously to the application of the CBIR algorithm each image has been described by using low level features and the j-th image is identified to a k-dimensional feature vector $\boldsymbol{x}_j$. Our system can work currently with different low level features such as color and texture.

At every iteration of the procedure, the user inspects a (non-random) sample of images from the database and makes a number of positive and negative selections We would like to point out that the first screen in our method shows a set of representatives of the database obtained through a Partition Around Medoids (PAM) method [5]. In this way, the user is able to inspect a variety of images and hopefully some images are similar to his/her query. The information provided by the user is captured in a binary variable Y where $Y = 1$ or $Y = 0$ indicates that a given image in the sample is classified as an example or counter-example respectively. We have to model the distribution of Y with the low level features associated to the image.

Generalized linear models (GLMs) extend ordinary regression models to encompass non-normal response distributions and modeling functions of the mean. In particular, logistic regression models are the most important for categorical response data. For a binary response variable Y and t explanatory variables $X_1, \ldots, X_t$, the model for $\pi(x) = P(Y = 1 \mid x)$ at values $x = (x_1, \ldots, x_t)$ of predictors is $logit[\pi(x)] = \alpha + \beta_1 x_1 + \ldots + \beta_t x_t$, where $logit[\pi(x)] = \ln \frac{\pi(x)}{1-\pi(x)}$.

A practical question arises at this point: the user evaluates a small number of images, let us say n_r, at each iteration r of the whole search process, and n_r is very small compared to the dimension, k, of the vector of characteristics $\boldsymbol{x}$. Our approach consists in partitioning the set of characteristics into C homogeneous subsets, i.e. we consider $\boldsymbol{x} = (x^{(1)}, \ldots, x^{(C)})$ i.e. $\boldsymbol{x} \in \mathbb{R}^k$ where $k = \sum_{c=1}^{C} k_c$. Each $x^{(i)}$ corresponds to a set of semantically related characteristics (for instance color).

Then, we consider separately each subvector $x^{(j)}, j = 1, \ldots, C$ and fit a regression model. For simplicity, let us denote by $\boldsymbol{u}$ one of these subvectors and by t its dimension. We will estimate the probability that $Y = 1$ given the feature vector $\boldsymbol{u}$ i.e. $P(Y = 1|\boldsymbol{u}) = \pi(\boldsymbol{u})$. In order to estimate this probability we take into account that our data in a given iteration, will be the sample $(\boldsymbol{u}_i, y_i)$ with $i = 1, \ldots, n$ where $\boldsymbol{u}_i$ and y_i are the feature vector and the user preference for the i-th image evaluated in the iteration. We fit a logistic regression model assuming that the data $(\boldsymbol{u}_i, y_i)$ with $i = 1, \ldots, n$ are independent. The random variable Y, giving the random preference for an image with feature vector $\boldsymbol{u}$, has a Bernoulli distribution (where 1 means success or that image is considered relevant) where the probability of one is $\pi(\boldsymbol{u})$. In summary, Y conditioned to $\boldsymbol{u}$ is distributed as $Y \sim Bi(1, \pi(\boldsymbol{u}))$. Furthermore, the different Y_i's (given the feature vectors u_i's) are conditionally independent. Furthermore the logistic regression model will assume that

$$\pi(\boldsymbol{u}) = \frac{\exp(\beta_0 + \beta_1 u_1 + \ldots + \beta_t u_t)}{1 + \exp(\beta_0 + \beta_1 u_1 + \ldots + \beta_t u_t)}. \tag{2}$$

We can compute the maximun likelihood estimator of $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_t)'$, $\hat{\boldsymbol{\beta}}$, by using the Fisher Scoring method as usual (see [1], pp. 145-149). The probability $\pi(\boldsymbol{u})$ is estimated by replacing in the equation 2, the parameters $\boldsymbol{\beta}$ by the corresponding MLE $\hat{\boldsymbol{\beta}}$. The corresponding confidence interval for this probability can be calculated by using the distribution of $\hat{\boldsymbol{\beta}}$. Let us denote the corresponding confidence interval as $I(\{(\boldsymbol{u}_1, y_1), \ldots, (\boldsymbol{u}_n, y_n)\})$.

For each image the output of the logistic regression module provides a set of C confidence intervals for C relevance probabilities, they have been obtained by considering C different sets of low level characteristics of the image. Clearly, we need to use an aggregation operator to combine them.

4 An IOWA Operator

We need to aggregate the different confidence intervals associated to the different subvectors. Let us denote by $I_1(\boldsymbol{x})$, $I_2(\boldsymbol{x}), \ldots, I_C(\boldsymbol{x})$ the confidence intervals for

the probabilities of relevance associated to a given image with low level features $\boldsymbol{x}$. The indices $1, \ldots, C$ correspond to different subsets of characteristics of the image.

For any interval $I_i(\boldsymbol{x}) = (\pi_i - l_i, \pi_i + l_i)$ we define an inducing order variable that takes into account both the interval amplitude $2l_i$ and its midpoint π_i according to: $F_Q(\pi_i - l_i, \pi_i + l_i) = \pi_i - \frac{1}{3}l_i$ for $i = 1, \ldots, C$. Notice that, in this way, we aggregate the continuous intervals making use of a COWA operator with BUM function $Q(y) = y^2, y \in [0, 1]$ with an attitudinal character $\lambda = \frac{1}{3}$.

Let us denote by v_i the aggregated value of $I_i(\boldsymbol{x})$ for $i = 1, \ldots, C$, then $\{v_i\}_{i=1}^{C}$ are the values of the variable, which induces the order within the intervals. Let us denote by $I_{(1)}(\boldsymbol{x}), \ldots, I_{(C)}(\boldsymbol{x})$ the ordered intervals, then we use a vector of weights $w = (w_i, \ldots, w_C)$ to aggregate them. The final interval will be $\sum_{j=1}^{C} w_j I_{(j)}(\boldsymbol{x})$. Next, we will explain our proposal for the weighting vector.

A key issue in defining an OWA operator is the choice of the vector of weights. Several approaches have been presented including learning from data, exponential smoothing or aggregating by quantifiers. Our proposal (already detailed in [7]) is to construct a parametric family of weights as a mixture of a binomial and a discrete uniform distributions. One of the advantages of the use of this family of weights is that the binomial distribution allows us to concentrate the higher values of the weights around $\mu = (m - 1)\alpha$, while the discrete uniform component of the mixture allows us to keep the weights away from μ high enough so the information from all relevance probabilities is taken into account in the aggregation process.

5 Ranking the Database

This section explains how we combined historical information from all iterations together with all the theoretical concepts explained in previous sections to obtain one single relevance probability value that would allow us to rank the images in the database.

5.1 Learning from Previous Iterations

Let us denote by n_r^+ and n_r^- the number of images marked as relevant and non-relevant at the r-th iteration respectively. The experience shows that n_r^- is usually much greater than n_r^+, thus the sample is clearly unbalanced.

In order to correct the bias, we use a randomization procedure that takes into account the positive and negative selections of the r-th iteration as well as the ones selected by the user in all iterations previous to the r-th. Let us denote by $\mathbf{N}_r^+$ and $\mathbf{N}_r^-$ the sets of relevant and non-relevant images stored along the iterations $1, \ldots, r$ respectively. The logistic regression model described in section 3 is fed with a set of n^+ and n^- randomly selected images without replacement from $\mathbf{N}_r^+$ and $\mathbf{N}_r^-$ respectively. In our procedure the sample sizes n^+ and n^- remain constant through all iterations. A reasonable condition would be that an image selected (as positive or negative) in a particular iteration, q,

has more probability to be included amongst the n^+ and n^- sample images than one selected in iteration $q-1$. This way, the probability of a positive image to be included is: $P(x_q^+) = \frac{2^q}{\sum_{t=1}^{r} 2^t n_t^+}$, where q is the iteration where image x_q^+ was positively evaluated, r is the present iteration, and n_t^+ is the number of relevant images in iteration t. For a non-relevant image, the probability is: $P(x_q^-) = \frac{q}{\sum_{t=1}^{r} t n_t^-}$, where q is the iteration where image x_q^- was negatively evaluated, and n_t^- is the number of non-relevant images in iteration t.

$P(x_q^-)$ increases linearly with respect to the iterations, while $P(x_q^+)$ increases exponentially. This difference is because if the user considers an image as non-relevant he will not change his mind at any point of the query. On the other hand, as the search progresses, it can be assumed that the user is much more interested in the most recently relevant selections than in the previous ones. This is the reason why a much more abrupt memory function (like the exponential) is used.

5.2 The Algorithm

Summarizing, in every iteration the ranking procedure works as follows:

- We have a database composed of N images, each one with feature vector $\boldsymbol{x}_j$, with $j = 1 \ldots N$. Each feature vector is splitted in C subvectors $x_j^{(c)}$, with $c = 1 \ldots C$.
- For every $x_j^{(c)}$, S intervals for the estimation of the relevance probability are computed: $I_{j,s}^c$, with $s = 1 \ldots S$. The intervals $I_{j,s}^{(c)}$ are the outcome of S logistic regression models. The input of these models are random sets composed of n^+ and n^- sample images.
- The S intervals $I_{j,s}^{(c)}$ are averaged thus obtaining one single interval for every subvector and image: $\tilde{I}_j^{(c)} =< I_{j,s}^{(c)} >_s$, where $<>_s$ means arithmetical averaging over the s variable.
- The C intervals corresponding to image j, $j = 1 \ldots N$, are aggregated by means of an IOWA to obtain a single and only interval for every image $\tilde{I}_j$.
- COWA operator with attitudinal character $\lambda = 1/3$ is used to select one single value from the $\tilde{I}_j$ intervals to represent them. These N values are used to re-rank the database.

6 Experimental Results

6.1 Experimental Setup

A database with a total number of about 4700 images has been used for the experiments. The semantic content of the images contained in this experimental database covers a wide variety of themes such as flowers, horses, paintings, landscapes, trees, etc.

In order to evaluate our proposed search procedure, a test was repeated for several distinct users and images. The objective of the test was to find a certain

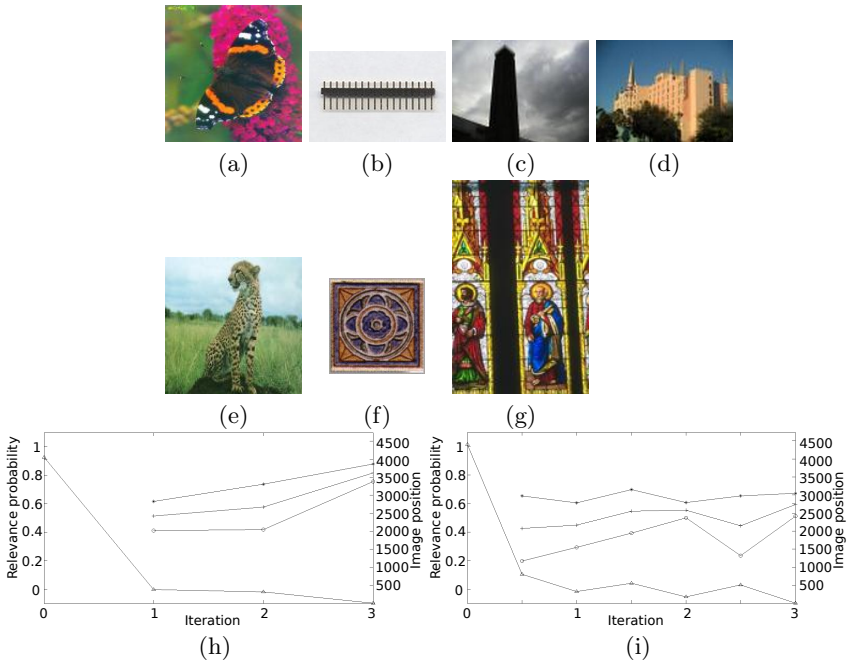


Fig. 1. Target images used in experiments. Figure also shows in (h) and (i) the evolution of (+) relevance probability, (*) superior and (o) inferior limits of confidence interval, and (△) position through iterations for the target image in a particular search.

image (target image) amongst the 4700 images of the database. The initial ranking of the target image was always higher than 2500. The developped system permits to see 32 thumbnailed images at a time. Initially the images are randomly ordered, except for the first 64 images (two first pages of the interface). These first 64 images are the medoids obtained with a PAM method [5] applied to the complete database. This new strategy would allow the user to find what he/she may consider as relevant images more rapidly just by inspecting only a few of the first pages of the database. In further iterations the images are ranked according to the procedure explained in section 5.1. Several parameters need to be adjusted for the algorithm to work properly. We set the values of n^+, n^- and S (see section 5.1) to 4, 6 and 8 respectively. The number of images selected by the user in every iteration could not be inferior to 6 considering the sum of relevant plus non-relevant, and could not be zero for any of them individually. The search is considered successful if the target appears in the ranking list in a position between 1 and 32. If the number of iterations required to complete the search is superior to 20 the system considers that the search failed. Different users were asked to do the search for different images, while a total number of seven different images corresponding to different themes were used (see figure 1).

The feature vector, with dimension $k = 50$, was splitted into $C = 10$ subvectors with 5 components each. We have avoided to mix features of different nature in the same subvector. A subvector could not contain information from color and texture at the same time.

6.2 Descriptives

A total number of 40 queries were analyzed. Table 1 shows several descriptive values of the experiment. The iterations means are calculated considering all images and users. It must be noticed that the mean value of this variable is 3.07, with quartiles 1 to 3 ranging from 2 to 3.25 iterations. These values show that the number of iterations needed to successfully complete a query is quite small. From this point of view the proposed procedure can be considered very effective.

The number of relevant and non-relevant selections (n^+ and n^- respectively) are analyzed considering all iterations, users and images. Table 1 clearly shows that the number of positive selections is usually much more smaller than negative ones, therefore, our criteria for the random selection of images balancing the sample sizes (see section 5.1) is justified.

Table 1. Descriptive values (quartiles, mean and standard deviation) of the distribution of the number of iterations, n^+ and n^-

	Iterations	Pos	Neg
Mean	3.0732	5.2143	14.8333
Std	2.5039	3.4632	17.3826
Q_1	2	3	4
Q_2	2	5	7
Q_3	3.25	7	17

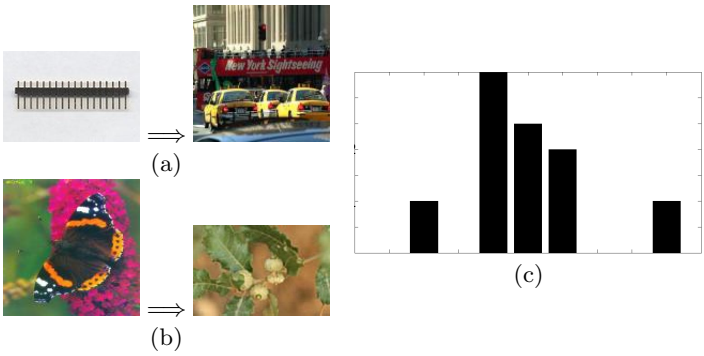


Fig. 2. Pairs of images used for tests with two sequential images (a), (b). First images on the left and second images on the right. The histogram of iterations for double sequential queries is also shown in (c).

Figure 1(h)(i) exhibits several typical learning curves for different particular searches. The target ranking improves (approximates to the first page) through iterations, while relevance probability increases and confidence interval value reduces. This curves also depend on the user abilities and experience. This can be seen on figure 1 (i) where, although the search ends in 6 iterations, the learning curves are not as straight as in figure 1(h). Figures 3(a) and (b) show the histograms of the amplitudes of the confidence intervals for a subset of texture and color information respectively, for image in figure 1(b) where all users and iterations are considered. Most of the information in these histograms is concentrated around extreme values: smaller than 0.01 meaning that the relevance probabilities are very precise and reliable information can be extracted from them, or between 0.49 and 0.5 pointing out the complete absence of knowledge about the relevance probability values.

Another interesting experiment was performed in order to to evaluate if our procedure is able to cope with a change in the user's mind in the middle of a search. For some images the users were asked to modify the search. Two target images were sequentially shown, and after the first target image was found, the users were asked to find the second one but without changing any of the parameters of the system: the memory for positive and negative selections as well as the database ranking remained without change from the first to the second target. Figure 2 shows the pairs of images used for the tests.

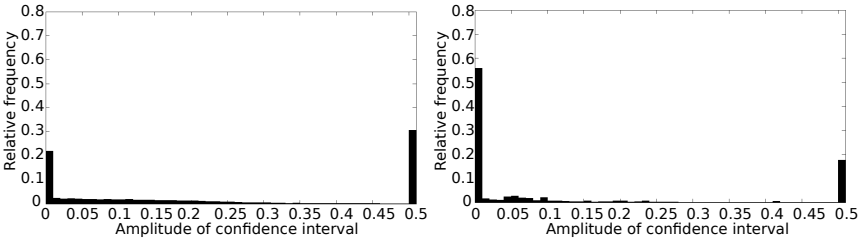


Fig. 3. Histograms of the amplitude of confidence intervals for a subset of (a) texture information and (b) color information for the queries of image 1(b)

A total number of 10 tests were carried out. Figure 2(c) shows the histogram of the number of iterations needed to complete the search of both images sequentially. Examples of target ranking and relevance probability evolution are depicted in figure 4(a) and 4(b) for images 2(a) and 2(b) respectively. The results are shown to be very successful. In all cases the users were able to find both target images in sequential order. As a conclusion we can say that our system shows very satisfactory recovering capabilities.

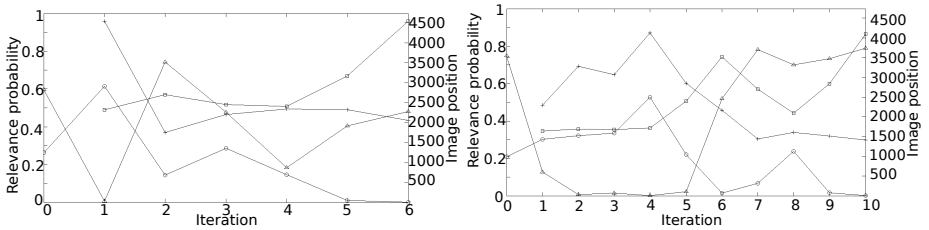


Fig. 4. Figure shows in (a) and (b) the evolution of (△) first and (o) second target ranking and (+) first and (□) second relevance probability for double sequential searches

7 Conclusions and Further Developments

We have presented in this paper a content based image retrieval system that uses as fundamental tools logistic regression, fuzzy aggregation operators and bootstrap techniques. In fact, it could be considered as a very general and flexible framework. We have shown that it can be adapted easily to a particular problem (to find a target image) providing successful results. A major difficulty in this context is concerned with the ratio between the dimension of the feature vector associated to each image and the number of images evaluated by the user. This drawback was approached in [7] by fitting several partial generalized linear models and combining their relevance probabilities by means of an OWA operator. We have improved our previous work by describing the relevance probabilities through their confidence intervals and aggregating them by means of an IOWA operator. Although other improvements have been added, in our opinion the correct aggregation of a more complete and complex information has become the major strength of the algorithm.

References

1. Agresti, A.: Categorical Data Analysis, 2nd edn. Wiley (2002)
2. de Ves, E., Benavent, X., Ruedin, A., Acevedo, D., Seijas, L.: Wavelet-based texture retrieval modeling the magnitudes of wavelet detail coefficients with a generalized Gamma distribution. In: IEEE International Conference on Pattern Recognition (ICPR), pp. 221–224 (2010)
3. Minh, N.: Do and M. Vetterli. Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance. IEEE Transactions on Image Processing 11(2), 146–158 (2002)
4. Filev, D., Yager, R.: On the issue of obtaining owa operator weights. Fuzzy Sets and Systems 94, 157–169 (1998)
5. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data. An Introduction to Cluster Analysis. Wiley (1990)
6. Smeulders, A.W.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 Intellingence 22(12), 1349–1379 (2000)

7. León, T., Zuccarello, P., Ayala, G., de Ves, E., Domingo, J.: Applying logistic regression to relevance feedback in image retrieval systems. Pattern Recognition 40, 2621–2632 (2007)
8. Yager, R.R.: On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Trans. Systems Man Cybernet. 18, 183–190 (1988)
9. Zhou, X.S., Huang, T.S.: Relevance feedback for image retrieval: a comprehensive review. Multimedia Systems 8(6), 536–544 (2003)

On Optimum Thresholding of Multivariate Change Detectors

William J. Faithfull and Ludmila I. Kuncheva

School of Computer Science, Bangor University, Bangor,
Gwynedd, Wales, United Kingdom
{w.faithfull,l.i.kuncheva}@bangor.ac.uk
pages.bangor.ac.uk/~eese11

Abstract. A change detection algorithm for multi-dimensional data reduces the input space to a single statistic and compares it with a threshold to signal change. This study investigates the performance of two methods for estimating such a threshold: bootstrapping and control charts. The methods are tested on a challenging dataset of emotional facial expressions, recorded in real-time using Kinect for Windows. Our results favoured the control chart threshold and suggested a possible benefit from using multiple detectors.

1 Introduction

Detecting a change point in a sequence of observations is a well researched statistical problem with applications in areas such as Economics [1], Data Stream Mining [2] and Quality Control [3]. The basic premise of change point detection is that, given a sequence of observations $x_1, x_2, ..., x_n$, there exists a change point t such that $x_1, x_2, ..., x_t$ was generated exclusively by some process P_0 and $x_t, x_{t+1}, ..., x_n$ was generated exclusively by some other process P_1.

Detecting change points in multivariate data is a challenging problem. A variety of multivariate change detectors have been proposed [4–6], some of which amount to a novel combination of univariate detectors, while others take a dimensionality reduction approach. In the latter case, the multidimensional data is reduced to a single statistic which should ideally correlate with the appearance of change. One of the main issues with such detectors is identifying a threshold on the single statistic for flagging a change. Here we examine the suitability of two approaches to setting a threshold: bootstrapping and control charts. Figure 1 illustrates the multivariate change detection process.

2 Related Work

Change detection has been an active area of research for more than 60 years, developing out of methods for statistical quality control. Being well researched and statistically grounded, Control Charts are the basis for many methods such as

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 364–373, 2014.

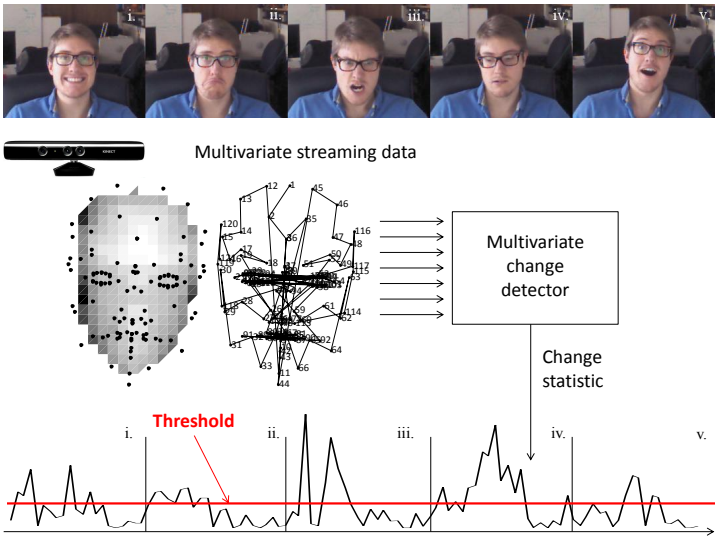


Fig. 1. Illustration of the process of change detection in streaming multidimensional data and the role of the threshold. The data was obtained from Kinect while a participant was acting a sequence of emotional states: *i*. Happiness, *ii*. Sadness, *iii*. Anger, *iv*. Indifference, *v*. Surprise.

CUSUM (Cumulative Sum) charts and EWMA (Exponentially Weighted Moving Average) charts. Some of the earliest work in the field is that of Shewhart [3, 7] and his development of the control chart for sequential process control, now widely adopted by industry. The field is now very broad, with a number of reference monographs including Wald [8], Basseville and Nikiforov [9] and Brodsky and Darkhovsky [10] although largely focussed on univariate data.

There are differing approaches to the problem of detecting change in multivariate data. Lowry and Montgomery [11] reviewed multivariate control charts for quality control. Consider n p-dimensional vectors of observations $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$. It is possible to simply create p individual charts, one for each feature, not reducing the dimensionality of the data. However, this approach does not account for correlation between the features. Even truly multivariate control chart approaches such as the Hotelling Control Chart [12] can be equated to dimensionality reduction and thresholding, as it reduces the p dimensions of the data to a single T^2 statistic. The list below demonstrates the inconsistency of approaches to setting such a threshold.

Work:	Decision method
Zamba & Hawkins [13]:	γ set according to a desired false alarm rate.
Song et al. [14]:	Original statistical test.
Dasu et al. [5]:	Monte Carlo Bootstrapping.
Kuncheva [15]:	Signficance of log-likelihood ratio.

The scope of this work is concerned with establishing a method for threshold setting that is applicable to multiple approaches to change detection.

3 Multivariate Change Detectors

Here we assume that the change detection criteria are calculated from pre-specified windows of data W_1 and W_2. Change is sought between the distributions in the two windows.

3.1 Parametric Detectors: Hotelling

The two windows of data contain points $\mathbf{x} = [x_1, \ldots, x_p]^T \in \Re^p$. Hotelling [16] proposes a statistical test for equivalence of the means of the two distributions from which W_1 and W_2 are sampled. The null hypothesis is that W_1 and W_2 are drawn independently from two multivariate normal distributions with the same mean and covariance matrices. Denote the sample means by $\hat{\mu}_1$ and $\hat{\mu}_2$, the pooled sample covariance matrix by $\hat{\Sigma}$, and the cardinalities of the two windows by $M_1 = |W_1|$ and $M_2 = |W_2|$. The T^2 statistic is calculated as

$$T^2 = \frac{M_1 M_2 (M_1 + M_2 - p - 1)}{p(M_1 + M_2 - 2)(M_1 + M_2)} \times (\hat{\mu}_1 - \hat{\mu}_2)^T \hat{\Sigma}^{-1} (\hat{\mu}_1 - \hat{\mu}_2) \tag{1}$$

Under the null hypothesis, T^2 has F distribution with degrees of freedom p and $M_1 + M_2 - p + 1$. The T^2 statistic is the Mahalanobis distance between the two sample means multiplied by a constant. The p-value of the statistical test is instantly available and the desired significance level will determine the change threshold.

The obvious problem with the Hotelling test is that it is only meant to detect changes in the position of the means. Thus it will not be able to indicate change of variance or a linear transformation of the data that does not affect the mean.

3.2 Semi-parametric Detectors: SPLL

The semi-parametric log-likelihood criterion (SPLL) for change detection [6] comes as a special case of a log-likelihood framework, and is modified to ensure computational simplicity. Suppose that the data before the change comes from a Gaussian mixture $p_1(\mathbf{x})$ with c components each with the same covariance matrix. The parameters of the mixture are estimated from the first window of data W_1. The change detection criterion is derived using an upper bound of the log-likelihood of the data in the second window, W_2. The criterion is calculated as

$$SPLL = \max\{SPLL(W_1, W_2), SPLL(W_2, W_1)\}. \tag{2}$$

where

$$SPLL(W_1, W_2) = \frac{1}{M_2} \sum_{\mathbf{x} \in W_2} (\mathbf{x} - \mu_{i*})^T \Sigma^{-1} (\mathbf{x} - \mu_{i*}). \tag{3}$$

where M_2 is the number of objects in W_2, and

$$i* = \arg\min_{i=1}^{c} \left\{ (\mathbf{x} - \mu_i)^T \Sigma^{-1} (\mathbf{x} - \mu_i) \right\} \tag{4}$$

is the index of the component with the smallest squared Mahalanobis distance between $\mathbf{x}$ and its centre.

If the assumptions for p_1 are met, and if W_2 comes from p_1, the squared Mahalanobis distances have a chi-square distribution with p degrees of freedom. The expected value is p and the standard deviation is $\sqrt{2p}$. If W_2 does not come from the same distribution, then the mean of the distances will deviate from p. Subsequently, we swap the two windows and calculate the criterion again, this time $SPLL(W_2, W_1)$. By taking the maximum of the two, SPLL becomes a monotonic statistic.

3.3 Non-parametric Detectors: Kullback-Leibler Distance

In this approach, the data distribution in window W_1 is represented as a collection of K bins (regions in $\Re^p$), with a probability mass value assigned to each bin. Call this empirical distribution $\hat{P}_1$. The data in W_2 is distributed in the bins according to the points' locations, giving empirical distribution $\hat{P}_2$. The criterion function is

$$KL(\hat{P}_2 || \hat{P}_1) = \sum_{i=1}^{K} \hat{P}_2(i) \log \left\{ \frac{\hat{P}_2(i)}{\hat{P}_1(i)} \right\} \tag{5}$$

where i is the bin number, and $\hat{P}(i)$ is the estimated probability in bin i.

If the two distributions are identical, the value of $KL(P_2||P_1)$ is zero. The larger the value, the higher the likelihood that P_2 is different from P_1. Note that we have only approximations of P_1 and P_2. The usefulness of the KL criterion depends on the quality of the approximations and on finding a threshold λ such that change is declared if $KL > \lambda$.

In Dasu et al.'s change detector [5], W_1 is expanded until change is detected, giving a good basis for approximating P_1. On the other hand, P_2 has to be estimated from a short recent window, hence the estimate may be noisy. Dasu et al. approximate the P_1 probability mass function by building *kdq* trees which can be updated with the streaming data. Other approximations are also possible, including the clustering approach for SPLL.

The KL distance criterion is not related to a straightforward statistical test that will give us a fixed threshold λ, which was one of the motivations behind our study.

4 Threshold Setting Approaches

Hotelling T^2 detector has the advantage of a statistically interpretable threshold. However, it has a serious shortcoming in that it only detects change in the mean of the data. To equip SPLL and KL with a similar type of threshold, here we examine two threshold setting approaches for the change detection statistic.

4.1 Bootstrapping

Let $|W_1| = M_1$. To determine a threshold, a bootstrap sample of M_1 objects is drawn from W_1. A discrete probability distribution $\hat{P}_1$ is approximated from this sample. Subsequently, another sample of the same size is drawn from W_1 and its distribution $\hat{Q}_1$ is evaluated. For example, if $\hat{P}_1$ is a set of bins, $\hat{Q}_1$ is calculated as the proportion of the data from the second bootstrap sample in the respective bins. The match between $\hat{P}_1$ and $\hat{Q}_1$ is estimated using, for example, KL distance (5), which gives the change statistic. Running a large number of such Monte Carlo simulations, a distribution of the change statistic is estimated, corresponding to the null hypothesis that there is no change (all samples were drawn from the same window, W_1). We can take the Kth percentile of this distribution as the desired threshold. This approach was adopted by Dasu et al. [5] where the probability mass functions were approximated by a novel combination of kd-trees and quad trees, called kdq-trees. We direct the reader to [5] for an in-depth definition of kdq-trees. One drawback of this approach is the excessive computation load when a new threshold is needed.

4.2 Control Chart

A less computationally demanding alternative to bootstrapping is a Shewhart individuals control chart to monitor the change statistic. Inspired by this, our hypothesis is that the process underlying an appropriate change statistic will exhibit an out-of-control state when change occurs. Using a window of T observations, we calculate the centre line $\bar{x}$ as the mean of the values of the statistic returned from the change detector, and its standard deviation $\hat{\sigma}$. The upper and lower control limits are calculated as

$$\bar{x} \pm 1.96 \frac{\hat{\sigma}}{\sqrt{T}}. \tag{6}$$

If either of the control limits are exceeded, change is signalled. This (rather naive) threshold estimation assumes that the change statistic has normal distribution, and that we have a sufficiently large window so as to get reliable estimates. The above value is for significance level $\alpha = 0.05$. The bootstrap threshold does not rely on any such assumption but is more cumbersome.

5 Experimental Investigation

All thresholds considered here, including the threshold of the Hotelling method, are meant to control the type I error ("convict the innocent", or accepting that there is a change when there is none). If we set all these thresholds to 0.05, we should expect to have false positive rate less than that. Nothing is guaranteed about the type II error ("free the guilty", or missing a change when there is one). Thus we are interested to find out how the three chosen change detectors behave for the two type of thresholds, in terms of both error types.

5.1 Facial Expression Data

We chose a challenging real-life problem to test the change detectors. Sustained facial expressions of five emotions were taken to be the stable states, and the transition from one emotion to another was the change.

While a number of facial expression databases exist, they require camera equipment and intermediate computer vision techniques to record data. In our approach, we utilise the Face Tracking toolkit distributed with the Kinect SDK to extract data directly from the device. This approach lends itself to analysis of real-time streaming data The advantage of having a minimal setup is that data capture does not have to be intrusive. This presents the opportunity of capturing real-time data about a participant's posture and facial expression whilst they interact with the computer.

The Kinect Face Tracking SDK utilises the Active Appearance Model (AAM) [17], taking into account the data from the depth sensor to allow head and face tracking in 3D. The features we take from the Kinect are as follows:

- Features extracted by the Kinect software
  - Face Points : 123 3D points on the face
  - Skeleton Points : 10 3D points on the joints of the upper body
  - Animation Units: 6 Animation Units $[-1, 1]$

- Six animation units and their equivalents in the Candide3 model

Animation Unit	Candide3 [18]	Description
AU0	AU10	Upper Lip Raiser
AU1	AU26/27	Jaw Lowerer
AU2	AU20	Lip Stretcher
AU3	AU4	Brow Lowerer
AU4	AU13/15	Lip Corner Depressor
AU5	AU2	Outer Brow Raiser

5.2 Data Capture

Each participant sat with their eyes trained on a computer screen, with a Kinect observing them. Emotional transitions are triggered by visual instructions. The participants were asked to hold their facial expression until instructed to change it. The duration of a facial expression is 3 seconds. The timestamps of these instructions are logged to provide the true positive values for the experiment. Thus each experimental run produces about 5 expressions $\times$ 3 seconds $\times$ 30 FPS = 540 frames. Figure 2 shows an example of one of the animation units throughout one run. The periods of sustained facial expressions are labelled. The initial warm-up period, as well as the transition periods of 7 frames are also indicated.

The process is facilitated by a bespoke application written in the C# language, which utilises the Kinect SDK to retrieve frames from the sensor and extract the features. The application acts as a TCP client which connects to a server

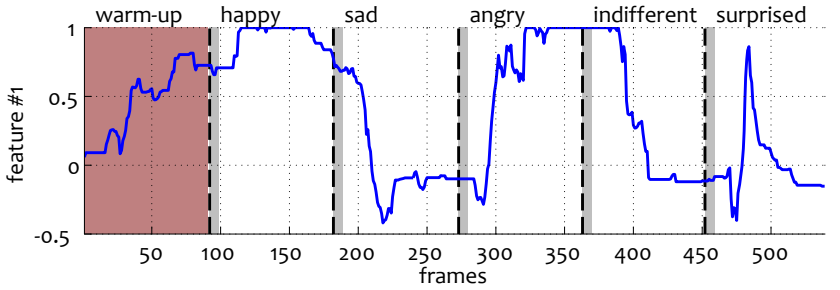


Fig. 2. An example of an animation unit along one experimental run for collecting data. The dashed vertical lines are the time points where the participant is prompted to change their facial expression. The shaded regions are transition stages.

running in MATLAB, where the extracted features and timestamps are streamed in real-time, ready for analysis.

5.3 Experimental Methodology

The experiment was conducted using the Animation Units from six participants, each of whom recorded ten runs using the apparatus. Human reaction time to visual stimuli is 180-200 ms. In a recording at approximately 30 frames per second, a true positive detection should appear no earlier than $180/30 = 6$ frames after the labelled change (prompt to change the facial expression). For each run, we test Hotelling, KL Distance with Bootstrapping, KL Distance with Control Charts, SPLL with Bootstrapping and SPLL with Control Charts. The protocol below was followed for each run and for each participant:

1. Split the data into segments by label.
2. Sample a window W_1 of T contiguous frames from a random segment S, with cardinality $|S| = M$ and random starting frame F, $7 \leq F \leq (M - T)$.
3. Sample W_2 from a random segment. If drawn from the same label as W_1, test for false positives, else test for true positives.
4. Calculate the threshold from W_1 using the chosen method.
5. Calculate change statistic from W_1 and W_2 and compare with the threshold. Store 'change' or 'no change', as well as the time taken to execute the iteration steps.
6. Repeat 1–5 K times sampling W_1 and W_2 from the same label, K times sampling W_1 and W_2 from different random labels. Calculate and return the true positive and false positive rates for the chosen detector and threshold.

Five hundred runs were carried out for determining the bootstrapping threshold.

To simulate a window of running change statistic only from data window W_1, we adopted the following procedure. A sliding split point m was generated, which was varied from 3 to $T - 3$. This point was used to create windows W_1', with

data from 1 to m, and W_1'', with data from $m+1$ to T. The statistic of interest was calculated from these sub windows, which were assumed to come from the same distribution.

We used $T = 50$, in order that the window size be above 50% of an expression duration. While there is a great deal of literature on the subject of adaptive windowing [19–21], this is beyond the scope of this paper. Such a technique could be used to set T. We set $K = 30$. The experiment was performed on a Core i7-3770K 4.6GHz Windows machine with 16GB RAM.

5.4 Results

We can examine the relative merit of the detectors and thresholds by plotting them on a Receiving Operating Characteristic (ROC) curve. The x-axis is '1− Specificity' of the test, which is the false positive rate, and the y-axis is the 'Sensitivity' of the test, which is the true positive rate. Each run for each participant can be plotted as a point in this space. An ideal detector will reside in the top left corner (point (0,1)), for which true positive rate is 1 and false positive rate is 0. The closer a point is to this corner, the better the detector is.

Figure 3 shows 30 points (6 participants × 5 detector-threshold combinations).

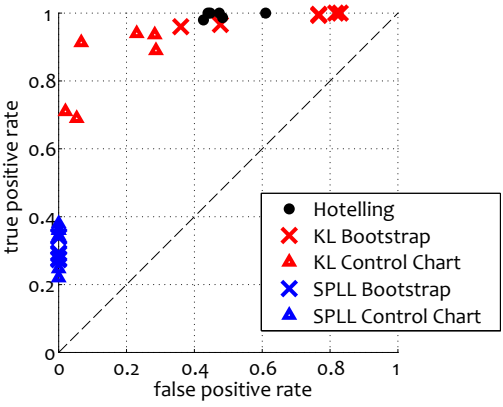


Fig. 3. Results for the 5 detector-threshold combinations. Each point is the average (FP,TP) for one participant, across the $K = 30$ iterations and 10 runs.

Each point corresponds to a participant. The marker and the colour indicate the detector-threshold combination. The figure shows that, although the detectors are not perfect individually, the points collectively form a high-quality ROC curve.

All thresholds were calculated for level of significance 0.05. Applying this threshold is supposed to restrict the false positives to that value. This happened only for the SPLL detector. The price for the zero FP-rate is a low sensitivity, making SPLL the most conservative of three detectors. The Hotelling detector

does not live up to the expectation of FP < 0.05. It is not guaranteed to have that FP rate if the assumptions of the test are not met - clearly the situation here. Between this test and KL with bootstrap threshold, Hotelling is both faster and more accurate (lower FP for the same TP). The best combination for our type of data appeared to be the KL detector with the control chart threshold. It exhibits an excellent compromise between FP and TP, and is faster to calculate.

Interestingly, the threshold-setting approach did not affect SPLL but did affect the KL-detector. The control chart approach improved on the original bootstrap approach by reducing dramatically the false positive rate without degrading substantially the true positive rate.

We note that the way we sampled W_1 and $W2$ may have induced some optimistic bias because the samples from the same label could be overlapping. This makes it easier for the detectors to achieve low FP rates than it would be in true streaming data. Nevertheless, this set-up did not favour any of the detectors or threshold-calculating methods, so the comparison is fair.

The execution time analyses favoured unequivocally the control-chart approach to finding a threshold. Also SPLL is the slowest of the detectors, followed by KL and Hotelling. Therefore we recommend the KL-detector with a control-chart threshold.

6 Conclusion

This paper examines the use of control charts as an alternative to the more traditional bootstrap approach for determining a threshold for change detectors. Our experimental study with a real-life dataset of facial expressions taken in real time favoured the KL-detector with a control chart threshold.

We also observed that the statistical significance of the thresholds (type I error) is not matched in the experiments, except for the SPLL detector. The non-parametric bootstrap approach, was expected to give a more robust threshold, not affected by a false assumption about the distribution of the change statistic. The opposite was observed in our experiments for the KL-detector. The reason for this could be that the window was too small to account for the variability of the data sampled from the same label. The results of the experiment led us to recommend the KL-detector with a control chart threshold for difficult streaming data such as facial expressions and behavioural analysis. SPLL with control chart threshold would be preferable where a conservative detector is needed. The same detection accuracy would be achieved with a bootstrap threshold but the extra computational expense is not justified.

Observing the excellent ROC curve shape offered by the collection of detectors, a combination of change detectors with different threshold-setting strategies looks a promising future research avenue. Investigation of adapting methods for classifier fusion to this problem is required, to assess the feasibility of creating a decision ensemble of change detectors.

References

1. Andrews, D.W.: Tests for parameter instability and structural change with unknown change point. Econometrica: Journal of the Econometric Society, 821–856 (1993)
2. Kifer, D., Ben-David, S., Gehrke, J.: Detecting change in data streams. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases. VLDB Endowment, vol. 30, pp. 180–191 (2004)
3. Shewhart, W.A.: Economic control of quality of manufactured product (1931)
4. Lowry, C.A., Woodall, W.H., Champ, C.W., Rigdon, S.E.: A multivariate exponentially weighted moving average control chart. Technometrics 34(1), 46–53 (1992)
5. Dasu, T., Krishnan, S., Venkatasubramanian, S., Yi, K.: An information-theoretic approach to detecting changes in multi-dimensional data streams. In: Proc. Symp. on the Interface of Statistics, Computing Science, and Applications, Citeseer (2006)
6. Kuncheva, L.: Change detection in streaming multivariate data using likelihood detectors. IEEE Transactions on Knowledge Data Engineering 25(5), 1175–1180 (2013)
7. Shewhart, W.A.: Quality control charts. Bell System Technical Journal 5, 593–603 (1926)
8. Wald, A.: Sequential analysis. Courier Dover Publications (1947)
9. Basseville, M., Nikiforov, I.V., et al.: Detection of abrupt changes: theory and application, vol. 104. Prentice Hall, Englewood Cliffs (1993)
10. Brodsky, B.E., Darkhovsky, B.S.: Nonparametric methods in change point problems, vol. 243. Kluwer Academic Pub. (1993)
11. Lowry, C.A., Montgomery, D.C.: A review of multivariate control charts. IIE Transactions 27(6), 800–810 (1995)
12. Hotelling, H.: Multivariate quality control. Techniques of statistical analysis (1947)
13. Zamba, K., Hawkins, D.M.: A multivariate change-point model for statistical process control. Technometrics 48(4), 539–549 (2006)
14. Song, X., Wu, M., Jermaine, C., Ranka, S.: Statistical change detection for multidimensional data. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 667–676. ACM (2007)
15. Kuncheva, L.I.: Change detection in streaming multivariate data using likelihood detectors. IEEE Transactions on Knowledge and Data Engineering 25 (2013)
16. Hotelling, H.: The generalization of Student's ratio. Annals of Mathematical Statistics 2(3), 360–378 (1931)
17. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 484–498. Springer, Heidelberg (1998)
18. Ahlberg, J.: Candide-3-an updated parameterised face (2001)
19. Bifet, A., Gavalda, R.: Learning from time-changing data with adaptive windowing. In: SDM, vol. 7 (2007)
20. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS (LNAI), vol. 3171, pp. 286–295. Springer, Heidelberg (2004)
21. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning 23(1), 69–101 (1996)

Commute Time for a Gaussian Wave Packet on a Graph

Furqan Aziz, Richard C. Wilson, and Edwin R. Hancock*

Department of Computer Science, University of York, YO10 5GH, UK
{furqan,wilson,erh}@cs.york.ac.uk

Abstract. This paper presents a novel approach to quantifying the information flow on a graph. The proposed approach is based on the solution of a wave equation, which is defined using the edge-based Laplacian of the graph. The initial condition of the wave equation is a Gaussian wave packet on a single edge of the graph. To measure the information flow on the graph, we use the average return time of the Gaussian wave packet, referred to as the wave packet commute time. The advantage of using the edge-based Laplacian of a graph over its vertex-based counterpart is that it translates results from traditional analysis to graph theoretic domain in a more natural way. Therefore it can be useful in applications where distance and speed of propagation are important.

Keywords: Edge-based Laplacian, wave equation, wave commute time, speed of propagation, graph complexity.

1 Introduction

One of the most challenging problems in the study of a complex network is to characterize the topological structure of a network, i.e., the way in which the nodes interact with each other. Each real-world network exhibits certain topological features that characterize its structure. Examples of such features are clustering coefficient, maximum degree, average degree, and average path-length. Over the recent years, researchers have developed different models that have similar properties as the real-world network. These models help us to understand or predict the structure of these systems. Examples of such models are scale-free networks [15] and small-world networks [14].

Recently, spectral methods have been successfully used for quantifying the complexity of a network. Passerini et al. [3] have used the spectrum of the normalized discrete Laplacian to define the von Neumann entropy associated with a graph. They have shown that this quantity can be used as the measure of the regularity of a graph. Han et al. [4] have approximated the von Neumann entropy using quadratic entropy and have shown that the approximate von Neumann entropy is related to the degree statistics of the graph. Escolano et al. [5] have used the diffusion kernel to quantify the intrinsic complexity of the undirected

* Edwin Hancock was supported by a Royal Society Wolfson Research Merit Award.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 374–383, 2014.

networks. They have also extended their work to directed networks [6]. Suau et al. [7] have analyzed the Schrödinger operator for characterizing the structure of a network. Lee et al. [9] have used the spectral methods to discover the genetic ancestry.

The structure of a complex network also plays an important role in the dynamics of information propagation. For this reason the study of a complex networks is becoming increasingly popular in epidemiology, where the goal is to study the mathematical models that can be used to simulate the infectious disease outbreaks in a social contact network. Grenfell [1] has discussed the traveling waves in measles epidemics. Abramson et al. [2] considered traveling waves of infection in the Hantavirus epidemics. Other real-life applications of information propagation over a network include the study of spreading a message over a social network and the study of a computer virus spreading over the internet [10].

While spectral method using discrete Laplacian have been successfully used, they suffer from certain limitations. Since the traditional graph Laplacian is an approximation of the continuous Laplacian to the discrete points, one of its limitations is that it cannot be used to translate most of the continuous results to a graph theoretic domain. For example the wave equation, defined using the discrete Laplacian, does not have finite speed of propagation. This makes it inappropriate for the applications that require spatial analysis or finite speed of propagation; e.g., spread of information in a network. The problem can be overcomed by treating edges of the network as real length intervals. This allows us to define a new kind of Laplacian, the edge-based Laplacian (EBL) of the graph [11][12]. The study of the edge-based Laplacian may be of great interest in appplications where the distance and speed of propagation are important.

In this paper our goal is to study the use of a wave equation, for the purpose of measuring the information flow across the network. The wave equation is defined using the edge-based Laplacian of a graph, where the initial condition is a Gaussian wave packet on a single edge of the graph. We define the wave packet hitting time, i.e., the time required for a wave packet to reach an edge f starting from an edge e, and the wave packet commute time, i.e., the time required for a wave packet to come back to the same edge from where it started. The remaining of this paper is organized as follows: We commence by introducing the edge-based Laplacian of a graph. Next we give a solution of a wave equation defined using the edge-based Laplacian, where the initial condition is a Gaussian wave packet. Based on the solution of wave equation we define wave packet hitting time (WHT) and wave packet commute time (WCT). Finally, in the experiment section, we apply the proposed method to different network models.

2 Edge-Based Laplacian of a Graph

Before introducing the edge-based Laplacian (EBL), in this section we provide some basic definitions and notations that will be used throughout the paper. A *graph* $G = (\mathcal{V}, \mathcal{E})$ consists of a finite nonempty set $\mathcal{V}$ of *vertices* and a finite set $\mathcal{E}$ of unordered pairs of vertices, called *edges*. A *directed graph* or a *digraph*

$D = (\mathcal{V}_D, \mathcal{E}_D)$ consists of a finite nonempty set $\mathcal{V}_D$ of vertices and a finite set $\mathcal{E}_D$ of ordered pairs of vertices, called *arcs*. So a digraph is a graph with an orientation on each edge. A digraph D is called *symmetric* if whenever (u, v) is an arc of D, (v, u) is also an arc of D. There is a one-to-one correspondence between the set of symmetric digraphs and the set of graphs, given by identifying an edge of the graph with an arc and its inverse arc on the digraph on the same vertices. We denote by $D(G)$ the symmetric digraph associated with the graph G. The *oriented line graph* is constructed by replacing each arc of $D(G)$ by a vertex. These vertices are connected if the head of one arc meets the tail of another, except that reverse pairs of arcs are not connected, i.e. $((u, v), (v, u))$ is not an edge.

We now define the EBL of a graph. The eigensystem of the EBL of a graph can be expressed in terms of the normalized adjacency matrix of a graph and the adjacency matrix of the oriented line graph [11][12]. Let $G = (\mathcal{V}, \mathcal{E})$ be a graph with a boundary ∂G. Let $\mathcal{G}$ be the geometric realization of G. The geometric realization is the metric space consisting of vertices $\mathcal{V}$ with a closed interval of length l_e associated with each edge $e \in \mathcal{E}$. We associate an edge variable x_e with each edge that represents the standard coordinate on the edge with $x_e(u) = 0$ and $x_e(v) = 1$. For our work, it will suffice to assume that the graph is finite with empty boundary (i.e., $\partial G = 0$) and $l_e = 1$. The eigenfunctions of the EBL are of two types; vertex-supported eigenfunctions and edge-interior eigenfunctions.

2.1 Vertex Supported Edge-Based Eigenfunctions

The vertex-supported eigenpairs of the EBL can be expressed in terms of the eigenpairs of the normalized adjacency matrix of the graph. Let A be the adjacency matrix of the graph G, and $\tilde{A}$ be the row normalized adjacency matrix. i.e., the $(i, j)th$ entry of $\tilde{A}$ is given as $\tilde{A}(i, j) = A(i, j) / \sum_{(k,j)\in\mathcal{E}} A(k, j)$. Let $(\phi(v), \lambda)$ be an eigenvector-eigenvalue pair for this matrix. Note $\phi(.)$ is defined on vertices and may be extended along each edge to an edge-based eigenfunction. Let ω^2 and $\phi(e, x_e)$ denote the edge-based eigenvalue and eigenfunction. Here $e = (u, v)$ represents an edge and x_e is the standard coordinate on the edge (i.e., $x_e = 0$ at v and $x_e = 1$ at u). Then the vertex-supported eigenpairs of the EBL are given as follows:

1. For each $(\phi(v), \lambda)$ with $\lambda \neq \pm 1$, we have a pair of eigenvalues ω^2 with $\omega = \cos^{-1}\lambda$ and $\omega = 2\pi - \cos^{-1}\lambda$. Since there are multiple solutions to $\omega = \cos^{-1}\lambda$, we obtain an infinite sequence of eigenfunctions; if $\omega_0 \in [0, \pi]$ is the principal solution, the eigenvalues are $\omega = \omega_0 + 2\pi n$ and $\omega = 2\pi - \omega_0 + 2\pi n, n \geq 0$. The eigenfunctions are $\phi(e, x_e) = C(e)\cos(B(e) + \omega x_e)$ where

$$C(e)^2 = \frac{\phi(v)^2 + \phi(u)^2 - 2\phi(v)\phi(u)\cos(\omega)}{\sin^2(\omega)}$$

$$\tan(B(e)) = \frac{\phi(v)\cos(\omega) - \phi(u)}{\phi(v)\sin(\omega)}$$

There are two solutions here, $\{C, B_0\}$ or $\{-C, B_0 + \pi\}$ but both give the same eigenfunction. The sign of $C(e)$ must be chosen correctly to match the phase.

2. $\lambda = 1$ is always an eigenvalue of $\tilde{A}$. We obtain a principle frequency $\omega = 0$, and therefore since $\phi(e, x_e) = C\cos(B)$ and so $\phi(v) = \phi(u) = C\cos(B)$, which is constant on the vertices.
3. If the graph is bipartite then $\lambda = -1$ is an eigenvalue of $\tilde{A}$. We obtain a principle frequency $\omega = \pi$, and therefore since $\phi(e, x_e) = C\cos(B + \pi x_e)$ and so $\phi(v) = -\phi(u)$, implying an alternating sign eigenfunction.

2.2 Edge-Interior Eigenfunctions

The edge-interior eigenfunctions are those eigenfunctions which are zero on vertices and therefore must have a principle frequency of $\omega \in \{\pi, 2\pi\}$. These eigenfunctions can be determined from the eigenvectors of the adjacency matrix of the oriented line graph.

1. The eigenvector corresponding to the eigenvalue $\lambda = 1$ of the oriented line graph provides a solution in the case $\omega = 2\pi$, and we obtain $|\mathcal{E}| - |\mathcal{V}| + 1$ linearly independent solutions.
2. Similarly the eigenvector corresponding to the eigenvalue $\lambda = -1$ of the oriented line graph provides a solution in the case $\omega = \pi$. If the graph is bipartite, then we obtain $|\mathcal{E}| - |\mathcal{V}| + 1$ linearly independent solutions. If the graph is non-bipartite, then we obtain $|\mathcal{E}| - |\mathcal{V}|$ linearly independent solutions.

This comprises all the principal eigenpairs which are only supported on the edges.

Note that although these eigenfunctions are orthogonal, they are not normlaized. To normalize these eigenfunctions we need to find the normalization factor corresponding to each eigenvalue and divide each eigenfunction with the corresponding normalization factor. Once normalized, these eigenfunctions form a complete set of orthonormal bases.

3 Wave Packet Commute Time

Recently, we have solved a wave equation on a graph, where the initial condition is a Gaussian wave packet on a single edge of a graph [8]. The wave equation is a second order partial differential equation, defined as

$$\frac{\partial^2 u}{\partial t^2}(\mathcal{X}, t) = \Delta_E u(\mathcal{X}, t), \tag{1}$$

where Δ_E is the EBL, and $\mathcal{X}$ represents the value of a standard coordinate x on an edge e. Let ω^2 represents the eigenvalue of the EBL with the corresponding eigenfunction $\phi_{\omega,n}(\mathcal{X}) = C(e, \omega)\cos\left(B(e, \omega) + \omega x + 2\pi n x\right)$. The complete solution is given as [8]

$$
\begin{aligned}
u(\mathcal{X},t) = \sum_{\omega\in\Omega_a} \frac{C(\omega,e)C(\omega,f)}{2} \Big(& e^{-a\mathcal{W}(x+t+\mu)^2} \\
& \cos\left[B(e,\omega)+B(f,\omega)+\omega\left\lfloor x+t+\mu+\frac{1}{2}\right\rfloor\right] \\
& + e^{-a\mathcal{W}(x-t-\mu)^2}\cos\left[B(e,\omega)-B(f,\omega)+\omega\left\lfloor x-t-\mu+\frac{1}{2}\right\rfloor\right]\Big) \\
& + \frac{1}{2|E|}\left(e^{-a\mathcal{W}(x+t+\mu)^2}+e^{-a\mathcal{W}(x-t-\mu)^2}\right) \\
& + \sum_{\omega\in\Omega_b} \frac{C(\omega,e)C(\omega,f)}{4}\left(e^{-a\mathcal{W}(x-t-\mu)^2}-e^{-a\mathcal{W}(x+t+\mu)^2}\right) \\
& + \sum_{\omega\in\Omega_c} \frac{C(\omega,e)C(\omega,f)}{4}\Big((-1)^{\lfloor x-t-\mu+\frac{1}{2}\rfloor}e^{-a\mathcal{W}(x-t-\mu)^2} \\
& -(-1)^{\lfloor x+t+\mu+\frac{1}{2}\rfloor}e^{-a\mathcal{W}(x+t+\mu)^2}\Big). \qquad (2)
\end{aligned}
$$

Here $\mathcal{W}(z)$ wraps the value of z to the range $[-\frac{1}{2},\frac{1}{2})$, and $\lfloor z\rfloor$ is the floor function.

Once we have the solution of the wave equation, we can define a number of interesting invariants to understand the properties of the flow of information across the network. This also helps us to quantify the structure of the network. We commence by defining the wave packet commute time of a graph. Given a graph $G=(\mathcal{E},\mathcal{V})$ we define the wave packet commute time (WCT) of an edge e as follows. Assume that the initial condition of the wave equation is a Gaussian wave packet on the edge $e\in\mathcal{E}$ and zero elsewhere. Then

$$\mathrm{WCT}(e) = \min_{t>0}\{t : u(e,0.5) > \delta\}, \qquad (3)$$

i.e., the WCT is the time when the wave packet with amplitude at least δ (at the middle of the edge), returns back to the edge e. Figure 4(a) demonstrates the wave commute time for a simple graph with 5 nodes and 7 links. Here the initial condition is a Gaussian wave packet on the edge $e1$ of the graph. The bottom right figure shows the fraction of the wave packet returned back at time $t=3$. Note that at time $t=1$, a wave packet with negative amplitude (a trough) returns to the edge $e1$. A trough will always be created when a wave packet is traveling along an edge (u,v) in the directed of v, and the degree of v is at least 3.

Edge-commute time can also be defined in terms of the hitting time of the wave packet. Given two edges $e,f\in\mathcal{E}$, the wave packet hitting time (WHT) can be defined as follows. Assume that the initial condition of the wave equation is a Gaussian wave packet on the edge $e\in\mathcal{E}$ and zero elsewhere. Then

$$\mathrm{WHT}(e,f) = \min_{t>0}\{t : u(f,0.5) > \delta\}, \qquad (4)$$

i.e., the WHT is the time when the wave packet with amplitude at least δ (at the middle of the edge), reaches the edge f, starting from edge e. The edge-commute time can then be defined as:

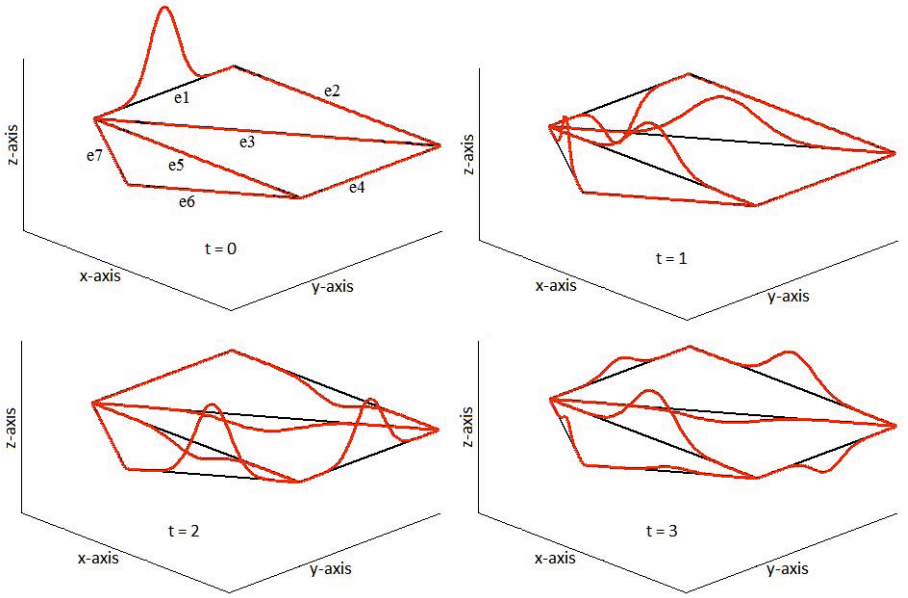


Fig. 1. Commute time of a Gaussian wave packet on a graph

$$\mathrm{WCT}(e) = \frac{1}{|\mathcal{E}|} \sum_{f \in \mathcal{E}} WHT(e, f), \tag{5}$$

i.e., the WCT for the edge e is the average of the WHT over all the edges of the graph. However, the WCT defined using the WHT is computationally more expensive, and therefore in the experiment section we use the WCT defined in Equation 3.

To quantify the complexity of a network, we define a global invariant based on the WCT as:

$$\mathrm{GWCT}(G) = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} WCT(e), \tag{6}$$

i.e., GWCT of a network is the average of the WCT over all the links of the network. In the next section we will show that GWCT provides a good measure for distinguishing graphs with different structures.

4 Experiments

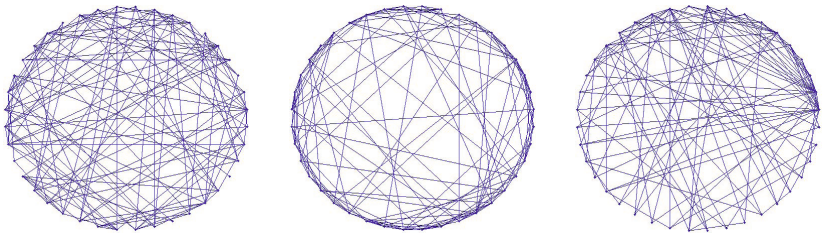
In this section we study the flow of information across a network using WCT and WHT and demonstrate the ability of GWCT to distinguish graphs with different structural properties. We experiment our proposed method on the following three different types of network models.

Erdős-Rényi Model(ER) [13]: An ER graph $G(n, p)$ is constructed by connecting n vertices randomly with probability p. i.e., each edge is included in the graph with probability p independent from every other edge. These models are also called $r$andom networks.

Watts and Strogatz Model(WS) [14]: A WS graph $G(n, k, p)$ is constructed in the following way. First construct a regular ring lattice, a graph with n vertices and each vertex is connected to k nearest vertices, k/2 on each side. Then for every vertex take every edge and rewire it with probability p. These models are also called $s$mall-world networks.

Barabási-Albert Model(BA) [15]: A BA graph $G(n, n_0, m)$ is constructed by an initial fully connected graph with n_0 vertices. New vertices are added to the graph one at a time. Each new vertex is connected to m previous vertices with a probability that is proportional to the number of links that the existing nodes already have. These models are also called $s$cale-free networks.

Figure 2 shows an example of each of these models.



(a) random graphs (b) small-world graph (c) scale-free graph

Fig. 2. Graph models

As mentioned earlier, one of the advantages of the wave equation defined using the EBL is that it has finite speed of propagation [11]. This makes it suitable for applications that require finite speed of propagation. In our first experiment, we demonstrate the ability of edge-based wave equation for identifying infected links in a network. For this purpose, we generate a BA network and a WS network each with 60 nodes and 175 links. We have computed the WHT for each edge of the graph starting from an edge e. The edge $e = (u, v) \in \mathcal{E}$ is selected, such that u is the highest degree vertex in the graph and v is the highest degree vertex in the neighbours of u. Figure 3 shows the cumulative frequencies of infected links for both graphs with different values of δ. As expected, the cumulative number of infected links decreases as δ increases. Note that the links in WS network are infected quickly than links in BA network. This is due to the presence of hub in BA network, which distributes the wave packet with small amplitudes to more links. The WS network, on the other hand, has more regular structure that allows the wave packet to transmit across the network with high amplitudes.

The above experiment shows that the WCT behaves differently on different graphs. This suggests that the WCT can be used to quantify the structure of a

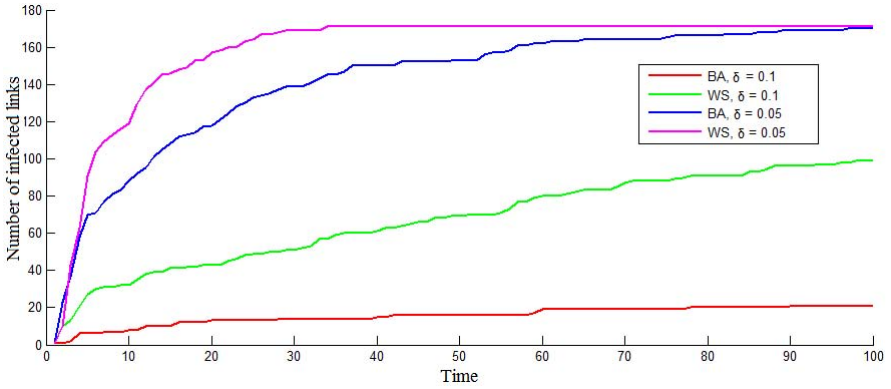


Fig. 3. Number of links infected with time

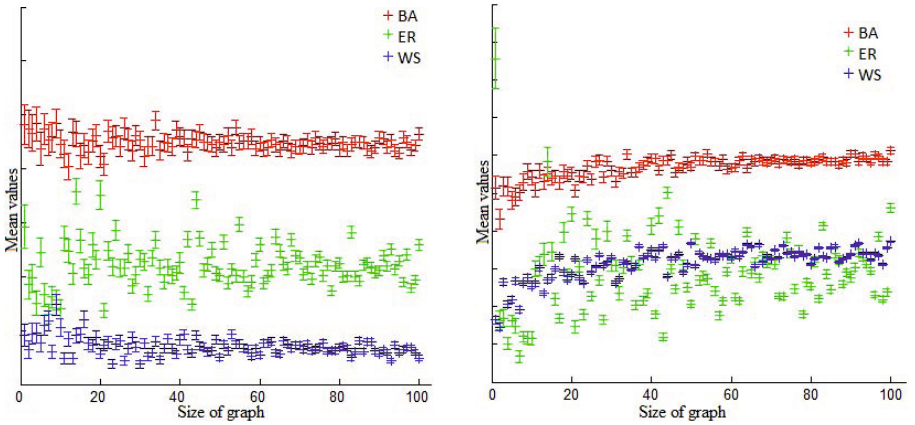
complex network. In our next experiment, we demonstrate the ability of WCT to distinguish networks with different substructures. For this purpose, we generate 100 graphs for each model with $n = 50 + (d-1)k$ with $k = 1, 2, ..., 100$, where n is the number of vertices. We have chosen the other parameters in such a way so that all three types of graphs with the same number of vertices have approximately the same number of edges. For ER models we choose $p = 10/n$, for WS models we choose $p = 0.25$ and $k = 8$, and for BA models we choose $n_0 = 5$ and $k = 4$. For each graph we compute the wave commute time and average it over all the edges. Figure 4(a) shows the average value for the three different types of graphs. Results suggest that the wave commute time is highly robust in distinguishing the graphs with different structures.

Figure 4(b) shows a similar analysis for vertex commute time, which is defined as the expected number of steps for a random walk starting from a vertex u, hits vertex v and then returns to u. The commute time of a vertex u to a vertex v can be computed from the eigenvalues and eigenvectors of the normalized Laplacian. Let (λ, ϕ) be the eigenpair of the normalized Laplacian. Then the commute time is defined as:

$$CT(u,v) = \sum_{i=2}^{|\mathcal{V}|} \left(\sqrt{\frac{vol}{\lambda_i d_u}} \phi_i(u) - \sqrt{\frac{vol}{\lambda_i d_v}} \phi_i(v) \right)^2, \tag{7}$$

where d_u represents the degree of the vertex u and vol represents the sum of degrees for an unweighted graph. The x-axis in Figure 4(b) shows the average commute time over all vertices.

The mean and the standard error of the edge-commute time depend on the regularity structure of the graph. As the regularity of the graph increases, the value of the standard error decreases. Note that the value of WCT depends on the size of the smallest cycle to which the edge belongs. Figure 5 shows the mean values and the standard errors for the graphs generated in the previous experiment. Since WS networks are more regular as compared to BA networks, they therefore have smaller standard errors. Also, if the probability p of rewiring



(a) Wave commute time (b) Vertex commute time

Fig. 4. Wave commute time vs commute time

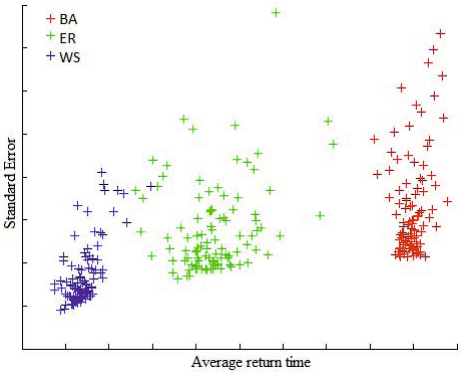


Fig. 5. Mean values and standard errors

is kept low, then the WS network has more small length cycles. Therefore the mean values of WS networks are small as compared to BA networks. Note that ER graphs exhibit more variation in the mean values due to their random structure. Their mean and standard error values lie between that of the BA graphs and the WS graphs.

5 Conclusion

In this paper we have studied the properties of the commute time (WCT) and the hitting time (WHT) of a Gaussian wave packet on a graph. The WCT and WHT are based on the solution of the wave equation defined using the edge-based Laplacian of a graph where the initial condition is a Gaussian wave packet

on a single edge of the graph. We have shown the application of WCT and WHT for quantifying the structure and information flow of a network. The advantage of using the edge-based Laplacian (EBL) is that this approach is more closely related to mathematical analysis than the usual discrete Laplacian. This allows us to implement equation on graphs which have finite speed of propagation.

References

1. Grenfell, B.T.: Travelling waves and spatial hierarchies in measles epidemics. Nature, 716-723 (2001)
2. Abramson, G., Kenkre, V.M., Yates, T.L., Parmenter, R.R.: Traveling Waves of Infection in the Hantavirus Epidemics. Bulletin of Mathematical Biology, 519–534 (2003)
3. Passerini, F., Severini, S.: The von neumann entropy of networks. International Journal of Agent Technologies and Systems, 58–67 (2009)
4. Han, L., Escolano, F., Hancock, E.R., Wilson, R.C.: Graph characterizations from von Neumann entropy. Pattern Recognition Letters, 1958–1967 (2102)
5. Escolano, F., Hancock, E.R., Lozano, M.A.: Heat diffusion: Thermodynamic depth complexity of networks. Physics Review E, 036206 (2012)
6. Escolano, F., Bonev, B., Hancock, E.R.: Heat Flow-Thermodynamic Depth Complexity in Directed Networks. In: Gimel'farb, G., Hancock, E., Imiya, A., Kuijper, A., Kudo, M., Omachi, S., Windeatt, T., Yamada, K. (eds.) SSPR&SPR 2012. LNCS, vol. 7626, pp. 190–198. Springer, Heidelberg (2012)
7. Suau, P., Hancock, E.R., Escolano, F.: Analysis of the Schrödinger Operator in the Context of Graph Characterization. In: Hancock, E., Pelillo, M. (eds.) SIMBAD 2013. LNCS, vol. 7953, pp. 190–203. Springer, Heidelberg (2013)
8. Aziz, F., Wilson, R.C., Hancock, E.R.: Gaussian Wave Packet on a Graph. In: Kropatsch, W.G., Artner, N.M., Haxhimusa, Y., Jiang, X. (eds.) GbRPR 2013. LNCS, vol. 7877, pp. 224–233. Springer, Heidelberg (2013)
9. Lee, A.B., Luca, D., Klei, L., Devlin, B., Roeder, K.: Discovering genetic ancestry using spectral graph theory. Genetic Epidemiology, 51-59 (2010)
10. Bradonjic, M., Molloy, M., Yan, G.: Containing Viral Spread on Sparse Random Graphs: Bounds, Algorithms, and Experiments. Internet Mathematics, 406–433 (2013)
11. Friedman, J., Tillich, J.P.: Wave equations for graphs and the edge based Laplacian. Pacific Journal of Mathematics, 229–266 (2004)
12. Wilson, R.C., Aziz, F., Hancock, E.R.: Eigenfunctions of the edge-based Laplacian on a graph. Journal of Linear Algebra and its Applications, 4183–4189 (2013)
13. Erdõs, P., Rényi, A.: On the evolution of random graphs. Publications of the Mathematical Institute of the Hungarian Academy of Sciences, 17–61 (1960)
14. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature, 440–442 (1998)
15. Barabási, A., Albert, R.: Emergence of Scaling in Random Networks. Science 509–512 (1999)

Properties of Object-Level Cross-Validation Schemes for Symmetric Pair-Input Data

Juho Heimonen[1,2], Tapio Salakoski[1,2], and Tapio Pahikkala[1,2]

[1] TUCS - Turku Centre for Computer Science, Turku, Finland
[2] University of Turku, Turku, Finland
firstname.lastname@utu.fi

Abstract. In bioinformatics, many learning tasks involve pair-input data (i.e., inputs representing object pairs) where inputs are not independent. Two cross-validation schemes for symmetric pair-input data are considered. The mean and variance of cross-validation estimate deviations from respective generalization performances are examined in the situation where the learned model is applied to pairs of two previously unseen objects. In experiments with the task of learning protein functional similarities, large positive mean deviations were observed with the *relaxed* scheme due to training–validation dependencies while the *strict* scheme yielded small negative mean deviations and higher variances. The properties of the strict scheme can be explained by the reduction in cross-validation training set sizes when avoiding training–validation dependencies. The results suggest that the strict scheme is preferable in the given setting.

Keywords: cross-validation, pair-input, AUC, K-Nearest Neighbor.

1 Introduction

In supervised learning, the generalization performance is commonly estimated by training a model on one part of the dataset (training set) and evaluating it against another (validation set) to avoid optimistically biased estimates. Cross-validation (CV) is a procedure to estimate the generalization performance by aggregating the results of several such evaluations. [2].

A CV procedure consists of folds, each of which involving training and evaluating a model according to a training–validation split of the dataset. Since an *input* (i.e., a data point) can belong to the training set of one fold and to the validation set of another, CV can be used when the small size of the dataset prevents from obtaining large enough training and validation sets in a single split [6]. The properties of a CV estimator are influenced by the splitting scheme as well as how the performance is measured.

In a general case, CV procedures assume that data are identically distributed and the training set is independent from the validation set [2]. The conventional approach of randomly partitioning data into training and validation sets is not viable when the data contain dependencies [12].

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 384–393, 2014.

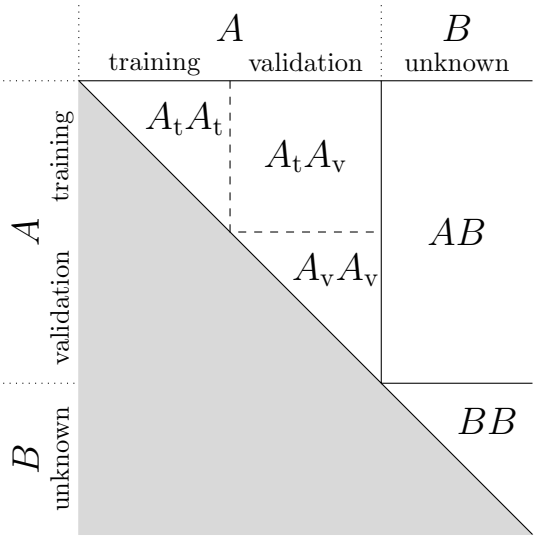


Fig. 1. There are three types of pairs in a symmetric pair-input learning task. The set A contains those objects that are pair members in the dataset on which a model is trained and cross-validated. The set B contains the objects not present in the dataset. The AA, AB, and BB types of pairs differ in the number of members seen in the dataset. The types A_tA_t, A_tA_v, and A_vA_v are the analogous types within a CV fold with the subscripts referring to the training (t) and validation (v) sets.

This study explores the properties of CV estimators in the case of symmetric pair-input data. Pair-input data consist of inputs that represent pairs of *objects* while symmetry refers to pair members being of a single type with a symmetric relation. Among others, data of this type are encountered in bioinformatics when considering the properties of protein pairs, such as binding [13] or functional similarity. Research in biosciences typically focuses on specific aspects of organisms and knowledge is consequently centered around a subset of proteins. Since the protein pairs of which a particular property is known stem from a limited set of proteins, it is common that a protein is a pair member in several inputs which leads to strong dependencies (see, for example, [13]).

Object pairs were categorized in [13] by their composition with respect to a given dataset. Figure 1 illustrates these three types: both members (AA), one member (AB), or no members (BB) belonging to the set A that contains the objects present in the dataset. It was observed that the CV estimator of the generalization performance of a model learned from AA pairs using a conventional scheme is acceptable when considering the performance on AA pairs but optimistically biased when considering the performance on AB or BB pairs [13].

This study examines two CV schemes in the situation where predictions will be made on BB pairs. They differ from conventional splitting schemes in that the splitting is performed on objects, not on inputs, and validation sets are formed

based on the selected objects. The *relaxed* scheme, involving models trained on the union of $A_\mathrm{t}A_\mathrm{t}$ and $A_\mathrm{t}A_\mathrm{v}$ (see Fig. 1), is expected to be optimistically biased because validation set inputs are exposed via shared pair members whereas the *strict* scheme, involving models trained on $A_\mathrm{t}A_\mathrm{t}$ and evaluated against $A_\mathrm{v}A_\mathrm{v}$, should not exhibit an optimistically biased behavior because the setup is analogous to learning from AA pairs to predict BB pairs. The strict scheme is expected to be pessimistically biased because the full model is trained on more data than the CV models [1] and have higher variance than the relaxed scheme because its training sets contain less data [11].

Experiments are performed on the prediction of the functional similarity of two proteins from their sequences. While not a typical formulation of the protein function prediction task, which is one of the major tasks in bioinformatics [9], functional similarity serves as an example of a symmetric pair-input problem.

2 Cross-Validation Schemes

Let $\mathcal{O}$ be a set of objects and $\mathcal{Z} \subset \mathcal{X} \times \mathcal{Y}$ a set of instances, where the input space $\mathcal{X} = \mathcal{O}^2$ and the output space $\mathcal{Y} = \{-1, 1\}$. An instance $z = (x, y) \in \mathcal{Z}$ consists of an input $x = (o, o') \in \mathcal{O}^2$ and its associated label y such that $y = 1 \iff x \in \mathcal{R}$, where $\mathcal{R} \subseteq \mathcal{O}^2$ is the symmetric relation of interest. A sequence $Z = ((x_1, y_1), \ldots, (x_n, y_n)) \in \mathcal{Z}^n$, where $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$ are the input and label sequences, respectively, is called a training set. The set $\mathcal{O}_Z = \{o : (\exists i)(X_i = (o, o') \vee X_i = (o', o))\}$ is the set of training set objects. An input cannot be associated with both labels. That is, $(x, y) \in \mathcal{Z} \implies (x, -y) \notin \mathcal{Z}$. Also, $(\exists i)(X_i = (o, o')) \implies (\nexists j)(X_j = (o', o))$ because the inputs (o, o') and (o', o) are assumed to have identical representations in addition to their associated labels being identical due to symmetry. Instances having $y = 1$ are called positive instances and those having $y = -1$ negative instances. Let $D_\mathcal{O}$ and $D_\mathcal{Z}$ be probability distributions over $\mathcal{O}$ and $\mathcal{Z}$, respectively.

The outputs of a prediction function $f_Z : \mathcal{X} \to \mathbb{R}$, learned from the training set Z, rank the inputs by how likely their associated $y = 1$. The generalization performance of a prediction function is measured by its conditional expected area under the ROC curve (AUC) [1]

$$\mathrm{A}(f_Z) = \mathrm{E}_{z_+ \sim D_+, z_- \sim D_-}[\mathrm{H}(f_Z(x_+) - f_Z(x_-))] \ , \tag{1}$$

where $z_+ = (x_+, 1)$, $z_- = (x_-, -1)$, and H is the Heaviside step function with $H(0) = \frac{1}{2}$, while D_+ and D_- are the conditional distributions of instances derived from $D_\mathcal{Z}$ given $y = 1$ and $y = -1$, respectively.

In each CV fold, a validation set $\mathcal{O}_V$ of objects is picked such that $\mathcal{O}_V \subset \mathcal{O}_Z$. The validation set V of instances is a subsequence of Z such that $(\exists i)(V_i = (x, y)) \iff ((\exists j)(Z_j = (x, y)) \wedge o \in \mathcal{O}_V \wedge o' \in \mathcal{O}_V)$, where $x = (o, o')$, (see $A_\mathrm{v}A_\mathrm{v}$ in Fig. 1) while the training set T of instances is a subsequence of Z. In the relaxed scheme $(\exists i)(T_i = (x, y)) \iff ((\exists j)(Z_j = (x, y)) \wedge (o \notin \mathcal{O}_V \vee o' \notin \mathcal{O}_V))$ (see $A_\mathrm{t}A_\mathrm{t}$ and $A_\mathrm{t}A_\mathrm{v}$ in Fig. 1) while in the strict scheme $(\exists i)(T_i = (x, y)) \iff$

$((\exists j)(Z_j = (x, y)) \wedge o \notin \mathcal{O}_V \wedge o' \notin \mathcal{O}_V)$ (see $A_t A_t$ in Fig. 1). The sequence $C = ((V_1, T_1), \ldots, (V_n, T_n))$ contains the validation and training set pairs of the n folds.

The CV performance $\hat{\mathrm{A}}_{\mathrm{CV}}(Z)$ is an estimator of $\mathrm{A}(f_Z)$ obtained from Z using the learning algorithm that yielded f_Z. The quality of a CV scheme is evaluated using the mean and variance of the deviation $\mathrm{B}(Z) = \hat{\mathrm{A}}_{\mathrm{CV}}(Z) - \mathrm{A}(f_Z)$ which follows the approach taken, for example, in [6] and [1]. The second moment about zero of $\mathrm{B}(Z)$ is also considered.

3 Estimation of AUC

The properties of $\hat{\mathrm{A}}_{\mathrm{CV}}(Z)$ are influenced by how the validation set V and the training set T are selected in each fold but also by how cross-validation AUC is calculated. The choice between the relaxed and strict schemes affects T, which is the focus of this study, while the selection of $\mathcal{O}_V$ affects both V and T.

Two methods to calculate cross-validation AUC are considered: averaging and pooled AUC [5,1]. The former is the mean AUC over folds whereas the latter is calculated from the concatenation of the predictions made in the folds.

In an earlier study, AUC estimators were analyzed in a non-pair-input situation. Non-zero mean deviations were observed for pooled AUC on certain kinds of data which was attributed to predictions from several models being compared although strictly not compatible. Also, estimators involving more comparisons of positive–negative instance pairs were observed to have lower variance than those with fewer comparisons. [1].

Object-leave-two-out CV includes a fold for each of the $\binom{m}{2}$ possible validation sets fulfilling the condition $|\mathcal{O}_V| = 2$, where $m = |\mathcal{O}_Z|$. If $\mathcal{X}$ is restricted to the inputs (o, o') such that $o \neq o'$ (like in this study, see Sect. 4.3), each validation set contains only one instance and all instances are included in exactly one validation set.

In object-n-fold CV, $\mathcal{O}_Z$ is partitioned into n parts of approximately equal sizes with the ith part being $\mathcal{O}_V$ in the ith fold. Consequently, some instances do not belong to any of the validation sets and the number of excluded instances increases as n increases (Fig. 2). To cover all instances, overlapping validation sets can be selected such that two parts form a validation set in each fold which results in $\binom{n}{2}$ folds. In this case, however, the pairs in which the members are from the same part appear in $n - 1$ validation sets while the other pairs appear only once (diagonal blocks vs. non-diagonal blocks in Fig. 2). As n approaches m, overlapping object-n-fold CV approaches object-leave-two-out CV (Fig. 2).

4 Experiments

The properties of the relaxed and strict schemes were investigated by conducting experiments on learning protein functional similarities. A protein is a biomolecule composed of amino acid chains folded into a three-dimensional structure that is capable of accomplishing (possibly jointly with other proteins) a particular

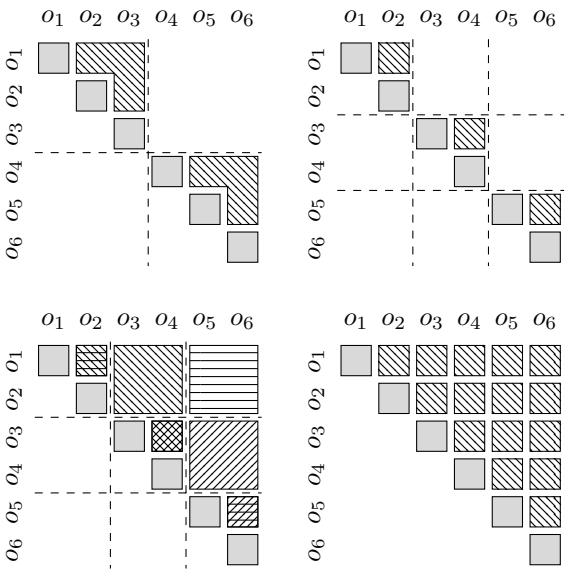


Fig. 2. The validation sets (*patterned areas*) of object-2-fold CV (*upper left*) cover more instances than those of object-3-fold CV (*upper right*). Both overlapping object-3-fold CV (*lower left*) and object-leave-two-out CV (*lower right*) cover all instances but three instances are present in two validation sets, distinguished by pattern types, in the former. *Dashed lines* indicate the boundaries of the parts of $\mathcal{O}_Z = \{o_1, \ldots, o_6\}$ while *grey squares* represent the excluded (o, o) pairs.

task. How the amino acid sequence (and the structure) of a protein defines its function is one of the major topics in bioinformatics.

The task of protein function prediction can be formulated as one of predicting the function of a protein from its sequence [8] while other sources of information may also be utilized as well [4,7]. In this study, instead of directly predicting the function, the functional similarity of two proteins is considered because it fulfills the requirements of symmetric pair-input data.

4.1 Data

Datasets were derived from the Universal Protein Resource[1] (UniProt) [14]. UniProt entries contain amino acid sequences of proteins together with diverse annotations, literature references, and cross-references to other databases. Its UniProtKB/Swiss-Prot section contains manually curated entries while the UniProtKB/TrEMBL section contains unreviewed, computer-annotated entries. Only the former was used in the experiments in order to minimize noise.

[1] `http://www.uniprot.org/`

The functional similarity of two proteins was determined by their Gene Ontology annotations. Gene Ontology[2] (GO) [3] is a comprehensive classification and widely adopted in bioinformatics. It provides hierarchical controlled vocabularies for three complementary domains – molecular function, biological process, and cellular component – referenced by UniProt entries.

Three datasets were created by considering one of the GO domains at the time and the fourth by considering the domains jointly. The information regarding the function was assumed to be complete when an entry had any GO annotation belonging to the given domain(s). All such proteins were included in the dataset while the others were discarded to avoid false negative labels. This produced datasets ranging approximately from 387,000 to 511,000 proteins in size.

4.2 Features and Labels

Each protein sequence was represented by a vector containing the frequencies of amino acids as well as the frequencies of bigrams of adjacent amino acids categorized into four classes according to [10]. A protein pair was represented by the sum of the two protein feature vectors. This low-dimensional representation is more suitable for the K-Nearest Neighbor classifiers used in the experiments (see Sect. 4.3) than high-dimensional representations.

A protein pair was labeled positive if its members had any GO annotation in common. The hierarchy of GO classes was not taken into account.

4.3 Experiment Details

The set $\mathcal{Z}$ was defined as the set of instances covering all protein pairs (o, o') such that $o \neq o'$ to avoid trivially positive instances skewing performance scores. Both $D_{\mathcal{O}}$ and $D_{\mathcal{Z}}$ were chosen to be uniform distributions. Since its exact value is impractical to calculate, the conditional expected AUC was estimated from a random sample S with the Wilcoxon–Mann–Whitney statistic [5]. For each dataset, the sequence S was drawn without replacement from $\mathcal{Z}$ such that $|S| = 10^4$. Let $\mathcal{O}_S = \{o : (\exists i)(S_i = (x, y) \wedge (x = (o, o') \vee x = (o', o)))\}$.

The relaxed and strict schemes were evaluated with all possible combinations of the four datasets, two validation set selection methods (object-ten-fold or object-leave-two-out), and two AUC calculation methods (averaging or pooled). Note that averaging AUC cannot be calculated in the object-leave-two-out case because each validation set contains only one instance.

The sampling distribution of deviations was obtained from one thousand independent repeats. In each repeat, a sequence $O = (o_1, \ldots, o_n)$ of 100 proteins was conditionally drawn without replacement from $\mathcal{O}$ given that $o_i \notin \mathcal{O}_S$. The training set Z was formed by including the inputs (o, o') fulfilling the condition $(\exists i)(O_i = o) \wedge (\exists j)(O_j = o')$.

In all experiments, K-Nearest Neighbor classifiers were trained with inverse distance weighing. The parameter K was varied from $K = 10$ to $K = 100$ in steps of ten to analyze its effect.

[2] `http://www.geneontology.org/`

Table 1. The observed mean deviations for $K = 50$. *10x* and *LTO* refer to object-ten-fold and object-leave-two-out CV while *A* and *P* refer to averaging and pooled AUC, respectively.

	Relaxed			Strict		
Dataset	A-10x	P-10x	LTO	A-10x	P-10x	LTO
Union	0.1799	0.1870	0.1919	−0.0113	−0.0209	−0.0187
Molecular function	0.1509	0.1614	0.1642	−0.0167	−0.0282	−0.0285
Biological process	0.0673	0.0909	0.0907	−0.0508	−0.0402	−0.0410
Cellular component	0.1745	0.1797	0.1842	−0.0220	−0.0290	−0.0246

Table 2. The observed variances of deviations for $K = 50$. *10x* and *LTO* refer to object-ten-fold and object-leave-two-out CV while *A* and *P* refer to averaging and pooled AUC, respectively.

	Relaxed			Strict		
Dataset	A-10x	P-10x	LTO	A-10x	P-10x	LTO
Union	0.0020	0.0017	0.0009	0.0028	0.0028	0.0015
Molecular function	0.0032	0.0025	0.0012	0.0051	0.0046	0.0025
Biological process	0.0111	0.0086	0.0027	0.0160	0.0147	0.0064
Cellular component	0.0016	0.0015	0.0009	0.0027	0.0028	0.0016

5 Results and Discussion

The relaxed and strict schemes resulted in positive and negative mean deviations, respectively, and the experiments with the relaxed scheme yielded lower variances of deviations than their counterparts with the strict scheme. Increases in K resulted in decreases in the means in both schemes, though the effect was minor in the strict scheme, while the variances increased in the strict scheme and decreased in the relaxed scheme. The peak generalization performance was reached in the given range of K with the *Union* and *Cellular component* datasets.

Illustrating typical observations, Tables 1 and 2 show the means and variances, respectively, of the observed deviations of CV estimates from respective (estimated) generalization performances for $K = 50$. The means of the observed generalization performances for $K = 50$ are 0.5857, 0.6524, 0.7423, and 0.6351, in the order of the datasets in the tables.

The absolute values of the deviation means are generally approximately an order of magnitude lower and the deviation variances higher but of the same order of magnitude in the experiments with the strict scheme than in their counterparts with the relaxed scheme. The *Biological process* dataset differs from the others by having notably lower absolute values in the relaxed setting, higher absolute values in the strict setting, and higher variances in both settings.

The above observations are reflected in the second moments about zero being approximately an order of magnitude lower in the experiments with the strict

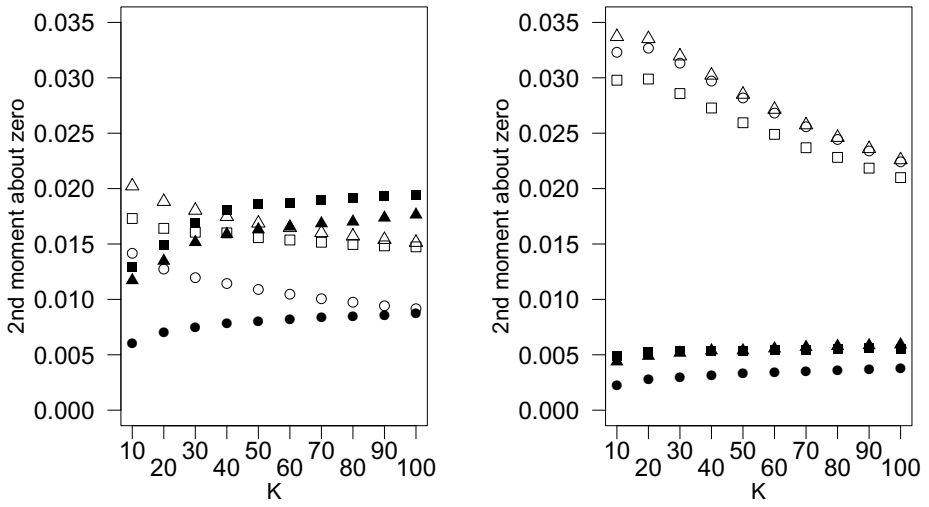


Fig. 3. The second moments about zero of the relaxed and strict schemes become equal at approximately $K = 30$, $K = 60$, or $K = 100$ in the *Biological process* dataset (*left*) whereas they are well-separated in the *Molecular function* dataset (*right*). Triangle, square, and circle refer to *P-10x*, *A-10x*, and *LTO* (see Tables 1 and 2) whereas hollow and solid symbols denote the relaxed and strict schemes, respectively.

scheme than in their counterparts with the relaxed scheme in all except the *Biological process* dataset. The changes in mean and in variance as K increases both contribute toward the decreasing and increasing trends in the second moments seen with the relaxed and strict schemes, respectively. However, as illustrated in Figure 3, the point after which the relaxed scheme yields lower second moments depends on the dataset and the CV details.

The absolute values of the deviation means are higher in the experiments with pooled AUC than in their counterparts with averaging AUC in all but one experiment pair. This is not surprising given that pooling can have either a positive or negative effect on deviations [1]. An increase in the number of positive–negative instance comparisons (*A-10x* < *P-10x* < *LTO*, see Table 2) generally has a decreasing effect on variance, as expected, although *A-10x* and *P-10x* are in the opposite order in two experiment pairs for high K values.

The observed deviation means suggest that the positive effect of training–validation dependencies generally dominates over the negative effect of the reduced size of training sets and, consequently, that the strict scheme is preferable to the relaxed scheme in the setting where the learned model will be applied to pairs of two previously unseen objects. However, given the limited number of experiments in this study, it remains unanswered to what extend these observations can be generalized to other datasets and/or learning algorithms. Particularly, the results obtained with the *Biological process* dataset raises the question whether

the unexpectedly small differences in the absolute values of the mean deviations between the two schemes are due to the properties of the dataset or due to the schemes generally yielding more similar absolute values as generalization performance increases. In the latter case, the strict scheme would not necessarily be preferable at high performance levels although it would still have the advantage of yielding conservative estimates.

5.1 Future Directions

The results of this study illustrate the potential of the strict scheme. In future experiments, the scheme will be applied to a variety of learning algorithms and datasets to get a better understanding of its behavior. With preliminary results from another dataset suggesting otherwise, it is of particular interest to investigate whether higher absolute values of deviations should be expected at higher levels of generalization performance as is hinted by the *Biological process* dataset. Different approaches to select validation sets (see, for example, [2]) will also be examined in order to discover their properties when operating on objects instead of on instances. Last, the analysis of the strict scheme will be expanded to the experimental setup outlined in [13] where some pairs of objects are not included in the dataset due to incomplete knowledge of objects.

The two schemes considered in this study are expected to fail to reliably estimate the generalization performance of a learned model when predictions will be made on inputs where an object seen in the dataset is paired with a previously unseen object (AB pairs in Fig. 1). Adapting the strict scheme to this setting likely requires only minor modifications.

6 Conclusions

Two CV schemes for symmetric pair-input data were considered. They differ from conventional CV schemes by acknowledging the fact that inputs represent pairs of objects. They first make training–validation splits on objects and then use the selected objects to form training and validation sets. The strict scheme avoids dependencies between the training and validation sets that would arise from shared pair members by discarding offending instances from the training sets. Consequently, its folds are analogous to learning a model from a dataset and making predictions on pairs that are composed of objects not encountered in the dataset. The relaxed scheme utilizes all instances in each fold and is hence similar to conventional CV schemes that assume independent instances.

The properties of the relaxed and strict schemes were examined in the task of learning functional similarities of proteins. Four datasets were derived from UniProt database and evaluated using various combinations of AUC calculation method and validation set selection method. Positive mean deviations were observed for the relaxed scheme while negative mean deviations were observed for the strict scheme. The strict scheme yielded lower absolute values of deviation means but higher deviation variances than the relaxed scheme. These observations can be explained by dependencies between training and validation sets,

relative training set sizes, and the properties of the AUC calculation methods used in the experiments.

The results suggest that the generalization performance of a model is better estimated by the strict scheme than the relaxed scheme in the situation where predictions will be made on pairs of previously unseen objects. Such pairs may be encountered in significant numbers, for example, when predicting protein–protein binding [13]. However, further experiments are needed to get a better understanding of the properties of the strict scheme.

Acknowledgements. This study was supported by the Academy of Finland. We would like to thank the anonymous reviewers for their helpful comments.

References

1. Airola, A., Pahikkala, T., Waegeman, W., De Baets, B., Salakoski, T.: An experimental comparison of cross-validation techniques for estimating the area under the roc curve. Computational Statistics and Data Analysis 55, 1828–1844 (2011)
2. Arlot, S., Celisse, A.: A survey of cross-validation procedures for model selection. Statistics Surveys 4, 40–79 (2010)
3. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. 25, 25–29 (2000)
4. Bork, P., Dandekar, T., Diaz-Lazcoz, Y., Eisenhaber, F., Huynen, M., Yuan, Y.: Predicting function: from genes to genomes and back. J. Mol. Biol. 283, 707–725 (1998)
5. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30, 1145–1159 (1997)
6. Braga-Neto, U.M., Dougherty, E.R.: Is cross-validation valid for small-sample microarray classification? Bioinformatics 20, 374–380 (2004)
7. Eisenberg, D., Marcotte, E.M., Xenarios, I., Yeates, T.O.: Protein function in the post-genomic era. Nature 405, 823–826 (2000)
8. Han, L., Cui, J., Lin, H., Ji, Z., Cao, Z., Li, Y., Chen, Y.: Recent progresses in the application of machine learning approach for predicting protein functional class independent of sequence similarity. Proteomics 6, 4023–4037 (2006)
9. Lee, D., Redfern, O., Orengo, C.: Predicting protein function from sequence and structure. Nat. Rev. Mol. Cell Biol. 8, 995–1005 (2007)
10. Mei, S., Fei, W.: Amino acid classification based spectrum kernel fusion for protein subnuclear localization. BMC Bioinformatics 11(suppl. 1), S17 (2010)
11. Nadeau, C., Bengio, Y.: Inference for the generalization error. Machine Learning 52, 239–281 (2003)
12. Pahikkala, T., Suominen, H., Boberg, J.: Efficient cross-validation for kernelized least-squares regression with sparse basis expansions. Machine Learning 87, 381–407 (2012)
13. Park, Y., Marcotte, E.M.: Flaws in evaluation schemes for pair-input computational predictions. Nat. Methods 9, 1134–1136 (2012)
14. The UniProt Consortium: Activities at the Universal Protein Resource (UniProt). Nucleic Acids Res. 42, D191–D198 (2014)

A Binary Factor Graph Model for Biclustering

Matteo Denitto, Alessandro Farinelli, Giuditta Franco, and Manuele Bicego

University of Verona, Department of Computer Science, Verona, Italy

Abstract. Biclustering, which can be defined as the simultaneous clustering of rows and columns in a data matrix, has received increasing attention in recent years, particularly in the field of Bioinformatics (e.g. for the analysis of microarray data). This paper proposes a novel biclustering approach, which extends the Affinity Propagation [1] clustering algorithm to the biclustering case. In particular, we propose a new exemplar based model, encoded as a binary factor graph, which allows to cluster rows and columns simultaneously. Moreover, we propose a linear formulation of such model to solve the optimization problem using Linear Programming techniques. The proposed approach has been tested by using a well known synthetic microarray benchmark, with encouraging results.

1 Introduction

Unsupervised learning, also known as *clustering*, is an active and historically fecund research area, which offers a wide range of solution techniques [2]. In recent years, the interest of the research community has been focused also on a particular kind of clustering problems, the so-called *biclustering*, also known, in other scenarios, as co-clustering. This term encompasses a large set of techniques generally aimed at "performing simultaneous row-column clustering" [3].

Bi-clustering techniques have been applied in different scenarios, such as document analysis [4], scene categorization [5], and, most importantly, expression microarray data analysis – see the reviews [3,6,7]. In this last scenario, the - starting point is a matrix whose rows and columns represent genes and experiments, respectively. Each entry measures the expression level of a particular gene in a particular experiment. The classical analysis in this scenario is to cluster genes, with the aim of discovering which genes show the same behavior over all the experiments – this permitting the discovery of co-regulation mechanisms. However, a more interesting question can be raised: are there genes that share similar expression *only in a certain subset of experiments*? Addressing this issue, which can not be faced using a standard clustering approach, can provide invaluable information to biologists, and represents the main goal of biclustering approaches.

Different biclustering techniques have been proposed in the past [3,6,7], each one characterized by different features, such as computational complexity, effectiveness, interpretability and optimization criterion . Many of such previous approaches are based on the idea of adapting a given clustering technique to the

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 394–403, 2014.

biclustering problem, for example by repeatedly performing experiments and genes clustering [8,9].

This paper follows the above-described research trend, and proposes a novel biclustering algorithm, which extends and adapts to the biclustering scenario the well known Affinity Propagation (AP) clustering algorithm [1]. This technique, which is based on the idea of iteratively exchanging messages between data points until a proper set of representatives (called exemplars) are found, has shown to be very effective (in terms of clustering accuracy) and efficient (due to its fast learning algorithm) in many different application scenarios, including image analysis, gene detection and document analysis [1]. In Affinity Propagation the clustering problem is formulated as an objective function and a set of constraints; the objective function summarizes the intracluster-similarity and the constraints guide the grouping of the points to a valid solution. Specifically, the objective function and the constraints are encoded as a binary factor graph [10], and the objective function is optimized by using the max-sum message passing algorithm [1,10].

Even if some variants of the AP approach have been applied to the microarray scenario – see for example [11,12] – its use in the biclustering context remains somehow unexplored, with few papers recently published (such as [9], and [13]). In particular, in [9] the AP model is used as the clustering module in an iterative rows and columns clustering scheme [8]: however no modifications to the basic AP model has been introduced, which is still used as a standard clustering method. In contrast, [13] proposes an exemplar-based strategy to find biclusters. However, while such approach shares many similarities with AP (e.g., it is exemplar-based and encodes the problem as a factor graph), a crucial difference is that the proposed factor graph is not binary thus drifting away from the spirit of the original AP scheme, which exploits the binary nature of the factor graph to derive efficient and fast update messages [14].

In this paper we propose an extension of the Affinity Propagation model, which i) is based on a binary factor graph, and ii) directly performs biclustering. In particular we extend the AP model in two ways: i) we consider as datapoints to be analysed the single entries of the input data matrix, instead of the classical row/column vector; ii) we add to the model a constraint which forces points belonging to the same cluster to represent a valid bicluster (namely *all* points of a subset of rows and columns). Given the new factor graph, a possible solution to optimize the objective function is to resort to the max-sum algorithm [1,10]. However, given the high number of cycles present in the factor graph, the max-sum algorithm is likely to produce poor quality solutions [15]. Therefore we derived an alternative linear formulation of the optimization problem, and use Linear Programming techniques to find the optimal solution of our model. Finally, while the space complexity of the model and the time complexity of the algorithm are both polynomial in the number of entries of the data matrix, the number of variables and constraints that our model introduces is very large (i.e., $O(n^2m^2)$ variables and $O(n^3m^3)$ functions for an input matrix with n rows and m columns). Hence, storing our model for typical biclustering matrices

(which can contain hundreds of rows/columns) is an issue. Consequently, we derived an aggregation methodology, which groups results obtained on smaller matrices: this allows the evaluation of the proposed approach on a standard expression microarray benchmark [6]. Obtained results confirm the potentials of the proposed method.

The remainder of paper is organized as follows: Sect. 2 presents Affinity Propagation, the starting point of our model; the proposed approach is then described in Sect. 3 and Sect. 4, whereas the experimental evaluation is given in Sect. 5; finally Sect. 6 concludes the paper.

2 Affinity Propagation

Affinity Propagation (AP) is a well known clustering technique recently proposed by Frey and Dueck [1]. The efficacy of this algorithm (in terms of clustering accuracy) and efficiency (due to the fast resolution) have been shown in many different clustering contexts [1].

The main idea behind AP is to perform clustering by finding a set of *exemplar points* that best represent the whole data set. This is obtained by representing the input data as a factor graph [16]: a bipartite graph that encodes an objective function as an aggregation (e.g., a sum) of functions (typically called factors). In the graph, the nodes (circles) define the data points and the factors (squares) are functions defined over a subset of nodes – for details please refer to [10]. The objective function is then optimized by running an iterative message passing approach, which, in the typical task of maximizing a sum of functions, is the max-sum algorithm [10].

In particular, in Affinity Propagation the factor graph is composed by two parts: the first encodes the choice of the points and their exemplars via a binary matrix C, where an entry $C(i,j) = c_{i,j}$ is set to one if the point i chooses j as exemplar. This choice is ruled by the pairwise similarity values $s_{i,j}$, which define the similarity between each pair of points i and j. The values $s_{i,i}$, given as an input, represent the *preference* for point i of being itself an exemplar: such choice influences the final number of clusters, which is automatically found by the algorithm. The second part of the factor graph define two constraints, which ensure to retrieve only valid solutions:

1. *1-of-N constraint*: every point has to chose one, and only one, exemplar. This can be represented by a function I over n nodes:

$$I_i = \begin{cases} 0, & \text{if } \sum_{i=1}^{n} c_{i,j} = 1 \\ -\infty, & otherwise \end{cases} \tag{1}$$

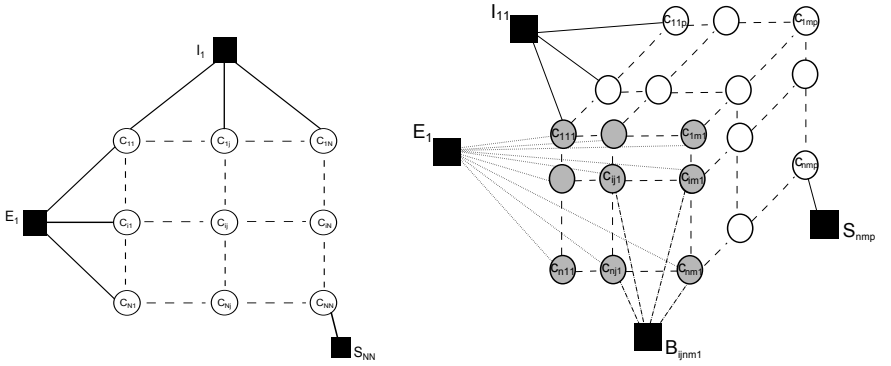
   where n is the number of the points;
2. *Exemplar consistency constraint*: if a point is chosen as an exemplar by some other data point, it must choose itself as an exemplar. This constraint avoids

circular choices ("a" chooses "b", "b" chooses "c", "c" chooses "a") and can be represented by a function E over n nodes:

$$E_j = \begin{cases} -\infty, & \text{if} \quad c_{jj} = 0 \quad \text{and} \quad \sum_{i=1}^{n} c_{i,j} \geq 1 \\ 0, & otherwise \end{cases} \tag{2}$$

where n is the number of data points.

Note that we have as many I and E functions as the number of data points in input. Figure 1(a) reports the factor graph used in AP.



(a) (b)

Fig. 1. Factor Graph for Affinity Propagation (a) and the proposed Factor Graph for Biclustering (b)

The objective function expressed by the AP factor graph is the sum of all the factors, i.e., the constraints expressed in Equations (1) and (2) and the sum of all similarity functions $S(i,j)$ which are defined as the similarity value $s_{i,j}$ multiplied by the variables $c_{i,j}$.

$$F = \sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij} \cdot c_{ij} + \sum_{i=1}^{n} I_i + \sum_{i=1}^{n} E_j \tag{3}$$

3 The Proposed Approach

In this section the proposed approach is presented. In general terms, given a data matrix $D = (d_{ij})_{i \in N, j \in M}$, with N set of rows ($|N| = n$) and M set of columns ($|M| = m$), a bicluster $B = (d_{ij})_{i \in T, j \in K}$ is a submatrix of D, for $T \subseteq N$ and $K \subseteq M$, which meets specific spatial constraints ruled by a certain similarity criterion. Here we assume that different biclusters do not overlap[1].

[1] i.e. each element of the data matrix must belong to a unique bicluster.

In our approach, instead of considering as basic elements the rows and the columns, we directly consider the single entries of the input data matrix. Starting from $\{d_{ij}\}_{i\in N, j\in M}$, we look for biclusters as sets of "coherent" entries of the matrix respecting the specific spatial constraint. To obtain this, we re-define the factor graph of Affinity Propagation: in particular, we have one variable for each pair of entries of the data matrix D to encode the exemplar choice; moreover, we introduce a constraint to ensure that points that belong to the same cluster represent a bicluster. In what follows, we define our model, specifying the variables, the constraints and the objective function, and motivate the use of an LP optimization approach.

3.1 The Model

Variables. Our goal is to cluster the single entries of the data matrix: therefore we encode the exemplar chosen by each entry of the data matrix D with a four-dimensional Boolean matrix, where an entry $C(i, j, t, k) = c_{ijtk}$ is 1 if the point in position (i, j) of the matrix chooses (t, k) as its exemplar. For reasons which will be clearer later, we replace the indices of the second point with a single value $(z = 1, 2, 3, \cdots, n \cdot m)$ obtaining a three-dimensional structure $C(i, j, z)$; again, a variable c_{ijz} is set to 1 if the point (i, j) chooses the point z as its exemplar. As in Affinity Propagation, this choice is based on a certain similarity matrix S, which now encodes the similarities between every pair of entries (i, j) and (t, k) of the input data matrix. As for C, we rearrange this four-dimensional matrix in a three dimensional one $S(i, j, z)$.

Functions. Following Affinity Propagation, we include in our model the constraint I_{ij} (which is similar to (1) and encodes that one data entry should choose only one exemplar) and E_z (which is similar to (2) and encodes that if $c_{i,j,z} = 1$ then $c_{\hat{i},\hat{j},z} = 1$, where $\hat{i}$ and $\hat{j}$ are the indices that correspond to z), which guarantee valid variable assignments. Next, we introduce an extra constraint, which ensures that the entries of the matrix which are in the same cluster do represent a bicluster. In this perspective, we observe that, given a certain value z, the bidimensional matrix

$$C(:,:,z) = \begin{bmatrix} c_{11z} & c_{12z} & \cdots & c_{1mz} \\ c_{21z} & c_{22z} & \cdots & c_{2mz} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1z} & c_{n2z} & \cdots & c_{nmz} \end{bmatrix} \quad \text{with } 1 \leq z \leq n \cdot m \tag{4}$$

immediately summarizes the relation between all the entries of the matrix and the entry z: in particular, $c_{ijz} = 1$ indicates that (i, j) has chosen z as its exemplar. Now, the constraints I_{ij} and E_z ensure that all the points in a given cluster had chosen the same exemplar, hence every matrix $C(:,:,z)$ represents a potential bicluster. However, to be a valid bicluster, such matrix should fulfil one of the two following conditions:

1. (trivial constraint) it should contain all zeros: there are no points choosing as exemplar the point z;
2. (bicluster integrity constraint) the coordinates of the entries with 1 (namely the coordinates of the entries in the bicluster) should represent *all* the points of a given subset of rows and columns: in simple words, after rows-columns re-arrangements, the ones in the $C(:,:,z)$ matrix should form a full rectangle (a rectangle with no zero elements).

This can be ensured by defining a constraint for every 4 points of the matrix $C(:,:,z)$: if c_{ijz} and c_{tkz} are set to 1, then also c_{ikz} and c_{tjz} should be set to 1. More formally, the *bicluster integrity constraint* is defined as:

$$B_{ijtkz} = \begin{cases} -\infty, & \text{if} \quad c_{ijz} = 1, c_{tkz} = 1 \quad \text{and} \quad c_{ikz} \cdot c_{tjz} = 0 \\ 0, & otherwise \end{cases} \tag{5}$$

Notice that the function B is defined, for every sheet z, on all the possible pairs of points (i,j) and (t,k).

Objective Function. Given the variables and the constraints above described – represented in Fig. 1(b) – we can now write the objective function, defined by the sum of the intra-biclusters similarity (via the matrix C and S) and the constraints (I, E, and B):

$$F = \sum_{i,j,z} c_{ijz} \cdot s_{ijz} + \sum_{i,j} I_{ij} + \sum_{z} E_z + \sum_{z,i,j,t,k} B_{ijtkz} \tag{6}$$

where: $1 \leq i \leq n, 1 \leq j \leq m, 1 \leq z \leq n \cdot m, 1 \leq t \leq n$ and $1 \leq k \leq m$.

3.2 Optimization of the Objective Function

Now, there are many possible approaches to maximize the objective function expressed by the factor graph in Fig. 1(b). In AP the binary nature of the nodes in graph is exploited to calculate an approximation of the maximum through the max-sum algorithm [1]. However, the biclustering integrity constraint (defined over every pair of entries of the matrix) induces a high number of cycles in the graph, and it is well known that the performances of the approximated maximization algorithms degrade in such conditions [15]. Therefore we follow an alternative route, giving a linear formulation of the objective function, and using linear programming (LP) techniques [17] to find the optimal variable assignment. In general, LP approaches maximize/minimize an objective function where the constraints defined on the data points are all linear [17]. In the objective function (6), the first three addends can be easily written in a linear form; in the following we will show how to transform the biclustering integrity constraint (5) into a linear set of constraints.

The idea is that, when considering the matrix $C(:,:,z)$, the biclustering integrity constraint is satisfied if, and only if, all rows (or columns) of this matrix

are either zero or equal to each other. By exploiting the Boolean nature of the variables, this can be enforced by checking if, for every pair of rows (or columns) $U = (u_1, \ldots, u_m)$ and $X = (x_1, \ldots, x_m)$, one of the following conditions is true: i) $U = X$, ii) U = 0, iii) X = 0. This can be expressed through Boolean algebra as: i) NOT $(\sum_i (u_i \oplus x_i))$, ii) NOT $(\sum_i u_i)$, NOT $(\sum_i x_i)$, where "+" denotes the OR operator and "$\oplus$" is the XOR operator. By using De Morgan laws and some proprieties of the Boolean algebra we can derive the set of linear constraints representing the OR operation between the previous i), ii) and iii) constraints as:

$$\begin{array}{rrcr} -u_1 + x_1 + u_2 < 2 & -u_1 + x_1 + u_3 < 2 & \cdots & -u_1 + x_1 + u_n < 2 \\ u_1 - x_1 + x_3 < 2 & -u_2 + x_2 + u_1 < 2 & \cdots & u_1 - x_1 + x_n < 2 \end{array}$$

this has to be done for all the pairs of rows (or columns) of every matrix $C(:, :, z)$.

Now, all the elements of the model (objective function and constraints) are linear, and the model can be solved by using LP approaches.

Let us analyse the complexity of the proposed approach. Given an input matrix formed by n rows and m columns, the model contains $O(n^2m^2)$ variables and $O(nm)$ functions for the constraints I and E. Unfortunately, when considering the *biclustering integrity constraint*, the number of functions to completely describe all possibilities raises to $O(m^3n^3)$. Even if being still polynomial (and not exponential) in the number of rows and columns of the data matrix, the number of functions to store in memory can be very large. In particular, for typical biclustering problems (e.g., microarray analysis), the data matrix can contain hundreds of rows and columns, hence our approach might require a prohibitive amount of memory to store the model. About time complexity, an Integer Programming problem is exponential in the number of constraints (in the worst case). Anyway, there are many well established methods which provide, on average, time satisfactory solutions. To overcome the scalability issue we run our algorithm on smaller matrices, extract biclusters and devise an aggregation algorithm to find biclusters in the original data matrix. We describe such aggregation algorithm in next Section.

4 Aggregation of Biclusters

Let a kernel be a window glass selecting a sub-matrix, we start by analyzing the data matrix by means of a fixed dimension kernel, which is shifted along the matrix, with no overlap. For every kernel, the optimal solution is retrieved (using our model and the LP approach). Once the whole matrix has been analyzed, the set of biclusters is then processed in three steps:

1. we apply a clustering algorithm on the exemplars retrieved in the different kernels, to partition the set of biclusters in groups of biclusters with coherent values. Here we adopt as a clustering algorithm the original Affinity Propagation method.

2. for every group of biclusters, we perform a hierarchical agglomerative grouping which, starting from single biclusters, repeatedly joins together the most similar groups of biclusters. Similarity between two groups of biclusters is defined as the number of rows and columns that they share – when the similarity of the nearest group is zero (no overlap) the algorithm stops. In other words, we perform a classical agglomerative clustering of biclusters by using as similarity the degree of column/rows overlap. Every group in the final partition now represents a set of biclusters with no row/column overlap with the other groups.
3. we post-process the final groups in order to be sure that they represent an actual bicluster: this is done by removing rows (or columns) which violate the bicluster definition.

Notice that, the third step is necessary because merging biclusters may not produce a bicluster as result. A possible alternative would be to merge only pairs of biclusters that result in a bicluster, however by so doing we would not obtain large biclusters given by the simultaneous merge of k biclusters (where $k > 2$). Having described our approach, we now turn to the empirical evaluation.

5 Results

The methodology proposed in this paper has been tested on a set of synthetic matrices which represent a classical benchmark in the microarray scenario [6]: such set comprises synthetic expression matrices, perturbed with different schemes[2]. In the experiments, we have 10 non-overlapping biclusters, each extending over 10 rows and 5 columns. Such datasets have been widely used to investigate the effects of noise on the performance of various biclustering approaches. The accuracy of the biclustering has been assessed with the so-called *Gene Match Score* [6]: the average bicluster *relevance* reflects to what extent the generated biclusters represent a true bicluster in gene dimensions, and the average bicluster *recovery* quanties how well each of the true biclusters is recovered by the biclustering algorithm (such scores vary between 0 and 1, where the higher the better the accuracy).

In our model we used as similarity the negative of the Euclidean distance (as in [9]), which allows to retrieve only constant value biclusters. As in the original Affinity Propagation model, a proper setting of the preferences (namely the self similarities) is crucial: in our experiments we found that a good choice is represented by the first integer number below the median (which represents the standard setting [1]). The Linear Programming model was implemented and resolved using CPLEX (version 12.4).

Figures 2(a) and 2(b) report the Gene Match Scores (the recovery and the relevance values respectively – see [6]) for different levels of noise and for different dimensions of the kernel, averaged over the different repetitions (also standard deviations are displayed). As expected the approach provides better solutions as

[2] All datasets may be downloaded from: `www.tik.ee.ethz.ch/sop/bimax`

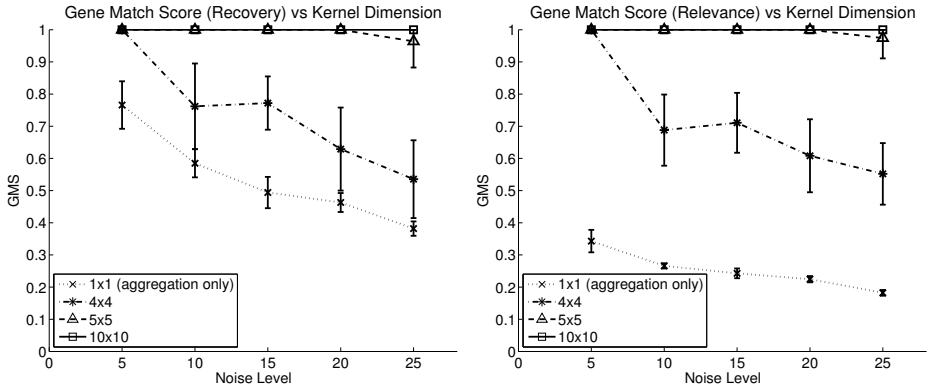


Fig. 2. Results for the proposed approach: (a) recovery and (b) relevance – for further information we refer to [6]

the kernel dimension increases. Please note that when using the [1x1] kernel only the aggregation algorithm described in Section 4 is employed (every data point is in its own bicluster). As we can see in Fig.2, increasing the noise completely corrupts the performances of the aggregation algorithm. Notice that, obtained results are competitive with other state of the art approaches (see figure 2 in [6], figure 1 in [18] or figure 3 in [9]), confirming the potentialities of the proposed approach.

6 Conclusions

In this paper we propose a novel model, inspired by Affinity Propagation [1], to retrieve biclusters from a data matrix. A key innovative element of our approach is to analyze directly the entries of the data matrix, instead of considering whole rows and columns, and to use Linear Programming techniques for computing the optimal solution [17]. The space/time complexity of the model does not allow to run our approach on typical biclustering problems, hence we partition the original data matrix in small kernels and analyse each such kernel with our approach. We then propose an aggregation approach to reconstruct the original biclusters. We evaluate our approach on standard benchmarking datasets for biclustering [6], and results show that the method is competitive with respect to other state of the art approaches.

Future work in this area includes two main research directions: first, investigate possible extensions of the approach to reduce the complexity of the data representation model, second to test the approach on real biological data sets, hence assessing the practical significance of the approach.

References

1. Frey, B., Dueck, D.: Clustering by passing messages between data points. Science 315, 972–976 (2007)
2. Jain, A., Murty, M., Flynn, P.: Data clustering: a review. ACM Computing Surveys 21, 264–323 (1999)
3. Madeira, S., Oliveira, A.: Biclustering algorithms for biological data analysis: a survey. IEEE transactions on Computational Biology and Bioinformatics 1, 24–44 (2004)
4. Dhillon, I.: Coclustering documents and words using bipartite spectral graph partitioning. In: Proc. Int. Conf. on Knowledge Discovery and Data Mining, pp. 269–274 (2001)
5. Irie, G., Liu, D., Li, Z., Chang, S.F.: A bayesian approach to multimodal visual dictionary learning. In: Proc. Int. Conf on Computer Vision and Pattern Recognition, pp. 329–336 (2013)
6. Prelic, A., Bleuler, S., Zimmermann, P., Wille, A., Bhlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: Comparison of biclustering methods: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22(9), 1122–1129 (2006)
7. Flores, J.L., Inza, I., Larrañaga, P., Calvo, B.: A new measure for gene expression biclustering based on non-parametric correlation. Computer Methods and Programs in Biomedicine 112(3), 367–397 (2013)
8. Getz, G., Levine, E., Domany, E.: Coupled two-way clustering analysis of gene microarray data. Proc. Natl. Acad. Sci. U S A 97(22), 12079–12084 (2000)
9. Farinelli, A., Denitto, M., Bicego, M.: Biclustering of expression microarray data using affinity propagation. In: Loog, M., Wessels, L., Reinders, M.J.T., de Ridder, D. (eds.) PRIB 2011. LNCS, vol. 7036, pp. 13–24. Springer, Heidelberg (2011)
10. Bishop, C.: Pattern Recognition and Machine Learning. Springer-Verlag New York, Inc. (2006)
11. Bayá, A., Granitto, P.: Clustering gene expression data with a penalized graph-based metric. BMC Bioinformatics 12 (2011)
12. Kiddle, S., Windram, O., McHattie, S., Mead, A., Beynon, J., Buchanan-Wollaston, V., Denby, K., Mukherjee, S.: Temporal clustering by affinity propagation reveals transcriptional modules in arabidopsis thaliana. Bioinformatics 26(3), 355–362 (2010)
13. Tu, K., Ouyang, X., Han, D., Honavar, V.: Exemplar-based robust coherent biclustering. In: SDM, pp. 884–895. SIAM (2011)
14. Givoni, I., Frey, B.: A binary variable model for affinity propagation. Neural Computation 21(6), 1589–1600 (2009)
15. Weiss, Y., Freeman, W.: Correctness of belief propagation in gaussian graphical models of arbitrary topology. Neural Computation 13(10), 2173–2200 (2001); cited By (since 1996) 168
16. Kschischang, F., Frey, B., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory 47(2), 498–519 (2001)
17. Dantzig, G.: Linear Programming and Extensions. Princeton University Press (August 1963)
18. Bicego, M., Lovato, P., Ferrarini, A., Delledonne, M.: Biclustering of expression microarray data with topic models. In: Int. Conf. on Pattern Recognition (ICPR 2010), pp. 2728–2731 (2010)

Improved BLSTM Neural Networks for Recognition of On-Line Bangla Complex Words

Volkmar Frinken[1], Nilanjana Bhattacharya[2], Seiichi Uchida[1], and Umapada Pal[3]

[1] Department of Advanced Information Technology
Kyushu University, Fukuoka, Japan
{vfrinken,uchida}@ait.kyushu-u.ac.jp
[2] Bose Institute
Kolkata, India
nilibht@gmail.com
[3] Computer Vision and Pattern Recognition Unit
Indian Statistical Institute, Kolkata, India
umapada@isical.ac.in

Abstract. While bi-directional long short-term (BLSTM) neural network have been demonstrated to perform very well for English or Arabic, the huge number of different output classes (characters) encountered in many Asian fonts, poses a severe challenge. In this work we investigate different encoding schemes of Bangla compound characters and compare the recognition accuracies. We propose to model complex characters not as unique symbols, which are represented by individual nodes in the output layer. Instead, we exploit the property of long-distance-dependent classification in BLSTM neural networks. We classify only basic strokes and use special nodes which react to semantic changes in the writing, i.e., distinguishing inter-character spaces from intra-character spaces. We show that our approach outperforms the common approaches to BLSTM neural network-based handwriting recognition considerably.

Keywords: Handwritten Text, BLSTM NN, Bangla Text, Complex Temporal Pattern.

1 Introduction

The recognition of complex temporal structures, or sequences, is a difficult problem. In particular long-distant dependencies are hard to model. A common example of sequences with long-distance dependencies is the pen trajectory of handwritten text. In the shape of the digit '0', e.g., the position of the end of the stroke depends upon the position at the beginning of the stroke.

While such long-distant dependencies pose severe problems for hidden Markov models, which explicitly makes use of the Markov-property, recently introduced recurrent neural networks, called long short-term memory (LSTM) neural networks are designed specifically for this task. For writing systems with a

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 404–413, 2014.

limited number of different output classes, like Latin[1] or Arabic, BLSTM neural networks are currently among the best performing recognition approaches for handwritten text.

However, this is not the case for highly complex scripts, such as Japanese or Chinese. The large number of different output classes pose a severe problem in the design of the networks, where normally each possible character is assigned to a different node in the network output layer. To the knowledge of the authors, no publication exists that shows how BLSTM neural network, which performs very well for English and Arabic texts, can deal with languages containing hundreds or thousands of characters.

In this regard, Bangla (the writing system of the Bengali language) shares several similarities to the Latin or Arabic writing system, but also to Chinese or Japanese. Bangla words are sequences of characters, each of which is either a basic character (similar to English) or a compound character, which is composed of several basic strokes, similar to radicals in a Chinese character.

This is the first work, to the knowledge of the authors, that explores the applicability of BLSTM neural network to Bangla words containing compound character. Therefore it constitutes an intermediate step to apply such networks to highly complex temporal pattern containing long-term dependencies. The main contribution of this paper is therefore to demonstrate how BLSTM neural network can deal with complex patterns of a language which has a large number of distinct characters.

The rest of the paper is structured as follows. Section 2 reviews relevant literature. The particularities of the Bangla writing Systems are introduced in Section 3. An explanation of the BLSTM neural networks is given in Section 4. The different approaches to Bangla compound character recognition are presented in Section 5. Section 6 provides an experimental evaluation and conclusions are drawn in Section 7.

2 Related Work

The most successful approaches to classify temporal pattern involve hidden Markov Models [21] or recurrent neural networks [8]. Long-term dependencies within sequences have been successfully addressed using BLSTM neural networks for handwriting recognition of Latin an Arabic scripts [11,9], speech recognition [12], or abstract sequences [10].

The recognition of on-line handwritten text is an active field of research [20], including on-line Chinese [13] and Japanese [17] handwriting recognition.

Some works are available on on-line isolated Bangla character/numeral recognition in [7,22,19,2,16]. In [3], handwritten words were segmented estimating the position of headline of the word. Preprocessing operations such as smoothing and re-sampling of points were done before feature extraction. They used 77 features considering 9 chain-code directions. Modified quadratic discriminant function (MQDF) classifier was used for recognition. In [4], the authors divided each

[1] Used for English, Spanish, German, etc.

stroke of the preprocessed word sample into several sub-strokes using the angle incurred while writing. Feature values representing its shape, size and relative position are computed. Then HMM was used for recognition. A system for segmentation and recognition of Bangla on-line handwritten text containing both basic and compound characters is described in [1]. At first, cursive words are segmented into primitives. Next primitives are recognized. Directional features were used in SVM for recognition.

In [18], Bangla character are recognized with hidden Markov models based on manually grouping complete strokes of ideal character shapes. As opposed to their work, we do not rely on a manual stroke grouping nor any information about ideal character shapes.

3 Bangla Writing System

Bangla is the second most popular language in India and the fifth most popular language in the world. More than 200 million people speak in Bangla and this script is used in Assamese and Manipuri languages in addition to Bangla language. Handwriting recognition of unconstrained Bangla text is difficult because of the presence of many complex shaped characters as well as variability involved in the writing style of different individuals. Writing two or more characters by a single stroke (a stroke is a collection of points from pen down to pen up) is another difficulty for on-line Bangla text recognition. The main difficulty of any character recognition system is the shape similarity. It can be noted that because of handwritten style, two different characters in Bangla may look very similar.

The Bangla writing system is made up of basic charactersas well as more than 300 compound characters, each of which consists of up to four basic shapes. Often, the shapes of the basic characters are preserved in the compound characters, but sometimes the basic characters take a new shape. An example of a Bangla word containing a compound character is given in Fig. 1.

With these property, the Bangla writing system shares several similarities to other eastern writing systems, in particular Chinese and Japanese ones, yet to a less complex extend. At the same time it shares some similarities to English or Arabic with its cursive writing style and the use of basic characters. This makes the Bangla writing system a well-suited testing ground for adapting BLSTM neural networks to Asian handwriting recognition in general.

4 BLSTM Neural Networks

The recognizer used in this paper is a BLSTM neural network, which is a recently developed recurrent neural network [11]. A hidden layer is made up of so called *long short-term memory* blocks instead of simple nodes. These memory blocks are specifically designed to address the *vanishing gradient problem* which describes the exponential decay of influence of sequence observations as a function of the distance within the sequence.

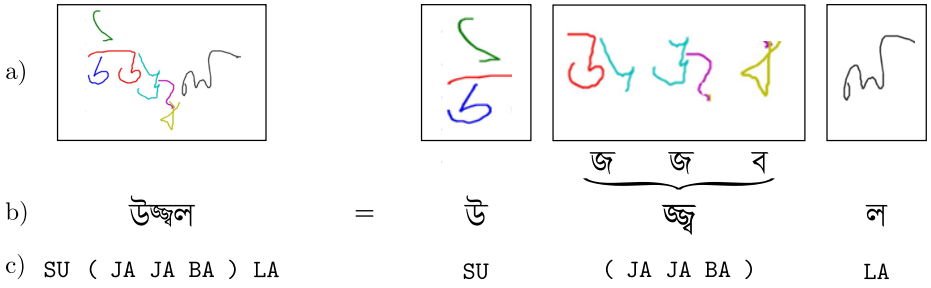


Fig. 1. A handwritten Bangla word, consisting of three characters. The first and last character consists of a basic character, each, encoded as SU and LA, respectively. The second character is a compound character composed out of three basic characters, which are encoded as JA, JA, and BA. In a) the decomposition of the strokes is given, in b) the printed Bangla text of the strokes and in c) the transcription according to our encoding of the symbols.

The network is made up of two separate input layers, two separate recurrent hidden layers, and one output layer. Each input layer is connected to one hidden layer. The hidden layers are connected to the output layer. The network is *bidirectional*, i.e. a sequence is fed into the network in both the forward and the backward mode. The input layers consist of one node for each feature. One input and one hidden layer deal with the forward sequence, the other input and hidden layer with the backward sequence. At each position p of the input sequence of length l, the output layer sums up the values coming from the hidden layer that has processed positions 1 to p and the hidden layer that has processed the positions l down to p. The output layer contains one node for each possible character in the sequence plus a special ε node, to indicate "no character". At each position, the output activations of the nodes are normalized so that they sum up to 1, and are treated as probabilities that the node's corresponding character can occur at this position. The output of the network is therefore a matrix of probabilities for each letter and each position. A visual representation of a BLSTM neural network is given in Fig. 2.

To arrive at a final recognition, the most likely path through the output probability sequence is sought after. The probability of a path through the output sequence $p(c_{k_1} c_{k_2} \dots c_{k_n}|O)$ is given by multiplying the individual probabilities $\prod_i y_{k_i}(i)$. To recognize class sequences that might be shorter than the input sequence, a given path can be shortened using operator $\mathcal{B}$. The operator first deletes consecutive occurrences of the same class (a) and then all ε entries (b):

$$\mathcal{B}(c_1, c_2, c_2, c_2, \varepsilon, c_2) \overset{(a)}{=} \mathcal{B}(\varepsilon, c_1, c_2, \varepsilon, \varepsilon, c_2) \overset{(b)}{=} c_1 c_2 c_2 \ .$$

For lexicon-based word recognition, all words from a dictionary V are matched to the output layer via dynamic programming, by implementing the operator $\mathcal{B}$. First the word $w = c^1 c^2 \dots c^m$ is written as $\hat{w} = \varepsilon c^1 \varepsilon c^2 \varepsilon \dots \varepsilon c^m \varepsilon$. With this, a set of character transitions rules ensures that the sequence of nodes from the

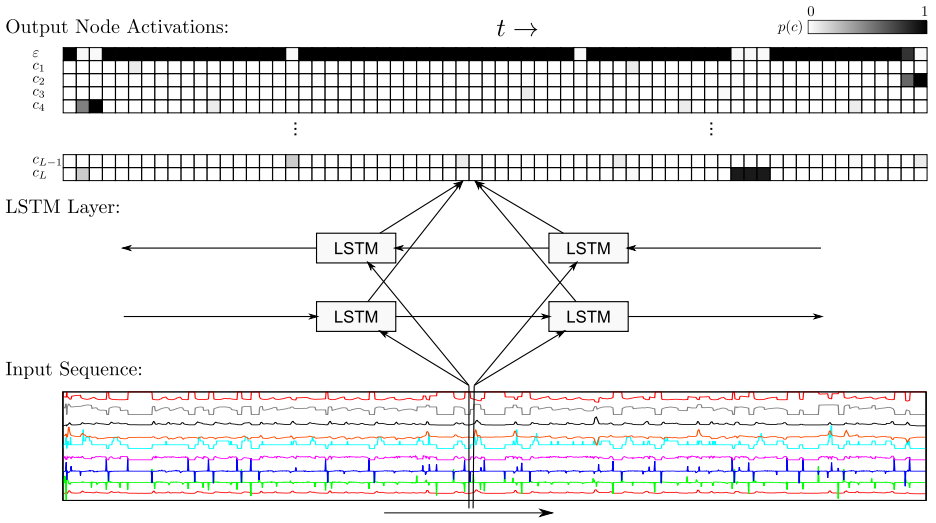


Fig. 2. The matrix of output activations returned by the neural network. The darker the color of the cell, the higher the corresponding output activation.

output path is in $\mathcal{B}^{-1}(w)$. The matching score is the product of all corresponding output activations. The word leading to the highest matching score is chosen to be the recognized word. For a detailed explanation of the recognition algorithm, the reader is referred to [11].

5 Compound Character Detection and Recognition

The target problem addressed by the research presented in this paper is the recognition of complex compound characters alongside simple basic characters. In order to do that, several different strategies (configurations) for encoding the characters have been explored.

In the first configuration, each characters, compound or basic, is mapped to a dedicated output node. This is the basic approach and resembles the known strategy employed in all published works of BLSTM NN-based handwriting recognition, known to the authors. In this paper, this configuration serves as a reference performance.

In all other configurations, compound characters do not have a dedicated output node, but are encoded in various different ways.

In the second configuration, the output layer in endowed with a special *compound connection* node '`+`'. The network is trained to activate this node between activations of the basic node to indicate a compound character. Consequently, a string such as "`SU JA + JA + BA LA`" encodes a compound character composed out of the 3 basic shapes '`JA`', '`JA`', and '`BA`' between the simple characters '`SU`' and '`LA`'.

Table 1. The 5 different configurations, the number $|OL|$ of nodes in the output layer OL, and as an example the encoding of a Bangla word containing the characters "SU JA·JA·BA LA", where JA·JA·BA is a compound character made from the basic characters JA (2 times) and BA

Configuration	$\|OL\|$	Encoding of "SU JA·JA·BA LA"
1	171	SU JAJABA LA
2	73	SU JA + JA + BA LA
3	73	SU ◇ JA JA BA ◇ LA
4	74	SU (JA JA BA) LA
5	72	SU JA JA BA LA

In the third configuration, the network is trained to indicate the end of each character with the node ◇. Compund characters are trained by returning the sequence of basic characters without a ◇ node activation. The same word as above is therefore encoded as "SU ◇ JA JA BA ◇ LA".

In the forth configuration, compound characters are with a starting and ending symbol '(' and ')'. As a main difference to the third configuation, these symbols do not occur between basic characters.

For the fifth configuration, no special characters are used in the output layer. The network is trained to simply return the sequence of basic characters and a post-processing step using a language model is needed to re-create the original word. An overview of the 5 different configurations can be seen in Tab. 1.

Finally, to fully exploit the diversity in the data representation, we combine the network outputs. We tested three different combination methods. For each combination method, we chose 2 networks from each configuration, hence 10 different networks. First, the network output is transformed into pairs of words and posterior probabilities. This is done by summing up the matching score returned from the dynamic programming in the the lexicon-based word recognition (see Section 4) for all possible words in the dictionary and dividing the matching score by that sum.

The first combination uses the *Max* combination rule [14]. It returns the word having the highest posterior probability. Similarly, in the second combination experiment, we implement the *Average* combination rule, which returns the word with the highest average posterior probability. The last combination experiments uses and *Exponentiated Borda count*. For combining recognizers with *Borda Count* [15], each recognizer's n-best list is used and the word at position i on the list is given the score $n - i + 1$. For each word in the dictionary, the score is summed up over each recognizer's list and the word with the highest score is returned. In contrast, *Exponentiated Borda Count* makes use of the score $(n - i + 1)^p$, with p being a free parameter and has been shown to outperform normal Borda Count [6].

6 Experimental Evaluation

A total set of 1 552 Bangla words were collected using an I-ball A4 Takenote tablet from 40 different writers, each contributing at least 30 words. Writers were

requested to write words from a given a lexicon of 149 words. Input data consist of (x, y) coordinates along the trajectory of the pen together with positions of pen-downs (stroke starting points). The sampling rate of the signal is considered fixed for all the samples. We have divided the input samples into a training and a testing set. Each set contains 149 directories, one for each of the input words. Training and testing contains distinct sets of inputs, i.e. a writer who contributed to one of the two sets did not contribute any words to the other set. Ground truths (transcriptions) are written in corresponding text file in each of 149 directories in both training and testing set[2].

To be processed by the BLSTM NN, each word is represented as a sequence of vectors, one for each sample point of the recorded pen trajectory. A vector $(d(s, p_1), d(s, p_2), \ldots, d(s, p_k))^{\mathrm{T}}$ consists of distances $d(\cdot, \cdot)$ between the local context s (the sub-stroke of length l ending in that point) and a set of k prototypes $\{p_1, p_2, \ldots, \}$, which are sub-strokes randomly selected from the training data. This form of dissimilarity space embedding feature description has been shown to work well [5]. In our implementation, the distance function d is the sum of the Euclidean distances between the individual points. We chose to consider $k = 12$ prototypes and a sub-stroke length of $s = 12$.

6.1 Setup

The initialization of the BLSTM neural networks is done by assigning random weights to the connections. Therefore, we trained 9 the BLSTM neural networks for each of the different output layer configurations and report the best results of all networks.

The number of LSTM nodes was set to 100, the learning rate was set to 10^{-4}, and the momentum to 0.9. These parameters have been optimized of and verified on different databases and shown to perform well. Due to the lack of a validation set, training was stopped after 600 training epochs.

The word recognition accuracy of the different configurations are: 17.21% for Configuration 1 (the reference system), 36.47% for Configuration 2, 48.38% for Configuration 3, 32.10% for Configuration 4, and 44.81% for Configuration 5. Hence, all alternative encodings for complex compound characters outperform the reference system substantially.

The reason for the first configuration, which is the common approach to BLSTM NN-based sequence recognition, to perform so poorly is very likely the large number of nodes in the output layer combined with a small training set, which is not sufficient to robustly train such a large network, leading to such a high confusion rate.

In Configuration 3 the training target for the extra node is to recognize inter-character transitions. This seems to be an easier task than the recognition of the opposite as done in Configuration 2. There, the extra node is trained to be activated only if a new basic stroke occurs, but not in a new character.

[2] The database is available upon request.

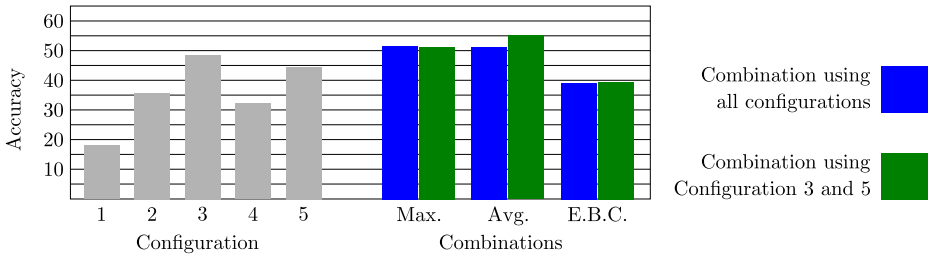


Fig. 3. The recognition accuracy using the different combination methods Maximum (Max.), Average (Avg.), and Exponentiated Borda Count (E.B.C.).

Configuration 4 has two extra nodes, both of which should only be activated in the surrounding of a compound character. Furthermore, as opposed to Configuration 3, these symbols are not trained to occur between basic characters. Hence the number of occurrences in the training set is much less, which might lead to a less robust recognition. This task seems to challenging as well and leads to the second worst recognition rate.

The fifth configuration achieves the second best recognition rate by training the network to recognize only the basic shapes, while the disambiguation is left to the lexicon-based word-recognition.

From these results we can see that the traditional approach of associating each output symbol with its own node in the output layer is not a suitable approach for recognition tasks with a large number of output classes. Instead, the recognition of composing strokes, combined with special output activations to guide the final recognition, seems more promising.

6.2 Combination Experiments

The final recognition results after combining the recognition systems are shown in Fig. 3 for combining the best two nets of all different configurations as well as combining only the best two nets of Configuration 3 and Configuration 5. The highest performance can be achieved using the Average combination rule and a combination of the Configurations 3 and 5 and it has a recognition accuracy of 55.05%. Nevertheless, the Average and the Maximum combination rules perform similarly well, much better than not only the baseline system but each single individual configuration. Exponentiated Borda Count does not perform well, which clearly underlines the importance of the recognition posterior probability which is not used in this combination method. The free parameter p was set to 1.2 which gave good results according to previous experiements. Normal Borda Count performed even worse with less than 20% recognition rate (not shown). The oracle combination, which returns the true class if it occurs in at least one of the recognizers, achieves a recognition accuracy of 70.77% when combining Configuation 3 and 5, and 79.10% when combining all 5 configuations, clearly showing that there is room left for improvement.

7 Conclusion

We have shown in this work how complex temporal pattern can be recognized with BLSTM NN. In the standard approach, each output class is represented by an individual node.

For languages with a large number of characters, unlike English and Arabic, BLSTM neural networks are still unexplored. The Bangla writing system, with a few hundred symbols, is a straightforward testing ground for such problems. In the proposed approach for on-line Bangla handwriting recognition, complex shapes are not recognized as a single unit, but as a composition of basic shapes. Dedicated output notes are used in addition to the nodes representing the basic shapes to separate compound characters and help in the recognition.

Through experimental evaluation we show that the proposed approach outperforms the common approach used for Latin and Arabic scripts. This is an important step towards Japanese or Chinese on-line handwriting recognition using BLSTM.

In the future, we will continue the research on more complex shapes in a variety of different input sources. For a robust text recognition, the problem of stroke ordering also needs to be addressed. In a one-dimensional input sequence, the order in which strokes are written influences the recognition output, yet writers do not always follow the common consent on the order in which to draw complex characters.

References

1. Bhattacharya, N., Pal, U., Kimura, F.: A System for Bangla Online Handwritten Text. In: 12th Int'l Conf. on Document Analysis and Recognition, pp. 1367–1371 (2013)
2. Bhattacharya, U., Guin, K., Parui, S.K.: Direction Code Based Features for Recognition of Online Handwritten Characters of Bangla. In: 9th Int'l Conf. on Document Analysis and Recognition, vol. 1, pp. 58–62 (2007)
3. Bhattacharya, U., Nigam, A., Rawat, Y.S., Guin, K.: An Analytic Scheme for Online Handwritten Bangla Cursive Word Recognition. In: 11th Int'l Conf. Frontiers in Handwriting Recognition, pp. 320–325 (2008)
4. Fink, G., Vajda, S., Bhattacharya, U., Parui, S.K., Chaudhuri, B.B.: Online Bangla Word Recognition Using Sub-Stroke Level Features and Hidden Markov Models. In: Int'l Conf. of Frontiers in Handwriting Recognition, pp. 393–398 (2010)
5. Frinken, V., Bhattacharya, N., Pal, U.: Design of Unsupervised Feature Extraction System for On-Line Bangla Handwriting Recognition. In: 11th IAPR International Workshop on Document Analysis Systems (page accepted for publication, 2014)
6. Frinken, V., Peter, T., Fischer, A., Bunke, H., Do, T.-M.-T., Artieres, T.: Improved Handwriting Recognition by Combining Two Forms of Hidden Markov Models and a Recurrent Neural Network. In: Jiang, X., Petkov, N. (eds.) CAIP 2009. LNCS, vol. 5702, pp. 189–196. Springer, Heidelberg (2009)
7. Garai, G., Chaudhuri, B.B., Pal, U.: Online Handwritten Indian Script Recognition: A Human Motor Function Based Framework. In: 16th Int'l Conference on Pattern Recognition, vol. 3, pp. 164–167 (2002)

8. Gers, F., Schmidhuber, J.: Recurrent Nets that Time and Count. In: IEEE-INNS-ENNS Joint Conf. on Neural Networks, vol. 3, pp. 189–194 (2000)
9. Graves, A.: Offline Arabic Handwriting Recognition with Multidimensional Neural Networks. In: Guide to OCR for Arabic Scripts, pp. 297–314. Springer (2012)
10. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist Temporal Classification: Labelling Unsegmented Sequential Data with Recurrent Neural Networks. In: 23rd Int'l Conf. on Machine Learning, pp. 369–376 (2006)
11. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A Novel Connectionist System for Unconstrained Handwriting Recognition. IEEE Transaction on Pattern Analysis and Machine Intelligence 31(5), 855–868 (2009)
12. Graves, A., Schmidhuber, J.: Framewise Phoneme Classification with Bidirectional LSTM Networks. In: Int'l Joint Conf. on Neural Networks, vol. 4, pp. 2047–2052 (2005)
13. Katayama, Y., Uchida, S., Sakoe, H.: A new HMM for On-Line Character Recognition using Pen-Direction and Pen-Coordinate Features. In: 19th Int'l Conf. on Pattern Recognition, pp. 1–4 (2008)
14. Kittler, J., Hatef, M., Duin, R.P., Matas, J.: On Combining Classifiers. IEEE Trans. on Pattern Analysis and Machine Intelligence 20(3), 226–239 (1998)
15. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
16. Mondal, T., Bhattacharya, U., Parui, S.K., Das, K., Mandalapu, D.: On-line Handwriting Recognition of Indian Scripts – The First Benchmark. In: Int'l Conf. of Frontiers in Handwriting Recognition, pp. 200–205 (2010)
17. Nakagawa, M., Tokuno, J., Zhu, B., Onuma, M., Oda, H., Kitadai, A.: Recent Results of Online Japanese Handwriting Recognition and its Applications. In: Doermann, D., Jaeger, S. (eds.) SACH 2006. LNCS, vol. 4768, pp. 170–195. Springer, Heidelberg (2008)
18. Parui, S.K., Bhattacharya, U., Chaudhuri, B.B.: Online Handwritten Bangla Character Recognition Using HMM. In: Int'l Conf. on Pattern Recognition, pp. 1–4 (2008)
19. Parui, S.K., Bhattacharya, U., Shaw, B., Guin, K.: A Hidden Markov Models for Recognition of Online Handwritten Bangla Numerals. In: 41st National Annual Convention of CSI, pp. 27–31 (2006)
20. Plamondon, R., Srihari, S.N.: On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. IEEE Transaction on Pattern Analysis and Machine Intelligence 22(1), 63–84 (2000)
21. Rabiner, L.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE 77(2), 257–286 (1989)
22. Roy, K., Sharma, N., Pal, T., Pal, U.: Online bangla handwriting recognition system. In: 6th Int'l Conf. on Advances in Pattern Recognition, pp. 117–122 (2007)

A Ranking Part Model for Object Detection

Chaobo Sun and Xiaojie Wang

School of Computer, Beijing University of Posts and Telecommunications
{cbsun,xjwang}@bupt.edu.cn

Abstract. Object detection has long been considered a binary-classification problem, but this formulation ignores the relationship between examples. Deformable part models, which achieve great success in object detction, have the same problem. We use learning to rank methods to train better deformable part models, and formulates the optimization problem as a generalized convex concave problem. Experiments show that, using same features and similar part configurations, performance of detection by the ranking model outperforms original deformable part models on both INRIA pedestrians and Pascal VOC benchmarks.

Keywords: Object Detection, Deformable Part Model, Learning to Rank.

1 Introduction

Object detection is a task for localizing objects of specific categories, it has been playing a critical role in high-level image understanding. Previous models formulate object detection as a binary classification problem[9,17,13]. All candidate detections are judged a true object area or not. These detections can be sampled either by sliding windows on a feature pyramid[9,13], or from a shrunk space generated by objectness models[1,17].

Here comes the problem: Assuming that the feature of a detection candidate is $\boldsymbol{x}$, and with its label y. Classification models only focus on the relationship between $\boldsymbol{x}$ and y, while ignoring relationships between different $\boldsymbol{x}$. We argue that these types of relationships are also important: for example, if a $\boldsymbol{x_a}$ is better than $\boldsymbol{x_b}$(that is, candidate a with feature $\boldsymbol{x_a}$ have a higher overlap ratio with some objects than candidate b with feature $\boldsymbol{x_b}$), an ideal model should give a a higher score. Similarly, if two detections have close best overlap ratios, the model should give them close scores. Classification models fails to model these situations. As Figure 1 shows, we focus on "Why is a detection better than another" rather than "Why is a detection true".

This paper aims to overcome the shortcomings of previous object detection mentioned above. Our contributions are:

– We provide a ranking perspective on object detection. To search objects from candidate space, three types of information are available:item-wise, pair-wise and list-wise. Classification models use only item-wise information, while our model uses both pair-wise and list-wise information.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 414–423, 2014.

Fig. 1. In the image on the left-hand side, detection a and b are both true detections, but a is definitely a better detection than b; In the image on the right-hand side, c and d are both false detections whose overlap ratios with ground truth objects are below 0.5, but still we can tell c is better than d

- We propose a new objective function based on learning to rank theory, and apply it on deformable part models[13]. The objective function is a variant of LambdaRank[6].
- We formulate the optimization problem as a generalized-CCCP problem, and solve it in a similar way as CCCP[18].

We have a brief review on background works in section 2. The details of ranking formulation for object detction are discussed in section 3.2. In section 3.3 we describe our objective function. Section 4 shows the procedure of optimizing the objective function. Section 5 shows the results on well-known object detection datasets. In section 6 we conclude our work and discuss possible improvements.

2 Related Work

Research on generic object detection is originating from person detection[9]. From then on sliding-window methods with HOG pyramids have been a main stream on object detection. For every category of objects, sliding-window build a set of templates to represent all its poses. During training, cropped objects and backgrounds are extracted to train the template. During detecting, a matching score is computed at every position in the feature space, then the position with scores above a threshold is considered to be an object position[9].

Deformable part models(DPM)[13] have greatly pushed the research on object detection. As a variation of sliding windows, DPMs establish a set of hierarchical templates for every category of objects. Each template is organized into a root and its parts. Not only the appearance(vision features) of roots and parts, but also the parts' relative positions(structural features) to the root are taken into consider, so that DPMs can tolerate a certain degree of deformation.

Our work is mainly based on DPMs. We follow the definition of hierarchical templates, but improve the training procedure. Unlike an equivalent conversion from latent svm to latent struct svm[20], we use a totally new objective function based on the theory of learning to rank, and adopt it suitable for object detction.

We notice that there are several works on strong supervised models[4,2] that need additional annotations for parts. Additional annotations may help but reduces the difficulty of detection task. Our model can be applied on such models easily.

The work from Balschko[3] uses ranking svm[15] to model object detection, it is similar to our work in the sense that we both want to capture the essence why a detection is better than another. But there are significant differences: they use ranking to handle unlabeled data rather than take object detection as a ranking problem; they do not model the latent variables while we do, they use a svm-style objective function while we use a cross entropy style one, which is more flexible for adding list-wise information.

Learning to rank methods use cross entropy to measure distribution diverge between empirical probability and model probability[5]. LambdaRank[6] modifies the form of objective function by interpolating information retrieval measures. We adopt LambdaRank for more efficient computing in object detection.

During optimization of the cross-entropy style objective function, we find it an ensemble of Convex Concave Problems[18]. We also notice that finding a convex lower bound for the concave part of every CCCP would make the whole problem convex, so the two-stage optimization for original CCCP is suitable for our new problem.

3 Model

3.1 Deformable Part Models: A Review

Before introducing our model, let us have a brief review on deformable part models.

An object detection model defines a score function for detections. The function gives a confidence on the detections. Let x be the position of detections, I be the image on which detection is performed, a one-layer linear score function is defined as:

$$f(x, I; \omega, b) = \omega \cdot H_I(x) + b \tag{1}$$

where H_I is the feature pyramid, and $H_i(x)$ is the features covered by x. b is a bias.

DPMs introduce parts into original flat templates. Because locations of parts are un-observed variables, they may take any position in the image. Let position of parts be z, then the score function of DPM is simply a maximum of all possible zs:

$$f(x, I; \omega) = \max_z g(x, z, I; \omega) \tag{2}$$

The function g, which is a score function for joint x and z, is defined as:

$$g(x, z, I; \omega) = \omega_0^a \cdot H_I(x) + \sum_{k=1}^{K} [\omega_k^a \cdot H_I(z_k) - \omega_k^d \cdot d(x, z_i, v_i)] + \omega^b \tag{3}$$

Where $d(x, z_i, v_i)$ is the deformation function for z_i relative to x. v_is are ideal anchors for the ith part, they could be defined either heuristically[13] or by pre-defined rules[19]. We use ω to represent all parameters: ω^a for parameters of appearance, ω^d for parameters of deformation, and ω^b for bias. Note that $g(x, z, I; \omega)$ is linear function of ω

To train such functions, DPM then defines an svm-style loss function. Suppose we have n samples (x_i, I_i, y_i), where $y_i \in +1, -1$ representing whether x_i on I_i is a true detection or not.

$$L(\omega) = \frac{1}{2}\|\omega\|^2 + C \cdot \sum_{i=1}^{n} max(0, 1 - y_i f(x_i, I_i; \omega)) \tag{4}$$

3.2 Ranking Perspective on Object Detection

Classical learning to rank systems focus on selecting relevant items from a set of candidates. Object detection is similar to these models, if we interpret the searching space of object detection as a set of candidates. Following the way of Pascal VOC evaluation[11], the set of candidates are all positions of all images in a dataset. The aim of object detection is then selecting candidates that have more overlap ratios with ground truth objects.

In information retrieval systems, when modeling the relationship of some samples x and their corresponding labels y, there are three types of information:

- item-wise information, the direct relationship between x and y.
- pair-wise information, the relationship between a paired (x_i, x_j).
- list-wise information, the importance of x's position in the ordered list.

The key difficulties for applying ranking models on object detection is its large space of candidates. All rectangles in images are candidates to rank. Sliding window methods largely reduce the number of candidates by making the constraint that all candidates should be in certain sizes[9], while in recent years there are several useful technologies directly aiming to shrink the space of candidates[17,8].

We use a sliding window way to generate candidates, but it is very convenient to apply our model on a shrunk space of candidates.

3.3 Ranking DPM

Different from original DPM, we do not generate detections with their labels, we organize samples into a list of pairs. Instead of representing the list explicitly, we use a set of pairs, J, to represent the list. Every pair (i, j) in the set J means that detection x_i has a higher overlap ratio than x_j with some ground truth objects. Then the ordered list defined by J is strict partially ordered[16], and contains sufficient information of relationships between examples.

We define a simple empirical distribution on every pair (i, j) in J:

$$\bar{\mathrm{P}}_{\mathrm{ij}} \equiv \begin{cases} 1, (i,j) \in J \\ 0, (j,i) \in J \end{cases} \tag{5}$$

The empirical probability is a statistical measure of the pair-wise information in training datasets. During the train stage of our model, the score function is applied on each detction, and the score outputted would also generate a model distribution. We define it in a form of sigmoid function:

$$P_{ij} \equiv \frac{1}{1 + e^{-\sigma(f_i - f_j)}} \tag{6}$$

For simplicity, we use f_i to denote $f(x_i, I_i; \omega)$.

The two distributions should be as close as possible. We use cross entropy to measure the divergence of them:

$$C_{ij} = -\bar{P}_{ij} log(P_{ij}) - (1 - \bar{P}_{ij}) log(1 - P_{ij}) \tag{7}$$

Combining Eq.(5) ,(6) and (7), we got:

$$C_{i,j} = log(1 + e^{-\sigma(f_i - f_j)}) \tag{8}$$

To some extent, Eq.(8) can be a loss function on a single pair. Before summing up the C_{ij}s, it is necessary to examine how important the pair is in the whole strict ordered list. Note that C_{ij} is an increasing function of $f_i - f_j$, and an obvious fact is that for a pair of detections, they should be scored more discriminatively when they have bigger differences on overlap ratios. So it is reasonable to give the pairs weights according to the differences of overlap ratios.

This configuration is similar to [10], while we replace the changes of information retrieval measures with the differences of overlap ratios. As we discussed above, differences of overlap ratios plays a role similar to the changes of information retrieval measures, they both denote the importance of the binary relationship within the whole list.

Then we have the following loss function:

$$L(\omega) = \alpha \cdot \|\omega\|^2 + \sum_{(i,j) \in J} log(1 + e^{-\sigma(f_i - f_j)})(ov_i - ov_j) \tag{9}$$

α is a factor for regularization item. Note that x_i is a better detection than x_j, so the weight factor $ov_i - ov_j$ is always positive.

4 Optimization

4.1 Generalized CCCP

The loss function for every pair is not convex, but it is semi-convex in the sense that, the loss function is convex under specific constraints of f_i.

We have the following lemma:

Lemma 1. *if $f(x)$ and $g(x)$ are both convex, $g(x)$ is non-decreasing, then $g(f(x))$ is convex.*

With this lemma, we can prove that:

Theorem 1. *If f_i is concave and f_j is convex, Eq.*(9) *is convex.*

Proof $log(1 + e^{\sigma x})$ is convex and non-decreasing, then the loss function is convex if $f_j - f_i$ is convex. On the other hand, if f_i is concave, then $-f_i$ is convex, and then we get a convex $f_j - f_i$.

Recall that f is maximum of some linear functions, and therefore is convex. So, it is f_i that make Eq.(9) non-convex. But if we find a *concave* lower bound for f_i, the loss function is convex. As suggested in [18], we can obtain the concave function by fixing f_i with its best latent variable(part locations):

$$\begin{aligned} h_i &= g(I_i, x_i, z_i^*; \omega) \\ z_i^* &= \underset{z}{\operatorname{argmax}}\, g(I_i, x_i, z; \omega) \end{aligned} \tag{10}$$

Then the linear function h_i is both convex and concave, and thus is a concave lower bound for f_i, and the loss function is convex if we replace f_i with h_i.

As we show above, $f_j - f_i$ is a Convex-Concave Problem(CCCP)[18], and the whole loss function is a so-called "Generalized CCCP".

4.2 Optimization Procedure

To solve Generalized CCCP, we follow a similar way to solve standard CCCP, which uses an iteration of two stages:

- *Latent Variable Finding.* In this stage, for every i that there exists some $(i, j) \in J$, we extract z_i^*, and calculate h_j based on z_i^*.
- *Optimization.* In this stage, we try to optimizing the convex problem $\alpha \cdot \|\omega\|^2 + \sum_{(i,j)\in J} log(1 + e^{-\sigma(h_i - f_j)})(ov_i - ov_j)$.

It is worth noting that, for pairs (i, j) in the set of pairs J, some positive examples may be in the position of i in some pairs, while in the position of j in others. When scoring these examples, we have to calculate $h(x, I; \omega)$ and $f(x, I; \omega)$ simultaneously. For convenience of computing, we use $h(x, I; \omega)$ for scoring all examples from positive images.

In the latent variable finding stage, we use three subroutines:

detect_best denotes the procedure of finding z^* for ground truth boxes in I. Distance transform[12] is used in the max finding. All z^*s are extracted with their overlap ratios with ground truth boxes.

detect_hard denotes the procedure of detection on negative images, the positions with top scores are considered to be hard examples, and are selected. All examples are labeled with overlap ratio 0.

generate_pairs generates the set of pairs J using all examples, every pair with different overlap ratios are put into the set.

During the optimization stage, we use L-BFGS[7] as the loss function is derivable:

$$\nabla L(\omega) = 2\alpha \cdot \omega + \sum_{(i,j)\in J} \frac{-\sigma}{1 + e^{\sigma(h_i - f_j)}}(ov_i - ov_j)[\nabla h_i - \nabla f_j] \tag{11}$$

The training procedure is illustrated in Algorithm 1.

Data:
Positive Examples:$P = \{(I_1^P, B_1), \ldots, (I_n^P, B_n)\}$
Negative Examples:$N = \{I_1^N, \ldots, I_m^N\}$
Initial model parameters:ω^{old}
Result: New model parameters:ω^{new}

$\omega_0 := \omega^{old}$;
for *t:=1* **to** T **do**
  for *i:=1* **to** n **do**
    $F := detect\_best(I_i^P, \omega_{t-1})$;
    Add F to F_P;
  end
  for *j:=1* **to** m **do**
    $F := detect\_hard(I_j^N, \omega_{t-1})$;
    Add F to F_N;
  end
  $J := generate\_pairs(F_P, F_N)$;
  $\omega_t := l\_bfgs(J)$;
end
$\omega^{new} := \omega_T$

Algorithm 1. optimization

5 Experiments

We have evaluated our method on two well known datasets: INRIA pedestrians[9] and PASCAL VOC 2007[11]. Performance is measured in term of Average Precision (AP) according to the PASCAL VOC protocol[11].

We first initialize models with [14], and then apply our training procedure. To show whether the new objective function captures more information of training sets, we use the same features(an adopted version of HOG) suggested by [13].

5.1 INRIA Person

INRIA pedestrians dataset contains 1832 training images and 741 testing images [9]. Only persons are labeled with their bounding boxes. We evaluate our models and original DPM in Pascal VOC measures.

Figure 2 Shows the comparison of performances on INRIA dataset, our model promote the mAP measure from *0.8520* to *0.8571*. Simultaneously, our model gives a much smaller number(*1020*) of detections on test dataset compared to original DPM(*2952*). These results clearly shows that our model provides a more discriminative divide for object-related detections and backgrounds. The reduction of detection number would be very useful in practice.

But it is also worth noting that, it is not the main difference that our model provides a better divide. Our model has a higher precision almost at any recall value.

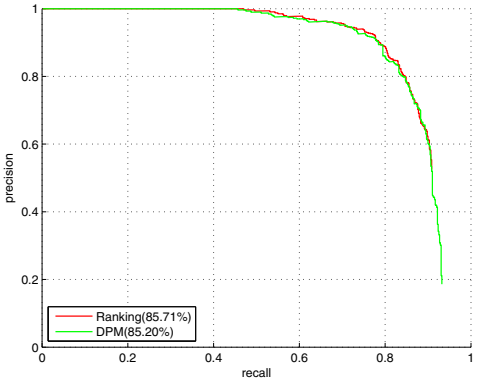


Fig. 2. Compared to original DPM, our model achieves better AP value while giving fewer candidates

5.2 Pascal VOC

Pascal VOC dataset is a more challenging benchmark. It contains 20 categories of objects, more complex backgrounds. Objects within each category have significant differences in appearances, scales and poses(for animals). And the numbers of objects in each category vary largely[11].

We evaluate our model and original DPM on the dataset. For convenience of comparison, we do not apply any post-processing technologies such as box predicting and context predicting.

Table 1 shows the results of our model(Ranking) and original DPM on Pascal VOC 2007. Our model outperforms original DPM on *bicycle*, *chair*,*dog*, *motorbike*, *sheep*, *train* and *tvmonitor*, while have a poorer performance on *bus* and *sofa*. Results on other categories are close.

The results show that in most cases, our model captures more characteristics of training data, and the characteristics are helpful or at least not harmful when applied on testing data. But still in some cases, the characteristics do play a role like noise.

Table 1. Evaluation results on Pascal VOC 2007, in Average Precision(%)

class	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow
DPM	31.04	59.73	4.02	12.12	23.47	**50.55**	54.63	17.12	17.71	22.79
Ranking	31.04	**59.93**	4.02	12.12	**23.77**	50.51	54.63	17.12	**17.95**	22.80

	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	AVG
DPM	22.14	4.59	58.29	47.88	41.76	8.54	18.76	**35.86**	45.37	40.84	30.86
Ranking	22.14	**4.81**	58.29	**48.01**	**41.90**	8.54	**20.00**	35.50	**45.39**	**40.90**	30.96

6 Conclusion and Future Work

In this paper, we proposed a new modeling perspective on object detection: learning to rank. Following this perspective, we defined an ranking model based on information retrieval theory and DPM, and then formulated the optimization problem to a generalized CCCP. We evaluated our model on INRIA and Pascal VOC datasets, and performances on both benchmarks outperform original DPM which is based on latent svm.

According to our observation, the usefulness of the model is not directly linked with the shallow features of datasets, like the number of training examples or the ratio of positive examples and negative examples. The investigation of reasons are a main direction of our future work.

While ranking is useful for object detection, there are still differences of object detection with classical ranking problems: a much larger space of candidates. So it would be more useful to run the ranking model on a smaller space of candidates generated by objectness methods.

Acknowledgments. This work was partially supported by National Natural Science Foundation of China(Project No.61273365), National High Technology Research and Development Program of China(No. 2012AA011104) and discipline building plan in 111 base(No. B08004) and Engineering Research Center of Information Networks, Ministry of Education.

References

1. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 73–80. IEEE (June 2010)
2. Azizpour, H., Laptev, I.: Object detection using strongly-supervised deformable part models. csc.kth.se (2012)
3. Blaschko, M.B., Vedaldi, A., Zisserman, A.: Simultaneous object detection and ranking with weak supervision. In: NIPS, vol. 1, p. 5 (2010)
4. Branson, S., Perona, P., Belongie, S.: Strong supervision from weak annotation: Interactive training of deformable part models. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 1832–1839 (November 2011)
5. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 89–96. ACM (2005)
6. Burges, C.J., Ragno, R., Le. Learning, Q.V.: to rank with nonsmooth cost functions. In: NIPS, vol. 6, pp. 193–200 (2006)
7. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing 16(5), 1190–1208 (1995)
8. Cheng, M.-M., Zhang, Z., Lin, W.-Y., Torr, P.H.S.: BING: Binarized normed gradients for objectness estimation at 300fps. In: IEEE CVPR (2014)
9. Dalal, N., Triggs, B., Europe, D.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893. IEEE (2005)

10. Donmez, P., Svore, K.M., Burges, C.J.: On the local optimality of lambdarank. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 460–467. ACM (2009)
11. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC 2007) Results, http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
12. Felzenszwalb, P., Huttenlocher, D.: Distance transforms of sampled functions. Technical report, Cornell University (2004)
13. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(9), 1627–1645 (2010)
14. Girshick, R.B., Felzenszwalb, P.F., McAllester, D.: Discriminatively trained deformable part models, release 5, http://people.cs.uchicago.edu/~rbg/latent-release5/
15. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 133–142. ACM (2002)
16. Dan, A.: Simovici and Chabane Djeraba. In: Mathematical Tools for Data Mining. Springer (2008)
17. van de Sande, K.E.A., Uijlings, J.R.R., Gevers, T., Smeulders, A.W.M.: Segmentation as selective search for object recognition. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 1879–1886. IEEE (2011)
18. Yuille, A.L., Rangarajan, A.: The concave-convex procedure (cccp). Advances in Neural Information Processing Systems 2, 1033–1040 (2002)
19. Zhu, L., Chen, Y., Yuille, A.: Learning a hierarchical deformable template for rapid deformable object parsing. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(6), 1029–1043 (2010)
20. Zhu, L., Chen, Y., Yuille, A., Freeman, W.: Latent hierarchical structural learning for object detection. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1062–1069. IEEE (2010)

Regular Decomposition of Multivariate Time Series and Other Matrices

Hannu Reittu, Fülöp Bazsó, and Robert Weiss

Technical Research Center of Finland, VTT
Wigner Research Center of Physics, Hungary

Abstract. We describe and illustrate a novel algorithm for clustering a large number of time series into few 'regular groups'. Our method is inspired by the famous Szemerédi's Regularity Lemma (SRL) in graph theory. SRL suggests that large graphs and matrices can be naturally 'compressed' by partitioning elements in a small number of sets. These sets and the patterns of relations between them present a kind of structure of large objects while the more detailed structure is random-like. We develop a maximum likelihood method for finding such 'regular structures' and applied it to the case of smart meter data of households. The resulting structure appears as more informative than a structure found by k-means. The algorithm scales well with data size and the structure itself becomes more apparent with bigger data size. Therefore, our method could be useful in a broader context of emerging big data.

1 Introduction

Szemerédi's Regularity Lemma (SRL), [14], is fundamental in graph theory. Roughly speaking, it states that any large enough graph can be approximated arbitrarily well by a bounded cardinality collection of pseudo-random bipartite graphs, a structure called 'an ϵ-regular partition'. In many graph problems it is possible to substitute a graph with such a simple random-like structure, thus opening a way for analysis, [7,5].

Despite very impressive theoretical impact, SRL has not attracted much interest among applied scientists. Direct use of SRL is prohibited by its nature as an existence result. For instance, the existing algorithms for finding regular partitions are somehow unpractical and require enormous size of the graph, never seen in applications. We suggest an efficient method that is able to find a counterpart of regular partition, that we call regular decomposition, in graphs and matrices found in applications.

Our approach of finding the regular decomposition is based on statistical model fitting using likelihood maximization,[9,10] and can be seen also as a variant of so called Stochastic block modeling, see e.g. [11,2]. The novelty of our approach comes from systematic use of SRL as a prototype and theoretical back up of structure we are looking for, defining a new *modeling space* for a graph-like data. Regular decomposition is a partition of nodes into k sets in such a way that structure between sets and inside sets are random-like.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 424–433, 2014.

Another important point is that such regular structures can be found in a realistic time. Even for very large data, or 'big data', the structure can be found from a modest size sample and the rest of data is partitioned in just one-pass. The number of sets in the partition of a regular decomposition, k, is optimized. The algorithm increases number of clusters as long as large clusters are created and then stops. This halting point is determined using Rissanen's minimum description length principle, MDL, see in [6], as the first local minimum of description length.

A real matrix can be treated as a weighted graph, where weights are the matrix elements and thus our graph-based approach can be used. In this paper we focus on case of multivariate time series. A key ingredient of our method is a Poisson random matrix model allowing maximum likelihood fitting of data to the model. The regular partition is a clustering of time series into few groups. We illustrate our method on analyzing time series of households electricity consumption measured by smart meters. Our first results are encouraging: reasonable classes of households could be identified comparing favorable to other customary methods like k-means.

1.1 Other Approaches of SRL Related Clustering and Graph Compression

An alternative, spectral approach for finding a block structure in real matrices by associating protruding eigenvalues or singular values to regular-type large blocks of real matrices, was developed and founded by Bolla, [3,4]. The following works [12,13] relax SRL so that a relaxed SRL can be used in the area of practical machine learning, producing some promising results. Other promising approaches for 'graph compression' can be found in [16,8], with different methods and without direct references to Regularity Lemma.

2 Regular Decomposition of Multivariate Time Series

We now describe our SRL-inspired approach for modeling of multivariate discrete time series as a special case of weighted graphs or real matrices. Consider a set of n non-negative finite series of length m, as an $n \times m$ matrix:Such a

$$\{X_i(t)\}_{i=1,2,\cdots,n}^{t=1,2,\cdots m}.$$

n is assumed to be large and formally we consider case $n \to \infty$, while m is a fixed constant. Matrix elements are reals with finite precision, p. The space of all such matrices is denoted as $\mathcal{M}_{n\times m}$. Such matrices can be seen as weighted bipartite graphs, with rows and columns as bipartition of nodes. Each time series is a node and each time instance is a node and between them there is a link with weight equal to the corresponding matrix element with those particular row- and column indexes. Say, for a row i and column j, the corresponding weight equals to $X_i(j)$, value of the time series i at the moment of time j. Our goal is to find 'regular' groups of time series, in the form of a particular partition of rows.

The main new tool is the following random multi-graph interpretation of a data matrix. For a given weighted link $\{i, j\}$ with weight $X_i(j)$, we associate a Poisson random variable, with expectation equal to the weight of the link. All links are independent. As a result we can associate to every data matrix a random multi-graph, where multiple links between pair of nodes are allowed.

The target is to 'compress' such a random multi-graph, using a regular decomposition of rows. More precisely, the goal is to minimize the expected code length over such a random source. In this case the modeling space is infinite, because the weights are just real numbers. To tackle this problem, we assume that we have a series of discrete modeling spaces with a finite precision, approaching the real-parameter model as the precision goes to infinity. In practical applications it suffices to have a finite modeling space, corresponding to a finite precision of reals. We define a modeling space of matrices from space, $\mathcal{M}_{n\times m}(p, a)$ with finite accuracy of real matrix elements (reals from range $[0, a]$ and the matrix elements are approximated by rationals from a range of $p + 1$ values):

Definition 1. *A Poisson random matrix modeling space, $\mathcal{M}_{n/k}(p, a)$, of regular decompositions of n time series of length m from $\mathcal{M}_{n\times m}(p, a)$ into k non-empty classes: $\mathcal{M}_{n/k}(p, a) = \{$set of all integer valued $n \times m$ random matrices, Y where rows are partitioned into k non-empty sets $V_1, V_2, \cdots, V_k$ and where matrix elements are all independent random variables and distributed according to Poisson distribution: matrix element $Y_i(j)$ with $i \in V_\alpha$ is a Poisson random variable with parameter $\lambda_\alpha(j)$ that belongs to a rational range $\lambda_\alpha(j) \in \{r : r = \frac{g\lceil a\rceil}{p}$, p is a fixed integer, and g is an integer s.t. $0 \le r \le a$, a is fixed positive real$\}\}$.*

The elements of the modeling space are denoted by Θ_k, incorporating a partition $V_1, V_2, \cdots, V_k$ and set of Poisson parameters $\{\lambda_\alpha(i)\}$. For a given integer $m \times n$ matrix X we can compute its probability, $P(X \mid \Theta_k)$, assuming a model Θ_k. Particularly the maximum likelihood model for a matrix X is found from a program:

$$\Theta_k^*(X) := \arg \max_{\Theta_k \in \mathcal{M}_{n/k}} P(X \mid \Theta_k), \tag{1}$$

where we omitted p, a-arguments since it is assumed in sequel. However we have a different problem: our real matrix A is interpreted as a source of random integer matrices X as described above. The task is to find a model that maximizes the expected likelihood of matrices drawn from such a source. According to information theory this can be seen as compression of the random source. Using log-likelihood this task is formulated as:

$$\Theta_k^*(A) := \arg \max_{\Theta_k \in \mathcal{M}_{n/k}} \sum_X P(X \mid A) \log(P(X \mid \Theta_k)) =$$

$$\arg \max_{\Theta_k \in \mathcal{M}_{n/k}} \left(\sum_X P(X \mid A) \log \frac{P(X \mid \Theta_k)}{P(X \mid A)} + \sum_X P(X \mid A) \log P(X \mid A) \right) =$$

$$\arg \max_{\Theta_k \in \mathcal{M}_{n/k}} (-D(P_A \mid\mid P_{\Theta_k})) - H(P_A)),$$

where sum is over all non-negative $n \times m$ integer matrices, D is the Kullback-Leibler divergence and H is entropy of a distribution. Thus we end up with following minimization of Kullback-Leibler divergence between the two distributions P_A and P_{Θ_k}:

$$\Theta_k^*(A) := \arg \min_{\Theta_k \in \mathcal{M}_{n/k}} D(P_A \mid\mid P_{\Theta_k}), \tag{2}$$

where we omitted the entropy term that does not depend on Θ_k. We call the following expression the expectation of maximum likelihood coding length of the source A:

$$l_k(A) := D(P_A \mid\mid P_{\Theta_k^*(A))}) + H(P_A), \tag{3}$$

Such an interpretation follows from basic of information theory (see e.g.[15]):

Theorem 1. *(Kraft's inequality) For a finite alphabet: $L = \{1, 2, \cdots m\}$ there exists a binary prefix coding scheme with (integer) code lengths $\{l_1, l_2, \cdots, l_m\}$ iff $\sum_{i=1}^{m} 2^{-l_m} \leq 1$ and where l_i is the length of the code for letter i.*

The prefix coding is such that no code is a prefix of another and coding is an injective mapping of letters to the set of all binary tuples. As a result, if we have a probability distribution P in L, we can be sure that there exists a prefix coding with code lengths $l_i = \lceil -\log P(i) \rceil$, because such integers fulfill the Kraft's inequality as a result of normalization of the probability distribution $\sum_i P(i) = 1$. Following the line of MDL, we can define a distribution in the space of matrices $\mathcal{M}_{n \times m}$ that we call normalized maximum expected likelihood distribution $P_{nml,k}$:

$$P_{nml,k}(A) = \frac{2^{-l_k(A)}}{\sum_{B \in \mathcal{M}_{n \times m}(p)} 2^{-l_k(B)}}, \tag{4}$$

Thus we proved:

Proposition 1. *For any matrix $A \in \mathcal{M}_{n \times m}$, there exists a prefix coding with code length:*

$$l_{nml,k}(A) = l_k(A) + COMP(\mathcal{M}_{n/k}),$$

where the model complexity is by definition:

$$COMP(\mathcal{M}_{n/k}) := \log \left(\sum_{B \in \mathcal{M}_{n \times m}} 2^{-l_k(B)} \right)$$

and $l_k(\cdot)$ is defined by Eqs. (2)-(3).

Remark 1. The model complexity, as it often happens, is uncomputable since it requires solving a program for a general matrix. An asymptotic formula could be possible. Based on our experience and results with simpler case of binary matrix we just conjecture:

Conjecture 1. For fixed k and m, and with $n \to \infty$, we have the asymptotic:

$$COMP(\mathcal{M}_{n/k}) \sim \log(S2(n,k)) \sim n \log k,$$

where $S2(n,k)$ is the Stirling number of the second kind.

The Stirling number is just number of partitions of n element set into k non-empty sets. Choosing the best partition among huge collection is the hardest part of our procedure.

Our next point is the MDL-optimum. However, we will not try to find the global optima, since it would require searching the whole range of possible values: $1 \leq k \leq n$. For large matrices this would be computationally too difficult. On the other hand, in the spirit of SRL, we suggest that it makes sense to search for large scale regular structure, corresponding to a local MDL optimum with smallest value of k. Even such a structure could be very useful for getting an overall picture of large data sets. Our Poisson-modeling space belongs to the exponential family. As a result the log-likelihood function is monotonously increasing function of k, preventing spurious local minimum of MDL, see [6]. Another point is that this space allows computation of goal function in a simple form. Thus we end up with the following program for finding the large scale regular structure of multivariate time series, belonging to the space $\mathcal{M}_{n\times m}$:

PROGRAM 1

Input: $A \in \mathcal{M}_{n\times m}$ and an integer k', the maximal depth of search. Output: optimal regular structure $\Theta_f^*(A) \in \mathcal{M}_{n/f}$ where

$$f = \inf\{k : 1 \leq k \leq k' - 1 \leq n - 1, l_{nml,k+1}(A) \geq l_{nml,k}(A)\}$$

or if f is not found in the range $1 \leq k \leq k'$, conclude that 'no regular structure found and the best structure found is $\Theta_{k'}^*(A)$'.

Next we need a program that find optimal structure for fixed k and matrix A. For this we just need to minimize the Kullback-Leibler divergence, as stated in Eq. 2. In both models P_A and $P_{\Theta_k(A)}$, matrix elements are independent and Poisson distributed. K-L divergence between two Poisson distributions with parameters λ and λ_0 is denoted and computed as:

$$D(\lambda \mid\mid \lambda_0) = \lambda_0 - \lambda + \lambda \log \frac{\lambda}{\lambda_0}. \tag{5}$$

If we fix partition of rows into k non-empty sets we should choose the parameters for Poisson variables. We use notation, row i belongs to row class $u(i)$. For a set V_i of partition we should select the parameters $\lambda_i(t)$, $1 \leq t \leq m$. As is well known the parameter that gives the maximum likelihood is the average:

$$\lambda_i(j) = \frac{\sum_{s \in V_i} a_s(j)}{\mid V_i \mid}$$

in sequel such a selection is always assumed. Using the two previous relations we end up with:

Proposition 2. *The regular decomposition of multiple time series, given by a matrix* $(A)_{i,j} = a_i(j)$ *from* $\mathcal{M}_{n\times m}$ *with fixed size* k *correspond to solution of the program:*

$$\min_{V_1,V_2,\cdots V_k} \sum_{1\le i\le n; 1\le t\le m} (\lambda_{u(i)}(t) - a_i(t)\log\lambda_{u(i)}(t)).$$

where $\lambda_i(j) = \frac{\sum_{s\in V_i} a_s(j)}{|V_i|}$ *and* $V_1, V_2, \cdots V_k$ *is a partition of* n*-rows into* k *non-empty sets and* $u(\cdot)$ *maps rows to the sets of the partition they belong.*

The trivial case is when we have k classes of time series where within each class all time series are identical. It is easy to see that our program finds such a structure.

Proposition 3. *If time series have* k *classes and within each class time series are identical but different across different classes. Then the program in Prop.2 finds such a structure.*

Proof. The Program is equivalent of minimizing sum of K-L divergences, D for all random multi links. It is known that $D \ge 0$ between any distributions and equality to 0 happens when the distributions are equal. Obviously this can be achieved in this case by using k-classes as regular partition, resulting in the global optima.

Next we present a result showing that under some condition 'noisy' clusters can be also classified correctly. The following lemma is needed to make assumptions in following Proposition 4 natural:

Lemma 1. *For every positive real* x *and for any fixed real* $y > 0$

$$x - y\log x \ge y - y\log y$$

and equality holds only when $x = y$.

Proposition 4. *Assume that we have* k *noisy clusters, meaning that* n *time series* $\{X_i(t)\}$ *can be clustered into* k *clusters in such a way that all members of any cluster have the same expected time series and use the notation if* $i \in V_\alpha$ *then* $EX_i(t) = \lambda_\alpha(t)$ *and* $X_i(t) = \lambda_\alpha(i) + \phi_i(t)$*, for all* t*. The discrete time is in the range from* 1 *to* m*. The noise components* ϕ_i *are assumed uniformly, over* i *and time, upper bounded by a constant. Assume that all mean profiles* $\lambda_\alpha > 0$ *and are bounded above by a constant, assume that all profiles* $\{\lambda_\alpha\}$ *fulfill condition that if* z *is a convex combination of these profiles, then* $\sum_{1\le t\le m}[z(t) - \lambda_\alpha(t)\log z(t) - (\lambda_\alpha(t) - \lambda_\alpha(t)\log\lambda_\alpha(t))] > cm^u$*,* $c, u > \frac{1}{2}, \forall\alpha$*, unless* $z = \lambda_\alpha$*. Assuming* k *is fixed and* $n \to \infty$ *in such a way that all clusters have size that is a fixed ratio from* n*, then noisy clusters are detected asymptotically correctly in the sense that probability of misclassification is exponentially small with the length of time series* m.

Proof. First we note that for average profiles the strong law of large numbers is valid, since the noise is uniformly bounded. Thus with probability 1 asymptotically, as $n \to \infty$, we can replace average profiles by their expected profiles, the errors are $o(1)$ and are not written explicitly in sequel. Let us show that Proposition 4 holds. Let us assume another partition instead of noisy clusters. Then the Poisson parameters of such clusters, assuming that they are also large, are convex combinations of the parameters of noisy clusters, denote one of them by $z(t)$. Let $X_i(t) = \lambda_\alpha(t) + \phi(t)$. The probability that it is better fitted to a new cluster with parameters $z(t)$ is

$$P\{\sum_t [\lambda_\alpha(t) - (\lambda_\alpha(t) + \phi_i(t)) \log \lambda_\alpha(t)] <$$

$$< \sum_t [(z(t) - (\lambda_\alpha(t) + \phi_i(t)) \log z(t)]\}.$$

Equivalently the condition is written (for brevity time is not written) as:

$$\sum_t [z - \lambda_\alpha \log z - (\lambda_\alpha - \lambda_\alpha \log \lambda_\alpha) - \phi_i \log \frac{\lambda_\alpha}{z}] > 0.$$

According to assumption (see Lemma 1 that guarantees a positive bound of following expression in any point when $z(t) \neq \lambda_\alpha(t)$):

$$\sum_t [z - \lambda_\alpha \log z - (m_\alpha - \lambda_\alpha \log \lambda_\alpha)] > cm^u$$

As a result the claim holds if $\sum_t \phi_i(t) \log(\lambda_\alpha / z) \leq cm^u$. All variables

$$\phi_i(t) \log(\lambda_\alpha(t)/z(t))$$

are uniformly bounded by some positive constant, g, and have zero expectation. As a result Azuma's inequality for sum of independent summands holds. According to this result our desired event has probability with a bound:

$$P(\sum_t \phi_i(t) \log(\lambda_\alpha/z) \leq cm^u) \geq 1 - e^{\frac{-c^2 m^{2u}}{g^2 m}}.$$

Since $2u > 1$ we have that for any other clustering than that underlying noisy clusters, only some time series would benefit with exponentially small probability.

For clarity we describe the most basic algorithm for finding the optimal partition using a greedy algorithm of the expectation-maximization (EM) type, [9]. Generally it finds only a local optima and it should be repeated several times to hit global optima (or replaced by simulated annealing-type algorithm).

Algorithm 1

INPUT a matrix $A \in \mathcal{M}_{n \times m}$, k a fixed integer $1 \leq k \leq n$.

STEP 1, find a random partition of set of n rows V into k non-empty sets $\mathcal{U}$,

STEP 2 (expectation) compute k-rows of Poisson parameters matrix $\Lambda = \{\lambda_\alpha(j)\}_{1\le\alpha\le k}^{1\le j\le m}$ corresponding to A and given partition $\mathcal{U}$, as averages over sets.

STEP 3 (-maximization) sequentially consider all rows i from 1 to n, and find an updated partition $\mathcal{U}'$, consider all variants $u'(i) = 1, 2, \cdots, k$ and compute target function for each choice:

$$\sum_{1\le t\le m} (\lambda_{u'(i)}(t) - a_i(t) \log \lambda_{u'(i)}(t))$$

Take $u'(i) = s$, that has the smallest value among k variants.

STEP 4 Check IF $\mathcal{U}' = \mathcal{U}$, then STOP, $\mathcal{U}'$ is the (local) optimal partition; other wisely : put $\mathcal{U} = \mathcal{U}'$ and GOTO to STEP 2.

It should be noted, that if the matrix is multiplied by a constant, the partition of regular structure does not change. This is due to linearity of K-L divergence for Poisson variables, see Eq. (5), with respect to scaling of parameters.

3 Classification of Households Based on Their Recorded Electric Power Usage Time Series

As an example we analyze, using regular decomposition method, smart meter reading of 783 residential, SME and other subjects of electric consumption provided by ISSDA, Irish Social Science Data Archive [1] with additional information on each household available. The time interval is 535 subsequent half hours or around 10 days. Each row of raw data was normalized by dividing all elements by the average of the row. We wish to have clustering based on a patterns of activity not on the level of power consumption. The overall result of regular decomposition is shown in Fig 1. The regular structure reveals many interesting

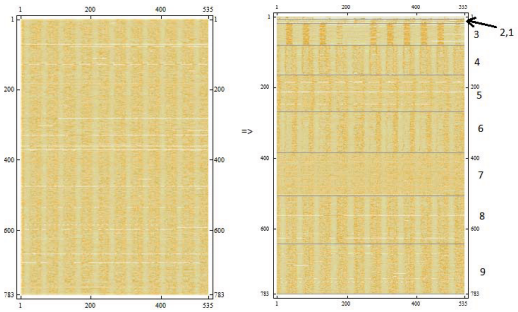


Fig. 1. Data matrix, left hand picture raw data of 783 users with 535 half hour period. Right hand side picture the same data after rearranging rows into 9 groups of regular behavior. $k = 9$ was found optimal using MDL-criteria. Pixel darkness corresponds to value. For instance group 3 has first 3 darker vertical bands, corresponding to working day activity, followed by 2 weaker bands of Saturday and Sunday. This is line that the group 3 constitutes of companies.

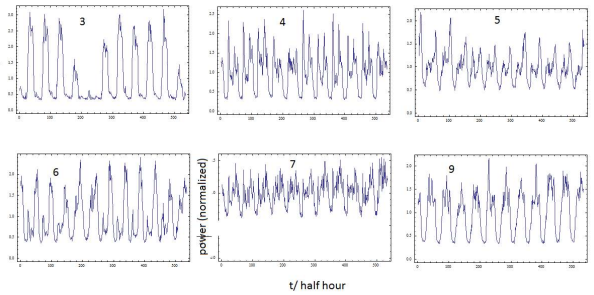


Fig. 2. The average profiles of 6 largest groups, the plots of parameters $\lambda_\alpha(t)$, $\alpha = 3, 4, 5, 6, 7, 9$. Peaks are day-time activity, weekend is shown as smaller peaks for group 3.

features, like the group 2 with quite regular time pattern consists of companies and similar sites (SME). On the contrary group 4 has quite diffuse time pattern and has mos members of households with retired persons. We made a similar analysis using k-means or k-medoids method. The result was quite poor, only two large sets were formed one with SME-type sites and the other having most of the residential. Thus the differentiation of residential sites failed. the regular decomposition method on the contrary was able to find 6 significant residential groups with different social and housing content.

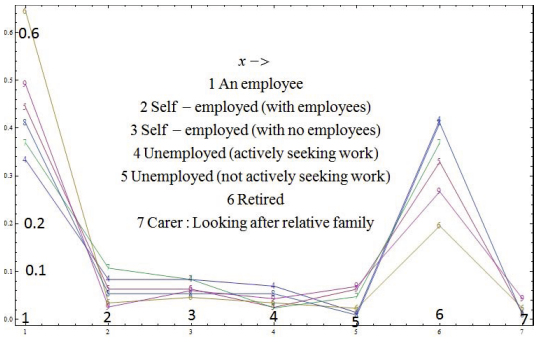


Fig. 3. Social status distributions along regular groups 4-9. A line per a group, the value is fraction of members that have a certain social status. Group 4 is like non-working household group, retired etc. Group 6 is the other extreme with highest amount of employees, that also correlates with a sharper day rhythm of electricity consumption.

4 Conclusions

We describe and demonstrate on real-life electric smart meter customer data, a novel information theoretic method for clustering multivariate time series into

few groups. The method is inspired by Szemerédi's Regularity Lemma (SRL) from graph theory. Our method compares favorable to such traditional method as k-means and k-medoids. Our method has potential in similar applications and particularly in case of emerging big data.

Acknowledgments. The authors are grateful for many useful discussions and contributions by Ilkka Norros, Tatu Koljonen, Marianna Bolla, Teemu Roos and Jari Hämäläinen. We thank ISSDA for providing the data files.

References

1. Irish social science data archive, `http://www.ucd.ie/issda/`
2. Aicher, C., Jacobs, A., Clauset, A.: Adapting the stochastic block model to edge-weighted networks, arXiv:1305.5782 [stat.ML] (2012)
3. Bolla, M.: Recognizing linear structure in noisy matrices. Linear Algebra and its Applications 402(205), 228–244 (2005)
4. Bolla, M.: Spectral Clustering and Biclustering. John Wiley and Sons (2013)
5. Frieze, A.: The regularity lemma and approximation schemes for dense problems. In: Proc. FOCS 1996 Proceedings of the 37th Annual Symposium on Foundations of Computer Science, vol. 12 (1996)
6. Grunwald, P.D.: The Minimum Descrption Length Principle. The MIT Press (2007)
7. Komlós, J., Simonovits, M.: Szemerédi's regularity lemma and its applications in graph theory. In: Miklós, D., Sós, V.T., Szonyi, T. (eds.) Combinatorics, Paul Erdős is Eighty, Budapest, pp. 295–352. János Bolyai Mathematical Society (1996)
8. Navlakha, S., Rastogi, R., Shrivastava, N.: Graph summarization with bounded error. In: ACM SIGMOD Int. Conf. on Management of Data, p. 419 (2008)
9. Nepusz, T., Négyessy, L., Tusnády, G., Bazsó, F.: Reconstructing cortial networks: case of directed graphs with high level of reciprocity. In: Bollobás, B., Kozma, R., Miklós, D. (eds.) Handbook of Large-Scale Random Networks. Bolyai Society of Mathematical Studies, vol. 18, pp. 325–368. Springer (2008)
10. Pehkonen, V., Reittu, H.: Szemerédi's-type clustering of peer-to-peer streaming system. In: Proc. Cnet 2011, San Francisco, U.S.A. (2011)
11. Peixoto, T.P.: Entropy of stochastic blockmodels ensembles. Physical Review E 85(056122) (2012)
12. Sárkozy, G., Song, F., Szemerédi, E., Trivedi, S.: A practical regularity partitioning algorithm and its application in clustering (September 28, 2012) arXiv:1209.6540v1 [math.CO]
13. Sperotto, A., Pelillo, M.: Szemerédi's regularity lemma and its applications to pairwise clustering and segmentation. In: Yuille, A.L., Zhu, S.-C., Cremers, D., Wang, Y. (eds.) EMMCVPR 2007. LNCS, vol. 4679, pp. 13–27. Springer, Heidelberg (2007)
14. Szemerédi, E.: Regular partitions of graphs. In: CNRS, Paris, pp. 399–401 (1978)
15. Cover, T.M., Thomas, J.A.: Elements of Information Theory, U.S.A, p. 542. John Wiley and Sons (1991)
16. Toivonen, H., Zhou, F., Hartikainen, A., Hinkka, A.: Compression of weighted graphs. In: Int. Conf. on Knowledge Discovery and Data Mining, p. 965 (2011)

Texture Synthesis: From Convolutional RBMs to Efficient Deterministic Algorithms

Qi Gao and Stefan Roth

Department of Computer Science, TU Darmstadt

Abstract. Probabilistic models of textures should be able to synthesize specific textural structures, prompting the use of filter-based Markov random fields (MRFs) with multi-modal potentials, or of advanced variants of restricted Boltzmann machines (RBMs). However, these complex models have practical problems, such as inefficient inference, or their large number of model parameters. We show how to train a Gaussian RBM with full-convolutional weight sharing for modeling repetitive textures. Since modeling the local mean intensities plays a key role for textures, we show that the covariance of the visible units needs to be sufficiently small – smaller than was previously known. We demonstrate state-of-the-art texture synthesis and inpainting performance with many fewer, but structured features being learned. Inspired by Gibbs sampling inference in the RBM and the small covariance of the visible units, we further propose an efficient, iterative deterministic texture inpainting method.

1 Introduction

Modeling prior knowledge on images and scenes is an important research problem in computer vision. Somewhat different challenges arise when only considering specific kinds of images or scenes, since the specific structure of the data needs to be taken into account. For example, visual textures, even though playing a large part in the composition of natural images, cannot be modeled well by directly applying the ideas for building generic image priors. The major reason is that generic image priors mainly consider the smoothness and continuity of the image, while texture models have to capture the specific textural structures.

To this end, the seminal FRAME texture model [16] uses Markov random fields (MRFs) with non-parametric, multi-modal potentials to allow for spatial structure generation. More recently, Heess *et al.* [5] suggested an MRF with parametric bi-modal potentials, which can be learned alongside the filters (features). Another class of probabilistic texture models extends restricted Boltzmann machines (RBMs) [6] toward capturing the spatial structure of textures, *e.g.* work by Kivinen *et al.* [7] and Luo *et al.* [8]. Common to these MRF- or RBM-based texture models is that they can be interpreted as a conditional Gaussian random field whose parameters are controlled by discrete latent variables. Moreover, all of them simultaneously perform regularization and generate textural structures through modeling the conditional covariance and mean, respectively. Due to their complex "mean+covariance" construction, these models are not

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 434–443, 2014.

easy to train in practice. Some compromises toward stabilization of training, *e.g.* tiled-convolutional weight-sharing [7,8], can be detrimental to the quality of the generated textures. Moreover, the relative importance of the mean vs. the covariance component of these models is unclear in light of modeling textures.

In this paper we ask how important the regularization effect from the covariance component is for modeling textures. To that end, we explore the ability and efficiency of "mean" only RBMs for modeling Brodatz texture images (`www.ux.uis.no/~tranden/brodatz.html`). We learn full-convolutional Gaussian RBMs (cGRBMs), for which the conditional Gaussian distribution has fixed identity covariance, and its mean is determined by the hidden variables. Block Gibbs sampling can be used for efficient learning and inference. Most importantly, we find that to learn good cGRBM texture models, the covariance of the visible units needs to be sufficiently small, quite a bit smaller than the typical best practice [6]. Similarly, the coefficient norms of the convolutional features must be carefully constrained during learning. Our contributions are threefold: First, our learned convolutional RBMs have several favorable properties – simplicity, efficiency, spatial invariance, and a comparatively small number of structured, more interpretable features – yet they outperform more complex state-of-the-art methods in repetitive texture synthesis and inpainting. Second, as the conditional covariance of the visibles is a diagonal matrix with small variance, we show that the "mean" units actually take the most important role in modeling textures. Third, inspired by the procedure of inference through block Gibbs sampling, we further propose an efficient deterministic method for texture inpainting based on the learned features.

Other Related Work. Hao *et al.* [4] modified high-order Gaussian gated Boltzmann machines for texture modeling, and also directly model the dependencies between two visible units. They learned 1000 features with convolutional weight sharing and achieved good texture classification performance. The performance in texture generation are less satisfactory, however, with block artifacts appearing in texture inpainting results. Efros et al. [1,2] proposed non-parametric methods for texture synthesis. In [2] the synthesized image is grown by one pixel at a time according to the best matches between its neighbors and patches from the texture. [1] instead stitches together small patches directly.

2 Texture Modeling and Mean Units

2.1 MRF Models Based on Filters and Potentials

A typical way of formulating a prior for a generic image $\mathbf{x}$ is through modeling the response to some linear filters with potential functions [3]:

$$p_{\mathrm{MRF}}(\mathbf{x};\boldsymbol{\Theta}) \propto \prod_{c,j} \phi\big(\mathbf{f}_j^{\mathrm{T}}\mathbf{x}_c;\boldsymbol{\theta}_j\big), \tag{1}$$

where $\mathbf{x}_c$ denotes the pixels of clique c, $\mathbf{f}_j$ are linear filters, and $\phi(\cdot;\boldsymbol{\theta}_j)$ are the potential functions. As the filter responses usually have heavy-tailed empirical distributions around zero, the potential functions are also chosen to be

Fig. 1. Tiled-convolutional (left) and full-convolutional (right) weight sharing. Lines converging to the hidden units (shaded) are the filters; they share their parameters when indicated using the same color or line type.

heavy-tailed (*e.g.*, Student-t). Many heavy-tailed potentials can be formulated as Gaussian scale mixtures (GSMs) [10]. For better understanding and more efficient inference, such GSM potentials allow augmenting the prior with hidden variables $\mathbf{h} = (h_{jc})_{j,c}$, one for each filter $\mathbf{f}_j$ and clique c, which represent the index of the Gaussian mixture component modeling the filter response [12]. It holds that $p_{\text{MRF}}(\mathbf{x};\boldsymbol{\Theta}) \propto \sum_{\mathbf{h}} p_{\text{MRF}}(\mathbf{x},\mathbf{h};\boldsymbol{\Theta})$. Given the hidden variables, the conditional distribution for the image is a zero-mean Gaussian

$$p_{\text{MRF}}(\mathbf{x}|\mathbf{h};\boldsymbol{\Theta}) \propto \mathcal{N}\big(\mathbf{x};\mathbf{0},\boldsymbol{\Sigma}(\mathbf{h},\boldsymbol{\Theta})\big). \tag{2}$$

As changing the hidden units only changes the conditional covariance, such basic image priors focus on modeling the covariance structure of the image, which is intuitive as they are primarily aimed at regularization.

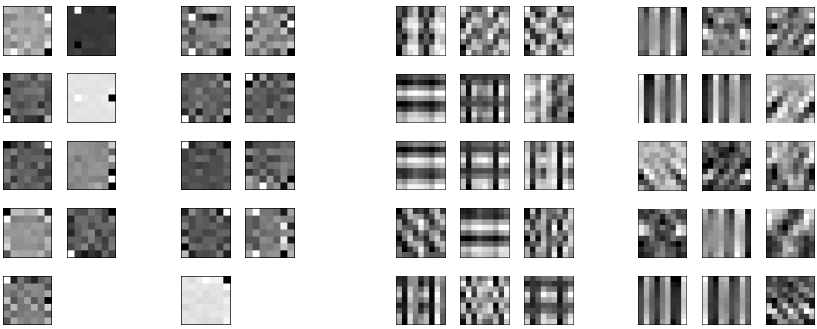
Heess *et al.* [5] showed that such generic MRF priors for natural images are not suitable for textures, and propose to extend them using bi-modal potential functions. Multi-modal potentials can also be modeled with Gaussian mixtures, however the components may no longer all have zero means. Given the hidden units, the conditional distribution of such an MRF texture model

$$p_{\text{MRFt}}(\mathbf{x}|\mathbf{h};\boldsymbol{\Theta}) \propto \mathcal{N}\big(\mathbf{x};\boldsymbol{\mu}(\mathbf{h},\boldsymbol{\Theta}),\boldsymbol{\Sigma}(\mathbf{h},\boldsymbol{\Theta})\big) \tag{3}$$

shows that bi-modal potentials capture not only the covariance structure, but also the local mean intensities. The seminal FRAME texture model [16] with its non-parametric potentials is also consistent with this observation. Comparing Eq. (3) with Eq. (2) suggests that modeling the conditional mean is a particular trait of texture models. The intuitive explanation is that the model does not "just" want to perform regularization, but instead generate textural structure.

Note that in these models the filters are applied convolutionally across the entire image. Since filters can be understood as weights connecting visible and hidden units, this is called convolutional weight sharing (*cf.* Fig. 1). Importantly, this keeps the number of parameters manageable, even on images of an arbitrary size, and also gives rise to the model's spatial invariance.

Nonetheless, learning such a "mean+covariance" model is difficult in practice, since the hidden units affect both conditional mean and covariance in complex ways. Since the filters need to be sufficiently large to generate coherent structures, the resulting covariance matrix will furthermore be quite dense, making both learning and inference rather inefficient. Moreover, the learned texture filters from [5] lack clear structure (see Fig. 2), making them difficult to interpret.



(a) D21, BiFoE (b) D53, BiFoE (c) D21, our cGRBM (d) D53, our cGRBM

Fig. 2. Comparison of learned texture filters/features

2.2 Boltzmann Machine Models

Models derived from restricted Boltzmann machines (RBMs) take a different route. A Gaussian RBM [6] models an image by defining an energy function over visible units $\mathbf{x}$ (here, the image pixels) and binary hidden units $\mathbf{h}$. The random variables have a Boltzmann distribution $p_{\mathrm{RBM}}(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \exp\{-E_{\mathrm{RBM}}(\mathbf{x}, \mathbf{h})\}$, where Z is the partition function. Gaussian RBMs have the property that the conditional distribution of the visible units given the hidden ones is a Gaussian

$$p_{\mathrm{RBM}}(\mathbf{x}|\mathbf{h}; \boldsymbol{\Theta}) \propto \mathcal{N}\big(\mathbf{x}; \boldsymbol{\mu}(\mathbf{h}, \boldsymbol{\Theta}), \boldsymbol{\Sigma}\big), \tag{4}$$

in which only the conditional mean depends on the hidden variables.

More recent variants of Boltzmann machines for texture modeling [7,8] not only model the conditional mean, but also the covariance akin to Eq. (3). [7] experimentally compared three models: Gaussian RBMs, Products of Student-t (PoT) [15], and their combination, which corresponds to modeling conditional mean, covariance, and mean+covariance, respectively. The results revealed the importance of the conditional mean for texture synthesis and inpainting.

Note that both [7,8] adopt tiled-convolutional weight sharing [11] (*cf*. Fig. 1). The apparent reason is that the states of hidden units are less correlated, thus making training of the models easier. Unfortunately, tiled-convolutional models involve many parameters, since several sets of features (filters) must be learned. For example, [7,8] learn and use more than 300 features for every texture, which are moreover not spatially invariant. Consequently, tiled-convolutional weight sharing requires copious training data, which for textures is often not available.

3 Learning Full-Convolutional RBMs for Textures

Mean units appear to be an important component of many texture models. As these models also include covariance units and/or complex weight sharing, it is not clear how important the mean units are. We now investigate this and explore the capability of "mean-only" Gaussian RBMs for textures.

3.1 Convolutional Gaussian RBM

A spatially invariant model is obtained through applying the Gaussian RBM convolutionally to all overlapping cliques of a large texture image. The energy function of the convolutional Gaussian RBM (cGRBM) is then written as

$$E_{\text{cGRBM}}(\mathbf{x}, \mathbf{h}) = \frac{1}{2\gamma}\mathbf{x}^{\mathrm{T}}\mathbf{x} - \sum_{c,j} h_{jc}\big(\mathbf{w}_j^{\mathrm{T}}\mathbf{x}_c + b_j\big), \tag{5}$$

where we add a weight γ to the quadratic term. Here $\mathbf{w}_j$ determine the interaction between pairs of visible units $\mathbf{x}_c$ and hidden units h_{jc}. Thus $\mathbf{w}_j$ are the features or filters, b_j are the biases, c and j are indices for all overlapping image cliques and filters, respectively. The conditional distribution of $\mathbf{x}$ given $\mathbf{h}$ is a Gaussian

$$p_{\text{cGRBM}}(\mathbf{x}|\mathbf{h}) \propto \mathcal{N}\bigg(\mathbf{x}; \gamma \sum_{c,j} h_{jc}\mathbf{w}_{jc}, \gamma\mathbf{I}\bigg), \tag{6}$$

where the vector $\mathbf{w}_{jc}$ is defined as $\mathbf{w}_{jc}^{\mathrm{T}}\mathbf{x} = \mathbf{w}_j^{\mathrm{T}}\mathbf{x}_c$. The conditional of $\mathbf{h}$ given $\mathbf{x}$ is a simple logistic sigmoid function

$$p_{\text{cGRBM}}(h_{jc}|\mathbf{x}) \propto \text{logsig}(\mathbf{w}_{jc}^{\mathrm{T}}\mathbf{x} + b_j). \tag{7}$$

3.2 Data

For a fair comparison with other models [5,7,8], we follow their use of the Brodatz texture images for training and testing our models. The images are rescaled to either 480×480 or 320×320, while preserving the major texture features, and then are normalized to 0-mean and unit standard deviation. We also divide each image into a top half for training and a bottom half for testing.

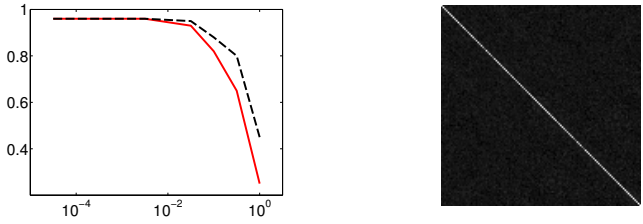
3.3 Learning

As the partition function of the model is intractable, we perform approximate maximum likelihood (ML) learning based on persistent contrastive divergence (PCD) [13]. Model samples are obtained using standard, efficient block Gibbs sampling, which alternately samples the visibles $\mathbf{x}$ or the hiddens $\mathbf{h}$ given the other. The parameters are updated using gradient ascent, *e.g.* for filters using

$$\mathbf{w}_j^{(t+1)} = \mathbf{w}_j^{(t)} + \eta\Big[\big\langle \tfrac{\partial E(\mathbf{x})}{\partial \mathbf{w}_j}\big\rangle_{\mathbf{X}^{\text{PCD}}} - \big\langle \tfrac{\partial E(\mathbf{x})}{\partial \mathbf{w}_j}\big\rangle_{\mathbf{X}^{\mathbf{0}}}\Big], \tag{8}$$

where η is the learning rate, $\langle\cdot\rangle$ denotes the average over the training data $\mathbf{X}^{\mathbf{0}}$ or the samples $\mathbf{X}^{\text{PCD}}$.

The standard learning procedure [6], however, does not ensure that a good convolutional RBM texture model is learned in practice. *E.g.* even the simple mean-only RBM baseline of [7] stabilizes learning using tiled-convolutional



(a) (b)

Fig. 3. *(a)* Texture similarity scores (TSS) of synthesized textures (black, dashed) and model samples (red, solid) *vs*. the choice of γ. *(b)* The covariance matrix of 1000 samples of $\mathbf{h}$ when $\gamma = 0.03$. Results are based on D21.

weight sharing and a slowly mixing Hybrid Monto Carlo (HMC) sampler. Consequently, care must be taken to be able to train a cGRBM for textures.

Choice of Parameter γ. The typical best practice in Gaussian RBMs is to set the weight γ to 1 when the training data is normalized [6]. But we find that $\gamma = 1$ is far from the optimum and its value can greatly affect the generative properties of the trained texture models. Fig. 3(a) shows how γ changes the texture similarity score (TSS) [5] (see Sec. 3.4 for details) of model samples and synthesized textures. Actually, since a texture sample drawn with the Gibbs sampler is a sum of the conditional mean and *i.i.d.* Gaussian noise, $\gamma = 1$ will lead to the textural structures being dominated by noise. But even if we synthesize textures by taking the conditional mean of the final sampling iteration, as we do here, we see that γ can greatly affect the quality of the texture and that the previous best practice of $\gamma = 1$ does not work well. This may be the reason why other texture models considered more complex pixel covariances and/or rely on less well-mixing samplers for stabilizing training.

Although Fig. 3(a) suggests that smaller values of γ should be preferred, an overly small γ will lead to a small covariance for the Gaussian in Eq. (3) and consequently to slow mixing of the Gibbs sampler. We find $\gamma = 0.03$ to be a good trade-off. To illustrate this, Fig. 3(b) shows the covariance matrix computed from 1000 consecutive samples of $\mathbf{h}$ corresponding to one feature. As it is close to a diagonal matrix, the variables in $\mathbf{h}$ are approximately independent, thus the sampler mixes well. We use $\gamma = 0.03$ for all our experiments.

Other Best Practices. To obtain structured filters and – in our experience – also better texture models, we have to impose some constraints on the filters. As usual, the filters are initialized with random values, but their coefficient norms are ensured to be small initially. During training they are moreover constrained to have 0-mean and limited to not increase above an empirical threshold of $0.05/\gamma$. Otherwise the filters will often get stuck in poor local optima without any clear structure. Since the biases do not change significantly during learning, we fix them to $b_j = -\frac{1}{3}\|\mathbf{w}_j\|$, similar to [9]. The bias depends on the current norm of filter coefficients to keep a reasonable portion of the hidden units being "on" (*cf*. Eq. 7).

Also note that the typical whitening of the training data cannot be applied for textures, even if it is common for natural image priors. Since whitening will remove the major structural pattern of a single texture, it is in our experience difficult for the RBM to represent the remaining spatial patterns.

The Learned Models. We trained cGRBM models for several Brodatz textures, each of which is trained based on 40 patches of size 76×76, randomly cropped from the corresponding preprocessed training image. As all the training images are rescaled, we simply fix the filter size to 9×9 for all models. The models for textures D6, D21 and D53 consist of 15 learned filters, while the model for D77 has 20 filters due to its slightly more complex pattern. Examples of the learned filters are shown in Fig. 2; we observe clearly apparent structure, *e.g.* unlike [5].

3.4 Generative Properties

To evaluate the generative performance of our learned cGRBM texture models, we quantitatively compute texture similarity scores (TSS) [5] between the synthesized textures and real texture patches, which is defined based on the maximum normalized cross correlation $\mathrm{TSS}(\mathbf{s}, \mathbf{x}) = \max_i \frac{\mathbf{s}^{\mathrm{T}} \mathbf{x}_{(i)}}{\|\mathbf{s}\| \cdot \|\mathbf{x}_{(i)}\|}$, where $\mathbf{s}$ is the synthesized texture and $\mathbf{x}_{(i)}$ denotes the patch of the same size as $\mathbf{s}$ within the texture image $\mathbf{x}$, at location i.

We collect 100 samples of size 76×76 for each model (each texture) using Gibbs sampling. Since in Gibbs sampling the texture samples are obtained by summing the final conditional mean and *i.i.d.* Gaussian noise, we use the conditional means from the last sampling step as the synthesized texture. For computing the TSS, only the center 19×19 pixels are considered (the same size as in [5,7,8]). Tab. 1 shows the means and standard deviations of TSS. Thanks to the full-convolutional weight sharing scheme, our simple cGRBM models only require 15 (or 20 for D77) features (filters) to exceed the generative properties of much more complex Boltzmann machine models [7,8] with many more (> 300) features. Note the considerable performance difference between our learned cGRBMs and the Gaussian RBM baseline "Tm" of [7], which is based on a standard learning procedure and tiled-convolutional weight sharing. Our cGRBMs even outperform the deep belief networks (DBNs) of [8]. Meanwhile, the 9×9 filter size of our models is also smaller than that of [7,8] (11×11). The BiFoE models [5] only use 9 filters of size 7×7, but the paper argues that more and larger filters do not lead to a large difference in model quality, but greatly reduce the efficiency of inference.

3.5 Texture Inpainting

In a texture inpainting application, following previous work [7,8], we take 76×76 patches from the testing texture images and create a 54×54 square hole in the middle of each patch by setting the intensity values to zero. The task is to generate texture in the square hole that is consistent with the given boundary.

Table 1. Means and standard deviations of TSS of the synthesized textures

Model	D6	D21	D53	D77
BiFoE [5]	0.757 ± 0.059	0.871 ± 0.032	0.827 ± 0.087	0.646 ± 0.022
Tm [7]	0.930 ± 0.021	0.890 ± 0.079	0.849 ± 0.061	0.866 ± 0.008
TmPoT [7]	0.933 ± 0.036	0.896 ± 0.070	0.853 ± 0.056	0.870 ± 0.008
TssRBM [8]	0.937 ± 0.047	0.948 ± 0.025	0.941 ± 0.022	0.841 ± 0.012
DBN [8]	0.952 ± 0.016	0.947 ± 0.032	0.950 ± 0.026	0.864 ± 0.160
cGRBM (ours)	$\mathbf{0.963} \pm 0.005$	$\mathbf{0.961} \pm 0.008$	$\mathbf{0.965} \pm 0.004$	$\mathbf{0.875} \pm 0.013$

Table 2. Means and standard deviations of MSSIM scores of the inpainted textures

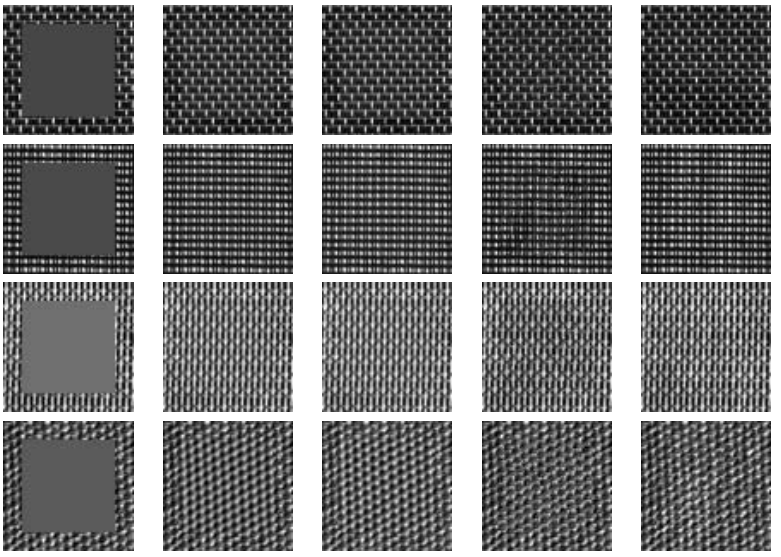
Model	D6	D21	D53	D77
Tm [7]	0.858 ± 0.016	0.866 ± 0.019	0.849 ± 0.023	0.764 ± 0.027
TmPoT [7]	0.863 ± 0.018	0.874 ± 0.012	0.860 ± 0.023	0.767 ± 0.032
TssRBM [8]	0.888 ± 0.023	0.912 ± 0.014	0.916 ± 0.024	0.763 ± 0.031
DBN [8]	0.889 ± 0.025	0.906 ± 0.016	0.924 ± 0.029	0.774 ± 0.023
cGRBM (ours)	$\mathbf{0.909} \pm 0.017$	$\mathbf{0.928} \pm 0.012$	$\mathbf{0.933} \pm 0.010$	$\mathbf{0.783} \pm 0.027$
Efros & Leung [2]	0.827 ± 0.028	0.801 ± 0.029	0.863 ± 0.018	0.632 ± 0.041
Deterministic (ours)	0.899 ± 0.019	0.918 ± 0.014	0.926 ± 0.016	0.775 ± 0.034

Inpainting is done through sampling conditioned on the given boundaries. This procedure is quite efficient when using a block Gibbs sampler. For each texture, we use 20 different inpainting frames and perform inpainting 5 times with different initializations, leading to 100 inpainting results. The quality of the inpainted texture is measured by the mean structural similarity (MSSIM) score [14] between the inpainted region and the ground truth. Fig. 4 shows examples of inpainting results and Tab. 2 gives a quantitative comparison with other models[1], which we outperform considerably despite a simpler model architecture.

4 Deterministic Texture Synthesizer

From Sec. 3 we know that the value for γ in our cGRBM model is small. Looking at Eq. (6), this on the one hand means that the sample will not deviate significantly from the conditional mean. Moreover, the norms of filter coefficients must be large to balance the small γ, which implies that most values of $\mathbf{w}_{jc}^{\mathrm{T}}\mathbf{x} + b_j$ will fall outside of the smooth transition area of the logistic sigmoid in Eq. (7). This suggests that, in applications, it may be possible to use deterministic functions to replace sampling the two conditionals. In particular, we apply a unit step function on $\mathbf{w}_{jc}^{\mathrm{T}}\mathbf{x} + b_j$, then use the obtained binary auxiliary variable to modulate the filters to reconstruct the image, and repeat the procedures until convergence (Alg. 1). Note that this is equivalent to a block coordinate descent on the model energy Eq. (5). Since this scheme only works well if some reference pixels are given, such as in texture inpainting, we use it in this context. While slightly worse than sampling the cGRBM, the performance of the deterministic approach is still

[1] Our implementation of Efros & Leung [2] uses a window size of 15×15.



(a) Inpainting frames (b) Our cGRBM (c) Our deterministic (d) Efros & Leung [2] (e) Ground truth

Fig. 4. Examples of inpainting results. From top to bottom: D6, D21, D53, D77.

better than the state of the art. In our inpainting experiment, our deterministic method only needed 30–50 iterations to reach convergence, while sampling the cGRBM usually required ~ 100 iterations. It is moreover quite efficient, because the computation in each iteration is very simple. By contrast, nonparametric methods (*e.g.* [2]) are often not as efficient due to the necessary matching step.

5 Summary

In this paper we analyzed the role of the conditional mean in modeling visual repetitive textures. We showed that simple Gaussian RBMs trained in a convolutional fashion are able to outperform much more complex state-of-the-art texture models, in which the latent variables also control the conditional covariance. We showed that the covariance of the cGRBM must actually be rather

Algorithm 1. Deterministic Texture Inpainting

Require: Image $\mathbf{x}$ to be inpainted
  repeat
    $h_{jc} \leftarrow H(\mathbf{w}_{jc}^{\mathrm{T}}\mathbf{x} + b_j)$ ▷ where H is a unit step function
    $\mathbf{x} \leftarrow \gamma \sum_{j,c} h_{jc}\mathbf{w}_{jc}$
  until no change of $\mathbf{x}$ (or $\mathbf{h}$)
  return $\mathbf{x}$

small to enable high texture quality, and suggest new best practices for modeling repetitive textures with RBMs. Our model requires only a small number of learned features, with a clearly emerging structure, and is spatially invariant. Inspired by efficient RBM inference using block Gibbs sampling, we further propose a fast, iterative deterministic texture synthesis method.

An open question for future work is to automatically determine the number of filters based on the complexity of the respective texture. Moreover, although a mean-only model can capture repetitive textures well, covariances might have to be considered for general textures. We leave this for future work.

References

1. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: SIGGRAPH 2001 (2001)
2. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: ICCV 1999, vol. 2, pp. 1033–1038 (1999)
3. Geman, D., Reynolds, G.: Constrained restoration and the recovery of discontinuities. IEEE T. Pattern Anal. Mach. Intell. 14(3), 367–383 (1992)
4. Hao, T., Raiko, T., Ilin, A., Karhunen, J.: Gated Boltzmann Machine in Texture Modeling. In: Villa, A.E.P., Duch, W., Érdi, P., Masulli, F., Palm, G. (eds.) ICANN 2012, Part II. LNCS, vol. 7553, pp. 124–131. Springer, Heidelberg (2012)
5. Heess, N., Williams, C.K.I., Hinton, G.E.: Learning generative texture models with extended Fields-of-Experts. In: BMVC 2009 (2009)
6. Hinton, G.: A practical guide to training restricted Boltzmann machines. Tech. Rep. UTML TR 2010–003, University of Toronto (2010)
7. Kivinen, J.J., Williams, C.K.I.: Multiple texture Boltzmann machines. In: AISTATS (2012)
8. Luo, H., Carrier, P.L., Courville, A., Bengio, Y.: Texture modeling with convolutional spike-and-slab RBMs and deep extensions. In: AISTATS (2013)
9. Norouzi, M., Ranjbar, M., Mori, G.: Stacks of convolutional restricted Boltzmann machines for shift-invariant feature learning. In: CVPR 2009 (2009)
10. Portilla, J., Strela, V., Wainwright, M.J., Simoncelli, E.P.: Image denoising using scale mixtures of Gaussians in the wavelet domain. IEEE T. Image Process. 12(11), 1338–1351 (2003)
11. Ranzato, M., Mnih, V., Hinton, G.E.: Generating more realistic images using gated MRF's. In: NIPS 2010 (2010)
12. Schmidt, U., Gao, Q., Roth, S.: A generative perspective on MRFs in low-level vision. In: CVPR 2010 (2010)
13. Tieleman, T.: Training restricted Boltzmann machines using approximations to the likelihood gradient. In: ICML 2008 (2008)
14. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE T. Image Process. 13(4), 600–612 (2004)
15. Welling, M., Hinton, G.E., Osindero, S.: Learning sparse topographic representations with products of Student-t distributions. In: NIPS 2002, pp. 1359–1366 (2002)
16. Zhu, S.C., Wu, Y., Mumford, D.: Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. Int. J. Comput. Vision 27(2), 107–126 (1998)

Improved Object Matching Using Structural Relations

Estephan Dazzi[1,2,⋆], Teofilo de Campos[2], and Roberto M. Cesar Jr.[1]

[1] Instituto de Matemática e Estatística - IME,
Universidade de São Paulo - USP, São Paulo, Brasil
[2] CVSSP, University of Surrey, Guildford, GU2 7XH, UK
{estephan.dazzi,roberto.cesar}@vision.ime.usp.br,
t.decampos@st-annes.oxon.org

Abstract. This paper presents a method for object matching that uses local graphs called keygraphs instead of simple keypoints. A novel method to compare keygraphs was proposed in order to exploit their local structural information, producing better local matches. This speeds up an object matching pipeline, particularly using RANSAC, because each keygraph match contains enough information to produce a pose hypothesis, significantly reducing the number of local matches required for object matching and pose estimation. The experimental results show that a higher accuracy was achieved with this approach.

Keywords: Local feature matching, SIFT, hierarchical k-means tree, RANSAC, graph-based structural information.

1 Introduction

An important problem in computer vision is object recognition through matching, which consists in localising objects in test images and estimating their 3D pose. Solutions to this problem are useful in different application domains, such as robotics, medical images analysis and augmented reality. One of the most succesful approaches for this problem involves establishing correspondences between interest points (keypoints) in test and training images; next, a pose estimation algorithm is used, e.g. based on RANSAC, which operates by removing outliers that do not conform with global pose parameters. In this paper, we present a novel method that is capable of providing more accurate solutions besides being computationally cheaper. Our method is generic, in the sense that it can be used with any keypoint extractor method; we validate it using SIFT features [1].

In keypoint-based object recognition, point-to-point correspondences are obtained by matching discriminative features and reducing the set of matches in a

⋆ We would like to thank FAPESP (grant 2011/50761-2), CNPq, CAPES (process 14745/13-5) and NAP eScience - PRP - USP for the support. During most of the production of this paper, T. de Campos had been working in Neil Lawrence's group at the DCS, University of Sheffield, 211 Portobello, Sheffield, S1 4DP, UK.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 444–453, 2014.

post-processing step. For instance, the ratio test [1] compares the distances to the first and the second nearest neighbor and only establishes a match if the former is significantly smaller than the latter. However, usually there are locations on manmade objects that have similar local appearances, thus discriminative matching may prevent features with similar descriptors from being matched, which becomes a problem particularly when matching keypoints coming from different images.

In our work, initially, we also produce matches solely based on photometric information, but we allow a large number of matches to be established. Then, aiming to eliminate most of the incorrect matches, we use *structural information* within the images; this is done by establishing matches between small sets of keypoints, which we treat as graphs. In this way, our method produces a better set of keypoint matches, even when there are locations on the objects with similar appearances. Not only those matches have a high probability of being correct, but this also benefits the next stage, based on RANSAC, which ends up using a small set of graph correspondences.

Differently from previous approaches that model an image as a global graph and then proceed by employing graph matching methods, such as the work of Sirmacek and Unsalan [2] which uses SIFT keypoints as graph vertices or the work of McAuley and Caetano [3], our approach is local: we decompose the scene and pattern into collections of local graphs and perform only local graph matching, leaving the global matching to the RANSAC procedure. We built upon insights from the work of Morimitsu et al. [4], which focused on fast object detection using a single training image. For that, they used a graph edge descriptor based on Fourier transform and explicitly stored several structures obtained from the training image, which are matched to similar structures found in the test image. In the present paper, we focus on an object recognition task in which there are several images per training object and also many objects stored. We use a more discriminative keypoint extractor (SIFT), and since it is not computationally feasible to explicitly store structures found in the (many) training images, we develop a strategy based on quickly evaluating, during execution time, different aspects of strucures within test and training images.

2 Methodology

The first step of our object recognition process involves extracting SIFT keypoints from all the training images. We use the ground-truth segmentation to eliminate keypoints that are not on the object. We store all the training keypoints in a global indexing structure, which allows to quickly find the approximate nearest neighbors of a query (test) keypoint. We chose to use the hierarchical k-means tree proposed by Muja and Lowe [5] due to its efficiency. For each SIFT keypoint extracted from a training image we store its normalized descriptor (a 128-D feature vector), its scale, its orientation, an identifier of its source image and its x, y position in that image.

Matching of a test object is done following a pipeline of three stages. First, photometric information is used: each SIFT descriptor of the test image runs

through the hierarchical k-means tree, producing many matches to the keypoints of the training images. In the second stage, most of the incorrect keypoint matches are eliminated using structural information within images. The strategy consists in substituting the matches previously established between one-to-one keypoints by matches established between small sets of keypoints, i.e., graphs, called *keygraphs* [4]. A keygraph is a graph whose vertices are keypoints, and whose edges carry structural information about its keypoints. The third stage of the matching process consists in using a modified RANSAC (Random Sample Consensus) algorithm, which employs matches established between keygraphs.

2.1 Keypoint Matching

SIFT keypoints are often located very close to each other and this can lead to poor pose estimation results with minimal sets. We select a maximal subset $\mathcal{S}$ of keypoints in the test image such that the distance, in pixels, between any two keypoints in $\mathcal{S}$ is above a threshold d_{pix}; we use $d_{pix} = 10$ pixels.

After selecting the set $\mathcal{S}$ of keypoints in the test image, we match them to the keypoints of the training images, which are stored in a hierarchical k-means tree. We let each test keypoint establish a match with at most *two* keypoints of *each* training image. In order to establish a match between keypoints, it is necessary that the Euclidean distance between their (normalized) SIFT descriptors is below a threshold t; we set t with a relatively high value, as the next stage eliminates possibly incorrect matches. If a test keypoint can establish more than two matches with a same training image, only the two closest matches are kept.

2.2 Keygraph Matching

A *keygraph* is defined as a graph $G = (V, E)$, where the vertex set V is composed of keypoints, and E is the set of graph edges. All the keypoints in a keygraph are present in the same image. Every keygraph has the same number of vertices, κ, and it consists in an *oriented circuit in the clockwise direction*, $G = (v_1, v_2, \ldots, v_\kappa)$.

Each keygraph in the test image can establish matches with keygraphs in every training image. Let $G = (v_1, v_2, \ldots, v_\kappa)$ and $H = (w_1, w_2, \ldots, w_\kappa)$ be keygraphs in a test and in a training image, respectively. The existence of a match between G and H, denoted as $\mathcal{M} = (G, H)$, implies κ matches between the keypoints (vertices) of G and H. For instance, (G, H) may imply the set of keypoints matches $\mathcal{M} = \{(v_1, w_1), (v_2, w_2), \ldots, (v_\kappa, w_\kappa)\}$, i.e., it implies the occurrence of κ matches between pairs of keypoints.

Obtaining Keygraphs in the Test Image. We begin with the subset $\mathcal{S}$ of keypoints in the test image and execute the Delaunay Triangulation, generating a set of triangles, i.e. we use keygraphs with $\kappa = 3$ vertices, $G = (v_1, v_2, v_3)$, represented as triangles whose edges are oriented in the clockwise direction.

Obtaining Keygraphs in the Training Images The keygraphs in the training images are not obtained using the Delaunay Triangulation. Instead, we first calculate the potential keygraph matches that may occur from the test image to each training image. Then we analyse which of those potential keygraph matches imply a valid keygraph in the training image.

Let $G = (v_1, v_2, v_3)$ be a keygraph in a test image, obtained using the Delauney Triangulation. For each training image, we verify whether G establishes keygraph matches with that image. As an illustration, consider the case in which every keypoint of G, v_1, v_2 and v_3, establishes two matches with keypoints of a same training image; then there are *eight* different possible matches between G and keygraphs of that training image: choose one of the two matches of v_1 *and* choose one of the two matches of v_2 *and* choose one of the two matches of v_3. Considering that the keypoint matches are (v_1, w_1), (v_1, w_2), (v_2, w_3), (v_2, w_4), (v_3, w_5) and (v_3, w_6), at most eight sets of keypoint matches (i.e. keygraph matches) can be established:

$$\mathcal{M}_1 = \{(v_1, w_1), (v_2, w_3), (v_3, w_5)\},\ \mathcal{M}_2 = \{(v_1, w_1), (v_2, w_3), (v_3, w_6)\},$$
$$\mathcal{M}_3 = \{(v_1, w_1), (v_2, w_4), (v_3, w_5)\},\ \mathcal{M}_4 = \{(v_1, w_1), (v_2, w_4), (v_3, w_6)\},$$
$$\mathcal{M}_5 = \{(v_1, w_2), (v_2, w_3), (v_3, w_5)\},\ \mathcal{M}_6 = \{(v_1, w_2), (v_2, w_3), (v_3, w_6)\},$$
$$\mathcal{M}_7 = \{(v_1, w_2), (v_2, w_4), (v_3, w_5)\} \text{ and } \mathcal{M}_8 = \{(v_1, w_2), (v_2, w_4), (v_3, w_6)\}.$$

Each one of those keygraph matches requires the existence of a specific keygraph in the training image; for instance, $\mathcal{M}_1 = \{(v_1, w_1), (v_2, w_3), (v_3, w_5)\}$ requires $H_1 = (w_1, w_3, w_5)$ in the training image. As we assume that mirroring is not a possible distortion of the test image, the circuit of a keygraph H in a training image must be oriented in the clockwise direction; if it is oriented in the counter-clockwise direction then H and the tentative keygraph match involving H are not accepted[1].The set of possible keygraph matches $\mathcal{M}_1$, $\mathcal{M}_2$, ..., $\mathcal{M}_8$ established by a test keygraph $G = (v_1, v_2, v_3)$ with a training image can also be reduced if v_1, v_2 or v_3 establish fewer keypoint matches with that image; naturally, this set becomes empty if v_1, v_2 or v_3 does not establish any match or none of the implied circuits is in the clockwise direction.

Discarding Keygraph Matches Using Structural Relations. After obtaining a set of (at most eight) tentative keygraph matches between a test keygraph G and keygraphs in a training image, we use five additional tests aiming to eliminate incorrect keygraph matches. This is very effective: the total number of keygraph matches is reduced in orders of magnitude. The tests are based on photometric and structural information within the keygraphs.

To illustrate the tests, let $G = (v_1, v_2, v_3)$ be a keygraph in the test image and $\mathcal{M} = \{(v_1, w_1), (v_2, w_2), (v_3, w_3)\}$ be a tentative keygraph match implied by G in a training image, such that $\mathcal{M}$ requires the existence of the keygraph $H = (w_1, w_2, w_3)$ in that training image.

[1] We assume that training objects are convex or that there is, in the training set, at least one viewpoint of the object where this assumption is valid.

The first test is based on the edges of a training keygraph. In $H = (w_1, w_2, w_3)$, there are three edges: $e^H_{1,2} = (w_1, w_2)$, $e^H_{2,3} = (w_2, w_3)$ and $e^H_{3,1} = (w_3, w_1)$, whose lengths in pixels in the training image are denoted as, respectively, $|e^H_{1,2}|$, $|e^H_{2,3}|$ and $|e^H_{3,1}|$ (an edge is a straight line connecting two vertices). This first test verifies whether the edges length respect a minimum and a maximum value: we use 10 and 100 pixels, i.e. this test verifies whether $10 \leq |e^H_{i,j}| \leq 100$. As the keygraphs in the test image in general have edges with a length equal to or slightly greater than $d_{pix} = 10$ pixels, this test allows the objects to appear in the test image considerably smaller than the (large) object in the training image.

The second test is based on the ratio of edges length in corresponding keygraphs. Considering the tentative match $\mathcal{M}$ between the keygraphs $G = (v_1, v_2, v_3)$ and $H = (w_1, w_2, w_3)$, the three ratios between the length of corresponding edges are $r_{ij} = |e^G_{i,j}|/|e^H_{i,j}|$. This test verifies whether the larger ratio, r_{ij}, is at most twice the smaller one, r_{kl}, i.e. $r_{ij} \leq 2r_{kl}$. This test still allows the occurrence of a large variation between the viewpoints of the object in the test and the training images, but since many training images are taken around an object, a very drastic viewpoint change is not allowed to occur.

The third test is based on the ratio of the scale of corresponding SIFT keypoints. The motivation of this third test is similar to that of the second one. In the test keygraph $G = (v_1, v_2, v_3)$, consider that the scale of the SIFT keypoint v_1 is s^G_1 and similarly we have the scales s^G_2 for v_2 and s^G_3. In a similar way, for the training keygraph $H = (w_1, w_2, w_3)$ we have the scales s^H_1, s^H_2 and s^H_3. Thus the three ratios between the scale of corresponding keypoints are $r_1 = s^G_1/s^H_1$, $r_2 = s^G_2/s^H_2$ and $r_3 = s^G_3/s^H_3$. Similarly to the second test, this third test verifies whether the larger ratio is at most twice the smaller one.

The fourth test is based on both edges and scales. It uses results calculated in the second and the third tests: the ratios between edges length r_{12}, r_{23} and r_{31} and the ratios between scales r_1, r_2 and r_3. Ideally, the value $E = r_{12} + r_{23} + r_{31}$ would be similar to the value $S = r_1 + r_2 + r_3$, as the change in the object size and viewpoint from the training image to the test image should impact similarly the edges length and the SIFT scale. However, as imprecisions can occur, we let the values E and S differ: this fourth test verifies whether the larger value is at most 50% greater than the smaller value, i.e., if $E > S$ this test verifies whether $E \leq 1.5S$ and if $S > E$ it verifies whether $S \leq 1.5E$.

The fifth test uses the orientation (angle) of SIFT keypoints. One of the three pairs of matched keypoints is selected, and the variation of angle between the test and the training keypoints is calculated; then, for the other two keypoint pairs, this variation is applied and it is verified whether the resulting angle is within a margin of error of 45 degrees from the original SIFT orientation. The test suceeds if both keypoint pairs agree with the angle variation implied by the first keypoint pair. If the test fails using a keypoint pair to calcule the angle variation, it can be evaluated again using the other two keypoint pairs to calculate the angle variation: it must suceed for at least one of the three pairs. For example, in the tentative match $\mathcal{M}$ of keygraphs $G = (v_1, v_2, v_3)$ and $H = (w_1, w_2, w_3)$, the pair (v_1, w_1) is used to calculate the angle variation. The angle of v_1 is 0° and the

angle of w_1 is 30°, i.e. from v_1 to w_1 occurs an increasing of 30°. Now, this angle variation (+30°) is verified with the other two pairs of keypoints. The angles of v_2 and w_2 are, respectively, 40° and 80°; applying the variation of +30°, we obtain that the angle of w_2 should be 70° (40° plus 30°), which is within the margin of error of 45°, as the true orientation of w_2, 80°, is just 10° above the 70° implied by the first keypoint pair. A similar verification is made by applying the variation of +30° to the pair (v_3, w_3). The whole evaluation can also be made using the angle variation from v_2 to w_2 or the angle variation from v_3 to w_3. We use a large margin of error of 45 degrees which allows the occurrence of imprecisions but still avoids the establishment of absurd keygraph matches.

Figure 2.2 illustrates the establishment of keygraph correspondences.

2.3 Third Stage: RANSAC on Keygraphs

One keygraph match generates $\kappa = 3$ keypoint matches. In the experiments in this paper, we use an affine transformation to instantiate an object, thus *one* keygraph match is necessary to instantiate an affine transformation. Compared to the tradicional RANSAC approach, which would require the random selection of three independent keypoint matches, the keygraph method requires the verification of a smaller number of poses.

Let $\mathcal{G}$ be the set of all keygraph matches between the test image and a training image, $\mathcal{G} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_{|\mathcal{G}|}\}$, in which $\mathcal{M}_i$ is a set of three keypoint matches; thus the set $\mathcal{P}$ of *keypoint* matches between those images is $\mathcal{P} = \mathcal{M}_1 \cup \mathcal{M}_2 \cup \ldots \cup \mathcal{M}_{|\mathcal{G}|}$. To evaluate the quality of an affine transformation which instantiates, in the test image, the object present in that training image, we count the number of keypoint matches that agree with it: for each keypoint in the training image, let x, y be its position in the test image as established by the keypoint match, and let x', y' be its position in the test image as predicted by the affine transformation under evaluation. If the distance between x, y and x', y' is below three pixels, we consider that this keypoint match agrees with the transformation. If at least six keypoint matches agree with a transformation (i.e. the three matches used to instantiate it plus three other matches), we consider that a correct pose of the object is found, and the algorithm returns this affine transformation. If more than one solution is found for a test image, the algorithm returns the one with more matches agreeing with it.

3 Experiments and Results

In our experiments we use a challenging object recognition dataset which contains ten different types of common household objects. For each object type, there are 25 training images taken around the object and 50 test images in which the object appears in a cluttered, realistic scene (in half of them there is one object instance, in the other half, two instances). This dataset was produced and made available by Hsiao et al. [6]. The authors evaluated it in a 3D object recognition task, in which a 3D model was created for each training object.

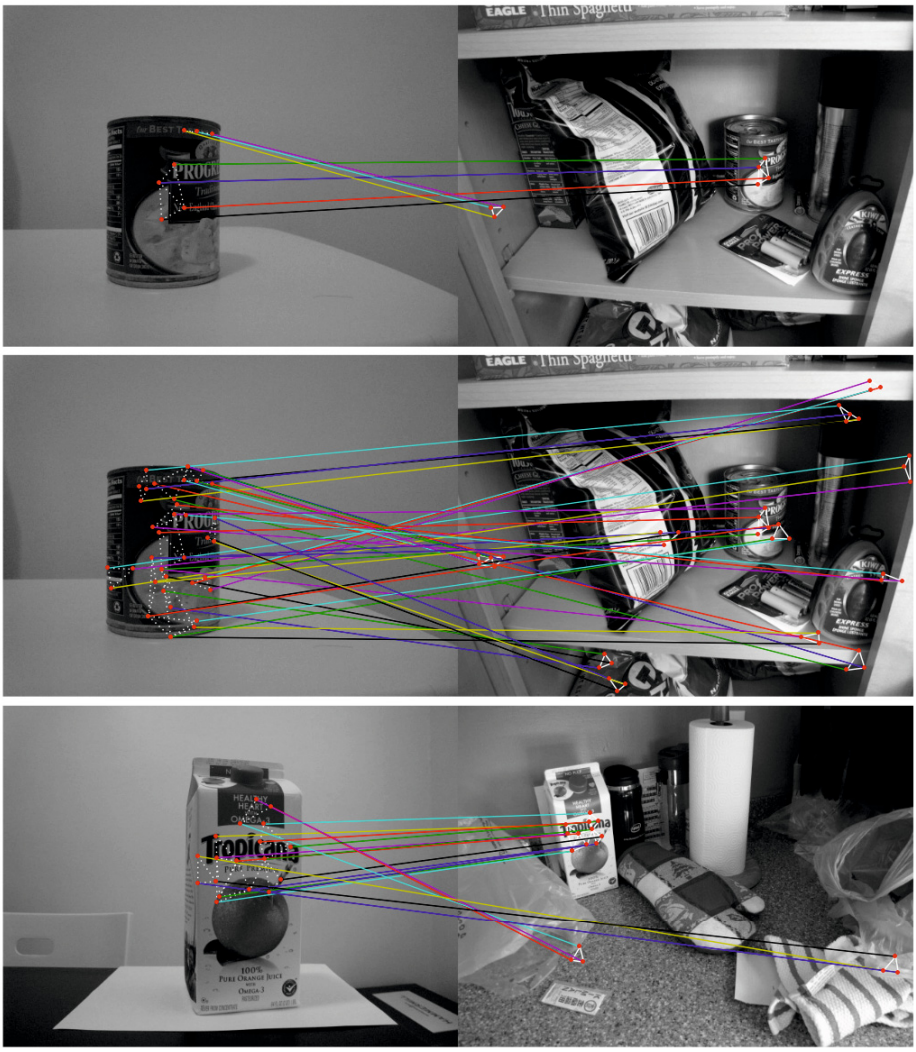


Fig. 1. Keygraph matches established between training (left side) and test (right side) images. Top: "clam chowder can" object: after using the five structural tests, three keygraph matches remain, where two of them are correct. Middle: for the same pair of images of the previous example, we show the keygraph matches that remain after using only four of the five tests; the fifth test, which uses SIFT angle, is not used. It can be seen that the number of (wrong) keygraph matches increases significantly. Bottom: "orange juice carton" object: after using the five tests, we obtain more correct keygraph matches than incorrect ones.

In the present paper, we follow a simpler approach, in which we recognize objects by using an affine transformation between a test and a training image.

We compare the keygraph method proposed in this paper to the ratio test approach [1]. We run each SIFT descriptor v of the test image through the hierarchical k-means tree [5] and obtain the nearest neighbor of v, which is at a distance d_1 from v, and the second nearest neighbor of v that is of a *different* object type than the first nearest neighbor, which is at a distance d_2 from v; a match is established between v and its nearest neighbor if $d_1/d_2 \leq 0.8$ [1]. Then, for every training image for which there are at least four keypoint matches to the test image, we use an exhaustive version of RANSAC, evaluating every possible combination of three keypoint matches to instantiate an affine transformation and then count the number of keypoint matches that agree with this pose. We consider that an instance of an object is found when at least four keypoint matches agree with a transformation (i.e. the three matches used to instantiate it plus one match). We use only four matches for the ratio test method, while for the keygraph method we use six matches (as explained in section 2.3), because the former usually produces fewer keypoints matches than the latter.

We use the same hierarchical k-means tree (with $k = 16$) for both methods, keygraph and ratio test. A query (test) keypoint is compared to a total of 4000 training keypoints stored in the tree leaves; the use of a smaller number of comparisons lowers the accurary of both methods. On average, a training image is described by 1080 SIFT keypoints (using the ground-truth segmentation) and a test image is described by 1070 keypoints (selected to compose the maximal subset $\mathcal{S}$ of keypoints). SIFT descriptors are normalized for zero mean and unit standard deviation; this normalization is useful because we use a threshold $t = 14$ for establishing keypoint matches in the first stage of the keygraph method.

We also compare our keygraph approach to the modified ratio test proposed by Hsiao et al. [6]. Aiming to establish more keypoint matches, the authors proposed to use the regular ratio test in conjunction with a modified ratio test which establishes discriminative matches with *clusters* of keypoints, such that establishing a match with a cluster produces matches to all the training keypoints in that cluster. For that, we create two additional hierarchical k-means trees (with $k = 16$ and $k = 32$). For each test keypoint, for each additional tree we verify whether that test keypoint establishes a discriminative match with a cluster composed of original training keypoints (i.e. a cluster that stores tree leaves); a discriminative match is established if $d_1/d_2 \leq 0.8$, in which d_1 is the distance to the closest cluster and d_2 is the distance to the second nearest cluster. We also use the traditional ratio test (using the original tree with $k = 16$), and employ all the established keypoint matches. Since the modified ratio test generates more keypoint matches then the ratio test, we consider that an instance of an object is found when at least six keypoint matches agree with a transformation, similarly to the keygraph method (for the ratio test, we consider that a pose is found when four keypoint matches agree it).

Table 1 summarizes the results obtained. When a pose is found, we manually verified it by checking if the correct viewpoint of the object was projected in

Table 1. Percentage of test images for which a correct object was found (and for which a wrong object was found), i.e. true positives (and false positives), for the original ratio test [1], the modified ratio test [6] and the keygraph method.

Object type	Ratio test (Lowe [1])	Modified ratio test (Hsiao et al. [6])	Keygraph method (this paper)
Clam chowder can	14% (4%)	22% (4%)	46% (2%)
Soy milk can	2% (12%)	2% (10%)	8% (6%)
Tomato soup can	14%	10% (4%)	36%
Orange juice carton	54% (4%)	58% (2%)	72%
Soy milk carton	44% (8%)	46% (4%)	54% (4%)
Diet coke can	0% (2%)	2% (6%)	2%
Pot roast soup	10% (2%)	6% (4%)	36%
Juice box	26% (10%)	32% (12%)	42% (6%)
Rice pilaf box	64%	62% (2%)	74% (2%)
Rice tuscan box	68% (4%)	58%	62% (2%)

the test image. For the test images with two object instances, we consider that finding just one of them is a correct solution.

Our method performs significantly better than the ratio test and the modified ratio test. In the matching stage (before RANSAC), the keygraph method established an average of 2.8 keygraph matches (7.4 keypoint matches) between a test image and each training image, while the modified ratio test, on average, established only 1.7 keypoint matches between a test and each training image; this number could be increased by using additional k-means trees in the modified ratio test, but we observed that this also increased the false positive rate.

Before the RANSAC stage, the computational time demanded by the ratio test method and the keygraph method is similar, as the time spent to establish keypoint matches through the hierarchical k-means tree is largely dominant in comparison to the next stage of keygraph matching.

4 Conclusion

In this paper we described a method for object matching based on keygraphs, rather than keypoints, i.e., objects are matched using sets of triangles, where each vertex is a keypoint detected and described using SIFT. In the first step, keypoints are matched using a hierarchical k-means tree. Delaunay triangulation generates keygraphs in the test image and the matched keypoints generate keygraphs in the training images. We proposed to use five triangle features in order to evaluate the match between keygraphs, removing a significant number of false matches before running RANSAC to select inliers to compute an affine transformation between training and test images. Our method achieved a significantly higher accuracy than two state-of-the-art methods, the ratio test [1] and the modified ratio test [6]. Furthermore, the number of keygraph matches

is small. On average, 2.8 keygraph matches are established between a test image and each training image. The quality of these matches is high, i.e., a small number of false matches occur. Besides, each keygraph match carries enough information to instantiate a pose hypothesis using an affine transformation. On the other hand, the ratio test method requires at least three keypoint matches.

As future work, we plan to use our method for 3D object recognition and pose estimation as in [6], which uses a structure-from-motion algorithm to create a 3D model of each training object. We believe that our approach is especially suited for this 3D setting. Only two keygraph matches between a test image and (possibly different) training images generate six keypoint matches, which is a good minimal set to generate a 3D pose [7]. This is an important advantage in comparison to a method that solely uses keypoints, which requires the selection of six keypoint matches to instantiate a 3D pose [7] or the selection of four keypoints matches with the use of an algorithm such as EPnP [8]. We expect that the keygraph method will demand a smaller number of 3D pose evaluations.

Another future work involves the use of Domain Adaptation (D.A.) techniques, which are useful when the training data is different from the test data (e.g. [9]). Such a domain change can occur due to variations in the object viewpoint, camera parameters, illumination change, motion etc. We expect that the use of D.A. will improve the first stage of our method (keypoint matching), as this stage is, essentially, a classification task through the hierarchical k-means tree. We also suggest the use of structured learning methods in order to optimize the weight of features used for graph matching, as done in [10].

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
2. Sirmacek, B., Unsalan, C.: Urban-area and building detection using SIFT keypoints and graph theory. IEEE Trans. on Geoscience and Remote Sensing (2009)
3. McAuley, J.J., Caetano, T.S.: Fast matching of large point sets under occlusions. Pattern Recognition (2012)
4. Morimitsu, H., Hashimoto, M., Pimentel, R.B., Cesar Jr., R.M., Hirata Jr., R.: Keygraphs for sign detection in indoor environments by mobile phones. In: Jiang, X., Ferrer, M., Torsello, A. (eds.) GbRPR 2011. LNCS, vol. 6658, pp. 315–324. Springer, Heidelberg (2011)
5. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: VISSAPP (2009)
6. Hsiao, E., Collet, A., Hebert, M.: Making specific features less discriminative to improve point-based 3D object recognition. In: CVPR (2010)
7. Szeliski, R.: Computer vision: algorithms and applications. Springer (2010)
8. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: An accurate O(n) solution to the PnP problem. IJCV (2009)
9. Ni, J., Qiu, Q., Chellappa, R.: Subspace interpolation via dictionary learning for unsupervised domain adaptation. In: CVPR (2013)
10. McAuley, J.J., Campos, T.d., Caetano, T.S.: Unified graph matching in Euclidean spaces. In: CVPR (2010)

Designing LDPC Codes for ECOC Classification Systems

Claudio Marrocco and Francesco Tortorella

Department of Electrical and Information Engineering
Università degli Studi di Cassino e del L.M.
Via G. Di Biasio 43, 03043 Cassino (FR), Italia
{c.marrocco,tortorella}@unicas.it

Abstract. In this paper we analyze a framework for an ECOC classification system founded on the use of LPDC codes, a class of codes well-known in Coding Theory. Such approach provides many advantages over traditional ECOC codings. First, codewords are generated in an algebraic way without requiring any selection of rows and columns of the coding matrix. Second, the decoding phase can be improved by exploiting the algebraic properties of the code. In particular, it is possible to detect and recover possible errors produced by the dichotomizers through an iterative mechanism. Some experiments have been accomplished with the focus on the parity-check matrix used to define the codewords of the LDPC code, so as to determine how the code parameters influence the performance of the proposed approach.

Keywords: ECOC, LDPC codes, Ensemble Methods.

1 Introduction

To face a classification problem with several possible classes the most immediate way is to build a single monolithic classifier capable of producing multiple outputs. However, over the last years, a widely diffused technique consists in decomposing the original problem into a set of two-class problems that can be faced through an ensemble of two-class classifiers. The rationale of this approach relies on the stronger theoretical roots characterizing dichotomizers and makes it possible to employ some very effective classifiers, such as AdaBoost or Support Vector Machines, which are not capable to directly perform multiclass classification.

In this context, the most simple approach is *One-vs-All* that subdivides an n-class problem into n two-class problems each one isolating a class from the others. Another approach, suggested by [11], is *One-vs-One* that defines as many binary problems as the possible pairs of different classes so dividing the n-class problem into a set of $n(n-1)/2$ two-class problems.

A further technique that emerged for its good generalization capabilities is the *Error Correcting Output Coding*(ECOC) [5]. ECOC is commonly used for many applications in the field of Pattern Recognition and Data Mining such as text classification [10] or face recognition and verification [20,13]. A bit string

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 454–463, 2014.

of length L (referred to as *codeword*) is associated with each class so as to have every class represented by a different codeword. The set of codewords is arranged in an $n \times L$ coding matrix $\mathbf{C}$ where the columns define L binary problems, each requiring a specific dichotomizer. The classification task is performed by feeding an unknown sample to the L dichotomizers and collecting their outputs in a vector that is then compared with the n codewords of $\mathbf{C}$ with a proper decoding procedure usually based on the Hamming distance.

In the literature, many studies focused their attention on different aspects of the ECOC technique to improve both the coding and the decoding phase. Several approaches have been proposed to design efficient codes through an analysis of data distribution [7,17,2] or a change of the learning algorithms of the dichotomizers [4,18]. Other techniques tries to minimize the number of employed dichotomizers [3] or to define new decoding rules not based on the Hamming distance [1,6].

In this paper we exploit an ECOC classification system based on well-known codes widely used in the field of Coding Theory, the *Low Density Parity Check* (LDPC) codes [9], to provide a strong theoretical framework both in coding and decoding phase. The aim is to adapt the ECOC framework to the strong algebraic topology of the Coding Theory that is also the original goal of the seminal paper of [5] where the ECOC system is described as a typical communications problem where each sample is transmitted over a communication channel.

We made a preliminary analysis of LDPC codes in the ECOC framework in [15] where a novel decoding rule was proposed to deal with a reject option employed on the dichotomizers. In this paper, instead, we focus on the analysis of the parity-check matrix used to define the codewords of the LDPC code so as to determine how the code parameters influence the performance of the proposed approach. Moreover, besides the design of the coding matrix, we propose an iterative decoding algorithm that exploits the redundancy of the code to increase the performance of the classification system.

The rest of the paper is organized as follows: the next section gives an overview of the Coding Theory. Sect. 3 describes how to design of the coding matrix for LDPC codes and the decoding procedure employed. Some experiments are reported in Sect. 4 while Sect. 5 draws some conclusions and possible future developments.

2 ECOC and Coding Theory

In the usual ECOC approach a multiclass problem is commonly faced by creating a certain number L of two-class subproblems that aggregate in different ways the original M classes into two classes. Each class label $\omega_i, \forall i = 1, \ldots, M$ is represented by a bit string of length L (referred to as *codeword*) only ensuring that every class is represented by a different codeword. Usually, an $M \times L$ coding matrix $\mathbf{C} = \{c_{ij}\}_{i=1,..,M;j=1,..,L}$, with $c_{ij} \in \{0, +1\}$, is created where each row defines a codeword and each column defines the two-class problem on which a dichotomizer has to be trained.

When an unknown sample $\mathbf{x}$ has to be classified by the L dichotomizers the outputs are collected in an output word $\mathbf{o} = \{o_1(\mathbf{x}), o_2(\mathbf{x}), \ldots, o_L(\mathbf{x})\}$ that is compared with the codewords of $\mathbf{C}$ with a proper decoding procedure. Different decoding strategies are commonly adopted [18] but we will only refer to the *Hard Decoding (HD)* where crisp decisions are taken on the outputs of the dichotomizers, i.e., when $o_j(\mathbf{x}) \in \{0, 1\}, \forall j = 1, \ldots, L$. Generally speaking, with the HD rule the decision is taken according to the Hamming distance[1] D_H between the output word and the codewords in $\mathbf{C}$ and the chosen class is typically the "closest" codeword to the output word.

Focusing on the Coding Theory, the main idea of Error Correcting Coding is to introduce redundancy by augmenting the length of the codewords so that it is still possible to recover the original information from the output of a noise-contaminated channel through sets of suitably distinct codewords. In this case, to correctly design the coding matrix $\mathbf{C}$ we have to refer to the algebraic properties of the Galois field $GF(2)$, i.e., a set of two elements, e.g. $\{0, 1\}$, where the mod 2 operations of sum and product are defined. Let us indicate with $GF^L(2)$ the vector space of all L-tuples over $GF(2)$. An (L, K, d) code $\mathcal{C}$ is a K-dimensional vector subspace of $GF^L(2)$ where each vector is a codeword of $\mathcal{C}$ and d is the minimum Hamming distance $d = \min_{i,j} D_H(\mathbf{c}_i, \mathbf{c}_j)$ between any pair of codewords. d is related to the redundancy (i.e., $L-K$) since $d \leq L-K+1$ and thus, it is a measure of the quality of the code since it is possible to decide for the correct codeword if the output word contains no more than $\lfloor (d-1)/2 \rfloor$ erroneous bits.

Let us denote with $\mathbf{u} = [u_0, u_1, ..., u_{K-1}]$ a K-bit source message associated with a codeword $\mathbf{c} = [c_0, c_1, ..., c_{L-1}]$ of $\mathcal{C}$. Since $\mathcal{C}$ is a K-dimensional vector subspace, it is possible to define a basis $\mathbf{g}_0, \ldots, \mathbf{g}_{K-1}$ for $GF^L(2)$. Considering the matrix $\mathbf{G} = \left(\mathbf{g}_0 \ldots \mathbf{g}_{K-1}\right)^T$ the codeword $\mathbf{c}$ corresponding to the source message $\mathbf{u}$ are determined using the linear combination of the basis vectors through $\mathbf{u}$, i.e.,

$$\mathbf{c} = \mathbf{u}\mathbf{G} \tag{1}$$

The matrix $\mathbf{G}$ is a $K \times L$ matrix referred to as *generator matrix* of $\mathcal{C}$. Such a matrix, and thus the codewords, can be evaluated from the *parity-check matrix* $\mathbf{H} = \left(\mathbf{h}_0 \ldots \mathbf{h}_{L-K-1}\right)^T$ since the relation $\mathbf{H}\mathbf{G}^T = \mathbf{0}$ holds. The parity-check matrix collects the $L - K$ vectors $\mathbf{h}_i$ of the basis of the orthogonal complement of $\mathcal{C}$, so that each codeword of $\mathcal{C}$ has to satisfy the condition $\mathbf{H}\mathbf{c}^T = \mathbf{0}$. In this way, the matrix $\mathbf{H}$ is an $(L - K) \times L$ matrix that defines $L - K$ parity-check equations that are used to verify if the received word is actually a codeword of $\mathcal{C}$.

To apply this approach to the ECOC framework, we have to determine the matrices $\mathbf{G}$ and $\mathbf{H}$ of the code $\mathcal{C}$ and thus, the values L and K from the original multiclass problem. Our goal is to keep K as low as possible and thus, we can choose $K = \lceil \log_2 M \rceil$. L, instead, is a variable parameter chosen by considering

[1] The Hamming distance between two words is given by the number of position where the bit patterns of the two words differ.

that higher values correspond to a higher error correction capability. When $\mathbf{G}$ is evaluated, codewords (and thus, the coding matrix $\mathbf{C}$) are calculated through eq. 1 and the learning phase can proceed by training a dichotomizer on each column of $\mathbf{C}$. In the decoding phase, when classifying an unknown sample $\mathbf{x}$, the output vector $\mathbf{o}$ is received. Generally speaking, $\mathbf{o}$ can be seen as the sum between a codeword $\mathbf{c}$ and an error pattern $\mathbf{e}$, i.e., $\mathbf{o} = \mathbf{c} + \mathbf{e}$ and thus, we have:

$$\mathbf{s} = \mathbf{H}\mathbf{o}^T = \mathbf{H}\mathbf{c}^T + \mathbf{H}\mathbf{e}^T = \mathbf{H}\mathbf{e}^T \neq \mathbf{0} \tag{2}$$

where $\mathbf{s}$ is a $L-K$-vector referred to as the *syndrome* of $\mathbf{o}$. Eq. (2) represents a parity-check condition and can be used to determine when a valid codeword is found (i.e., when the syndrome is equal to zero). This does not necessarily means that the ECOC system has correctly classified the sample $\mathbf{x}$. In fact, the error pattern $\mathbf{e}$ can be such that the vector $\mathbf{c} + \mathbf{e}$ corresponds to another codeword, different from the true one. This happens when at least d dichotomizers are wrong. If the number of errors is less than d, we obtain $\mathbf{s} \neq \mathbf{0}$ and the HD rule can be applied as in the usual ECOC framework.

3 Designing LDPC Codes

Among the several families of code provided by the Coding Theory, we considered the *Low Density Parity Check* (LDPC) codes presented by Gallager [9] in 1963. LDPC codes are a class of linear block codes characterized by a sparse pseudo-random parity-check matrix able to reach very high performance by strongly increasing the redundancy. The term "low density" indicates that LDPC codes are specified by a matrix $\mathbf{H}$ containing mostly 0's and relatively few 1's so that each parity-check equation defined by $\mathbf{H}$ involves a small number of bits of the output vector and each bit enters in a small number of parity-check equations.

The main difference between LDPC codes and classical block codes is the way they are decoded. Classical codes employ the HD rule and are algebraically designed to make this task less complex. LDPC codes, instead, are iteratively decoded using the sparseness of the parity-check matrix and thus, are designed with the properties of $\mathbf{H}$ as focal point. In this context, two different families of LDPC codes are usually employed: *regular* and *irregular* codes. To this end, let us define w_c and w_r as the number of 1's, respectively, in each column and each row of the parity-check matrix. A (w_c, w_r)-regular LDPC code is a binary linear code where w_c is constant for every column and $w_r = w_c \dfrac{L-K}{L}$ is also constant for every row. On the other hand, an LDPC code is irregular if the number of ones per row or per column are not fixed.

To easily represent an LDPC code, a bipartite graph (referred to as *Tanner graph* [19]) is commonly used to show how each component of the output vector is involved in the parity check constraints. The nodes of the graph are separated into L *variable nodes*, corresponding to every component of the output vector, and $L-K$ *check nodes*, corresponding to the parity check constraints, i.e., to the rows of $\mathbf{H}$. Edges only connect nodes of different types so that every check

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

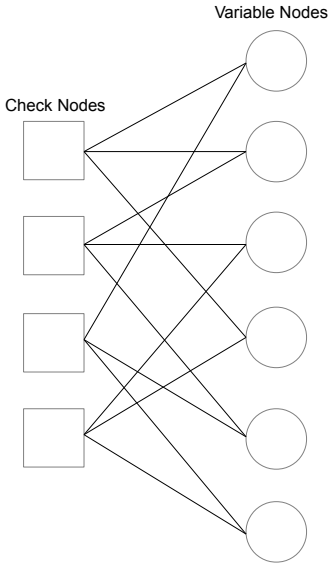


Fig. 1. The parity-check matrix **H** and its corresponding Tanner graph for a regular LDPC codes with $w_c = 2$, $w_r = 3$ and $L = 6$

node i is connected to a variable node j if and only if $h_{ij} = 1$. Denoting with *degree* of the node the number of connections deriving from a node, we can refer to a (w_c, w_r)-regular LDPC code when its Tanner graph has every variable node with degree w_c and every check node with degree w_r. An example of Tanner graph for a regular LDPC code is shown in Fig. 1.

Several different algorithms exists to construct suitable LDPC codes. The original LDPC codes presented by [9] are regular and consists of forming a sparse parity-check matrix by randomly determining the positions of 1's. Beyond the constant number of 1's in the **H** matrix, another important condition to be satisfied is that the overlapping of 1's per column and per row should be at most equal to one. The goal here is to avoid the presence of cycles of length 4 (referred to as *4-cycles*) in the corresponding bipartite graph [16]. A *cycle* in a Tanner graph is a sequence of connected vertices which start and end at the same vertex in the graph, and which contain other vertices only once. The length of a cycle is the number of edges it contains. Designing efficient LDPC codes without 4-cycles is however quite unavoidable. To this end, [14] proposed another common construction for LDPC codes useful to also define irregular codes and easily adapted to avoid 4-cycles by checking each pair of columns in **H** to see if they overlap in two places.

The main advantage of an LDPC code are very effective in the decoding procedure. In particular, LDPC codes can be usefully exploited to possibly correct

Table 1. Data sets and code parameters used in the experiments

Data Sets	Classes	Features	Samples	K	Dichotom.
SatImage	6	36	6435	3	7
Dermatology	6	33	366	3	7
Glass	7	9	214	3	7
Segmentation	7	18	2310	3	7
Ecoli	8	7	341	4	7
Optdigits	10	62	5620	4	14
Pendigits	10	16	10992	4	14
Yeast	10	8	1484	4	14
Vowel	11	10	435	4	15

the errors obtained after the Hard Decoding procedure. To this end, a very effective iterative message-passing decoding algorithm has been proposed in [9]. Such method is based on the principle that a bit of an output word involved in a large number of incorrect check equations is likely to be incorrect itself. The parity-check equations unlikely contain the same set of codeword bits because the sparseness of $\mathbf{H}$ helps to disperse variable bits into check nodes. To explain how the iterative decoding works, let us consider an initial hard decision for each received bit, i.e., $o_i \in \{0, 1\}$ and let us focus on the Tanner graph representation. The decoding algorithm is based on message passing between the nodes of the Tanner graph: a variable node sends its bit value to each of the check nodes to which it is connected and each check node answers determining if its parity-check equation is satisfied or not. If the majority of the check values received by a variable node are different from zero the variable node flips its current value.

A last remark to be done on LDPC codes is that depending on the structure of $\mathbf{H}$, the matrix $\mathbf{C}$ can contain equal columns as well as all-zeros or all-ones columns. Unlike the usual ECOC, when using the LDPC codes such columns are not eliminated otherwise the algebraic properties of the code would not be guaranteed. Actually, the all-zeros/all-ones columns are not considered during the training of the dichotomizers and the bits corresponding to them are then recovered within the output word before the decoding phase starts. As for the equal columns, they are assigned the same dichotomizer; in this way the number of dichotomizers is reasonable even though the number of total columns is high (100 or more). This is an important issue since the sparseness of the parity-check matrix guarantees a minimum Hamming distance linearly increasing with the code length while the decoding complexity linearly increases only with the number of employed classifiers.

4 Experiments

The goal of the experiments is to verify how the structure of the parity-check matrix affects the performance of the proposed approach. For this purpose, we have considered some data sets publicly available at the UCI Machine Learning

Repository [8]. All the employed data sets have numerical input features and a variable number of classes. For each data set, to avoid any bias in the comparison, 10 runs of a multiple hold-out procedure have been performed. In each run, we considered three subsets: a training, a tuning and a test set containing respectively the 50%, the 30% and the 20% of the samples of each class. The training set has been used to train the base classifiers while the test set to evaluate the performance of the multiclass classification system. The tuning set, instead, is used to optimize the parameters of the base dichotomizers that are SVM with RBF kernel [12]. The optimization of the parameters (γ of the kernel and C) has been done by following a grid approach.

Some parameters of the LDPC coding architecture have been varied and the matrix $\mathbf{H}$ obtained has been used with all the data sets to evaluate the coding matrix and to apply the iterative decoding rule. Accordingly, both regular and irregular codes were considered. For each data set a coding matrix has been determined according to Sect. 2 with K depending on the number of classes and L equal to 50 and 100 (the details on data sets, code parameters and dichotomizers are resumed in Table 1). Moreover, the coding matrix parameters w_c and w_r were varied: for regular codes, between the 10% and the 50% of, respectively, L and $L-K$ while for irregular codes w_c between the 10% and the 50% of L while w_r has been randomly chosen with the procedure in [14] trying to avoid the 4-cycles. For the sake of comparison we have also considered the results obtained with a decomposition One-vs-All (OVA) and One-vs-One (OVO) both with a Hard Decoding rule.

In each experiment (i.e., for each of the coding matrices and for each dataset), we have evaluated the mean classification error, calculated by averaging the error rates obtained on the test set in the 10 runs of the multiple hold-out procedure. However, since the results obtained on the different datasets are not commensurable, we have used a rank-based comparison. For each dataset separately, we evaluated the *reverse rank* for each coding matrix: the best performing matrix got the maximum rank (i.e., 22, since we considered 20 different LDPC coding matrices plus OVA and OVO), while the worst got 1. In case of ties, average ranks were assigned. In this way, if r_k^h was the rank obtained by the h-th coding matrix on the k-th dataset, the average performance of the h-th coding matrix on all the datasets was $r^h = \frac{1}{T}\sum_{k=1}^{T} r_k^h$, where T is the number of datasets considered.

Table 2 reports the results obtained on all datasets. We can observe that the best results are attained for relatively low values of the parameters w_r and w_c (20%-30%), while the best choice is a regular LDPC code with $L = 100$. As for the "classical" decompositions, OVO works worse than the most part of LDPC code matrices considered, while OVA is definitely the worst solution.

In order to have some more insights we have also split the datasets in two groups according the number of classes: the first group contains the datasets with 6-8 classes, while the second one collects the more numerous datasets (10-11 classes). The results on the two groups of datasets are separately shown in

Table 2. Results on all the datasets in terms of mean rank among the various datasets. The higher the value, the better performing the corresponding coding matrix.

L	10	20	30	40	50	OVA	OVO
50 Regular	12.00	12.33	15.33	8.44	7.78		
100 Regular	14.89	18.00	16.56	12.11	12.33	2.33	7.67
50 Irregular	14.67	14.33	11.33	11.33	12.89		
100 Irregular	15.22	11.78	13.56	8.78	13.00		

Table 3 and Table 4, adopting the same approach described before. What we can observe is that, while the best results are still in correspondence of low values of w_r and w_c, there is now some slight difference in performance when varying these parameters. In fact, the first group has remarkably better results for $w_c = 20\%$, while the second group works better when $w_c = 30\%$. Obviously this is not sufficient to establish a sort of relation between the number of classes and the value of w_r: more experiments are needed on datasets with higher and higher number of classes. Any way, a regular LDPC code with $L = 100$ is still the best choice in both cases.

Table 3. Results on the datasets with 6-8 classes in terms of mean rank among the various datasets. The higher the value, the better performing the corresponding coding matrix.

L	10	20	30	40	50	OVA	OVO
50 Regular	12.00	13.20	17.60	10.80	10.40		
100 Regular	16.80	19.60	13.80	12.80	10.80	3.00	11.80
50 Irregular	16.00	13.20	11.80	12.60	14.00		
100 Irregular	14.40	9.40	16.20	12.20	12.40		

Table 4. Results on the datasets with 10-11 classes in terms of mean rank among the various datasets. The higher the value, the better performing the corresponding coding matrix.

L	10	20	30	40	50	OVA	OVO
50 Regular	12.00	11.25	12.50	5.50	4.50		
100 Regular	12.50	16.00	20.00	11.25	14.25	1.50	2.50
50 Irregular	13.00	15.75	10.75	9.75	11.50		
100 Irregular	16.25	14.75	10.25	4.50	13.75		

5 Conclusions and Future Works

In this paper an approach based on the Coding Theory and in particular on the LPDC codes has been analyzed. Such approach provides several advantages over traditional ECOC code solutions since codewords are generated in an algebraic way without requiring any selection of rows and columns of the coding matrix.

Moreover, the decoding phase can be improved by employing an iterative mechanism that, exploiting the algebraic properties of the code, is able to detect and recover possible errors produced by the dichotomizers. Some preliminary experiments have been focused on the analysis of the parity-check matrix used to define the codewords of the LDPC code, so as to determine how the code parameters influence the performance of the proposed approach. The results seem encouraging even though more trials are needed to verify if some general relation can be drawn.

Some possible future developments will focus on a deeper analysis of the employed coding matrix and on the use of other decoding rules. In particular, the performance of the ECOC system in our experiments increases when the codeword length L increases and thus, an analysis on longer codewords should be conducted to better delineate this point. Another issue worth of consideration is the use of decoding rules working with soft-output dichotomizers, i.e., dichotomizers providing real valued outputs instead of crisp decisions. In such case, the confidence of the dichotomizer in making a classification can be used to provide an additional information to the decoding phase and suitable decoding techniques (e.g., a *loss-based* technique [18]) can be employed to improve the recognition performance.

References

1. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: A unifying approach for margin classifiers. Journal of Machine Learning Research 1, 113–141 (2000)
2. Alpaydin, E., Mayoraz, E.: Learning error-correcting output codes from data. In: Ninth International Conference on Artificial Neural Networks, ICANN 1999 (Conf. Publ. No. 470), vol. 2, pp. 743–748 (1999)
3. Bautista, M.Á., Escalera, S., Baró, X., Radeva, P., Vitrià, J., Pujol, O.: Minimal design of error-correcting output codes. Pattern Recognition Letters 33(6), 693–702 (2012)
4. Crammer, K., Singer, Y.: On the learnability and design of output codes for multiclass problems. In: Cesa-Bianchi, N., Goldman, S.A. (eds.) COLT, pp. 35–46. Morgan Kaufmann (2000)
5. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research 2, 263–286 (1995)
6. Escalera, S., Pujol, O., Radeva, P.: On the decoding process in ternary error-correcting output codes. IEEE Trans. Pattern Anal. Mach. Intell. 32(1), 120–134 (2010)
7. Escalera, S., Tax, D.M.J., Pujol, O., Radeva, P., Duin, R.P.W.: Subclass problem-dependent design for error-correcting output codes. IEEE Trans. Pattern Anal. Mach. Intell. 30(6), 1041–1054 (2008)
8. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
9. Gallager, R.G.: Low density parity-check codes. MIT Press (1963)
10. Ghani, R.: Combining labeled and unlabeled data for text classification with a large number of categories. In: Cercone, N., Lin, T.Y., Wu, X. (eds.) ICDM, pp. 597–598. IEEE Computer Society (2001)

11. Hastie, T., Tibshirani, R.: Classification by pairwise coupling. In: Jordan, M.I., Kearns, M.J., Solla, S.A. (eds.) Advances in Neural Information Processing Systems, vol. 10. The MIT Press (1998)
12. Joachims, T.: Making large-scale SVM learning practical. In: Schölkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods - Support Vector Learning, ch. 11. MIT Press, Cambridge (1999)
13. Kittler, J., Ghaderi, R., Windeatt, T., Matas, J.: Face verification via error correcting output codes. Image Vision Comput. 21, 1163–1169 (2003)
14. MacKay, D.J.C.: Good error-correcting codes based on very sparse matrices. IEEE Transactions on Information Theory 45, 399–431 (1999)
15. Marrocco, C., Simeone, P., Tortorella, F.: Coding theory tools for improving recognition performance in ECOC systems. In: Zhou, Z.-H., Roli, F., Kittler, J. (eds.) MCS 2013. LNCS, vol. 7872, pp. 201–211. Springer, Heidelberg (2013)
16. Moreira, J.C., Farrell, P.G.: Essentials of error-control coding. John Wiley & Sons (2006)
17. Pujol, O., Radeva, P., Vitrià, J.: Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes. IEEE Trans. Pattern Anal. Mach. Intell. 28, 1007–1012 (2006)
18. Simeone, P., Marrocco, C., Tortorella, F.: Design of reject rules for ECOC classification systems. Pattern Recognition 45(2), 863–875 (2012)
19. Tanner, R.M.: A Recursive Approach to Low Complexity Codes. IEEE Transactions on Information Theory 27, 533–547 (1981)
20. Windeatt, T., Ardeshir, G.: Boosted ECOC ensembles for face recognition. In: Proceedings of the International Conference on Visual Information Engineering, pp. 165–168 (2003)

Unifying Probabilistic Linear Discriminant Analysis Variants in Biometric Authentication

Aleksandr Sizov[1], Kong Aik Lee[2], and Tomi Kinnunen[1]

[1] School of Computing, University of Eastern Finland, Finland
[2] Institute for Infocomm Research (I^{2}R), Singapore

Abstract. Probabilistic linear discriminant analysis (PLDA) is commonly used in biometric authentication. We review three PLDA variants — *standard*, *simplified* and *two-covariance* — and show how they are related. These clarifications are important because the variants were introduced in literature without argumenting their benefits. We analyse their predictive power, covariance structure and provide scalable algorithms for straightforward implementation of all the three variants. Experiments involve state-of-the-art speaker verification with *i*-vector features.

Keywords: PLDA, speaker and face recognition, *i*-vectors.

1 Introduction

Biometric person authentication — recognizing persons from their physiological or behavioral traits — plays an increasingly important role in information security [1]. Face [2] and speaker [3] recognition are particularly attractive due to their unintrusiveness and low costs. Unfortunately, both involve prominent sample-to-sample variations that lead to decreased recognition accuracy; face images can be shot under differing lighting conditions or cameras and speech signals acquired using different microphones. Compensating for these nuisance variations is crucial for achieving robust recognition under varying conditions.

From various techniques studied, generative probabilistic models are among the top-performing ones for both face and speaker verification. A powerful, yet simple technique is *factor analysis* [4]. Given a feature vector that represents a single speech utterance or a face image, factor analysis captures the main correlations between its coordinates. A successful recent extension is the *probabilistic linear discriminant analysis* (PLDA) model [2,5], where we split the total data variability into within-individual and between-individual variabilities, both residing on small-dimensional subspaces. Originally introduced in [2] for face recognition, PLDA has become a *de facto* standard in speaker recognition. We restrict our focus and experiments to speaker recognition but the general theory holds for arbitrary features.

Besides the original PLDA formulation [2], we are aware of two alternative variants that assume full covariance: *simplified* PLDA [6] and *two-covariance model* [7]. It is worth noting that the three models are related in terms of their

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 464–475, 2014.

predictive power (degrees of freedom), covariance structure and computations. The main purpose of the current study is to provide a self-contained summary that elaborates the differences. The main benefit in doing so is that, instead of three different PLDA variants and learning algorithms, we show how to apply the *same* optimizer by merely modifying the latent subspace dimensions appropriately. As a further practical contribution, we provide an optimized open-source implementation[1].

2 Unified Formulation of PLDA and Its Variants

We assume that the training set consists of K disjoint persons. For the i-th person we have n_i enrolment samples, each being represented by a single feature vector [2] $\boldsymbol{\phi}_{i\cdot}$. The PLDA models described below assume these vectors to be drawn from different generative processes.

2.1 Three Types of a PLDA Model

The first one is a **standard PLDA** as defined in the original study [2]:

$$\boldsymbol{\phi}_{ij} = \boldsymbol{\mu} + \mathbf{V}\mathbf{y}_i + \mathbf{U}\mathbf{x}_{ij} + \boldsymbol{\varepsilon}_{ij}\,, \tag{1}$$

$$\mathbf{y}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\,, \tag{2}$$

$$\mathbf{x}_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\,, \tag{3}$$

$$\boldsymbol{\varepsilon}_{ij} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}^{-1})\,, \tag{4}$$

where $\boldsymbol{\phi} \in \mathbb{R}^{D\times 1}$, $\boldsymbol{\Lambda}$ is a diagonal precision matrix, $\boldsymbol{\mu}$ is a global mean, columns of the matrices $\mathbf{V} \in \mathbb{R}^{D\times P}$ and $\mathbf{U} \in \mathbb{R}^{D\times M}$ span the between- and within-individual subspaces. The second one is a **simplified PLDA** introduced in [6] and used in [9], [10], [11]:

$$\boldsymbol{\phi}_{ij} = \boldsymbol{\mu} + \mathbf{S}\mathbf{y}_i + \boldsymbol{\varepsilon}_{ij}\,, \tag{5}$$

$$\mathbf{y}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\,, \tag{6}$$

$$\boldsymbol{\varepsilon}_{ij} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}_f^{-1})\,, \tag{7}$$

where $\boldsymbol{\Lambda}_f$ is a full precision matrix instead of the diagonal matrix in the standard PLDA case and $\mathbf{S} \in \mathbb{R}^{D\times L}$. The third one is a **two-covariance model** introduced in [7] and used extensively in [12]:

$$\mathbf{y}_i \sim \mathcal{N}(\mathbf{y}_i|\boldsymbol{\mu}, \mathbf{B}^{-1})\,, \tag{8}$$

$$\boldsymbol{\phi}_{ij}|\mathbf{y}_i \sim \mathcal{N}(\boldsymbol{\phi}_{ij}|\mathbf{y}_i, \mathbf{W}^{-1})\,, \tag{9}$$

where both $\mathbf{B}$ and $\mathbf{W}$ are *full* precision matrices. Thus, unlike the two previous models, we no longer have any subspaces with reduced dimensionality.

[1] `https://sites.google.com/site/fastplda/`

[2] Traditionally, speech utterances have been represented as a sequence of acoustic feature vectors. In this paper we use the i-vector [8] representation that produces a fixed length vector from the variable length sequence. More on this in Section 4.

2.2 Exploring the Structure of the Models

All the latent variables in the standard PLDA formulation (1) have a Gaussian distribution. Thus, the distribution of the observed variables is also a Gaussian:

$$\boldsymbol{\phi}_{ij}|\mathbf{y}_i, \mathbf{x}_{ij} \sim \mathcal{N}(\boldsymbol{\phi}_{ij}|\boldsymbol{\mu} + \mathbf{V}\mathbf{y}_i + \mathbf{U}\mathbf{x}_{ij}, \boldsymbol{\Lambda}^{-1}), \tag{10}$$

and an integration of the channel latent variables $\{\mathbf{x}_{ij}\}$ leads to a closed-form result:

$$\boldsymbol{\phi}_{ij}|\mathbf{y}_i \sim \mathcal{N}(\boldsymbol{\phi}_{ij}|\boldsymbol{\mu} + \mathbf{V}\mathbf{y}_i, \mathbf{U}\mathbf{U}^{\mathsf{T}} + \boldsymbol{\Lambda}^{-1}). \tag{11}$$

We can now formulate (11) in a similar style as the two-covariance model:

$$\tilde{\mathbf{y}}_i \sim \mathcal{N}(\tilde{\mathbf{y}}_i|\boldsymbol{\mu}, \mathbf{V}\mathbf{V}^{\mathsf{T}}), \tag{12}$$

$$\boldsymbol{\phi}_{ij}|\tilde{\mathbf{y}}_i \sim \mathcal{N}(\boldsymbol{\phi}_{ij}|\tilde{\mathbf{y}}_i, \mathbf{U}\mathbf{U}^{\mathsf{T}} + \boldsymbol{\Lambda}^{-1}). \tag{13}$$

Comparing (12) with (8) and (13) with (9) reveals that the structure of a standard PLDA and a two-covariance model is the same and their only difference is in the covariance matrices. Let us call within- and between-individual covariance matrices of the n-th model as $\mathbf{W}_n^{-1}$ and $\mathbf{B}_n^{-1}$ (see Table 1), so that,

$$\mathbf{W}_3^{-1} = \mathbf{W}^{-1}, \tag{14}$$

$$\mathbf{B}_3^{-1} = \mathbf{B}^{-1}. \tag{15}$$

From (12) and (13) we conclude that

$$\mathbf{W}_1^{-1} = \mathbf{U}\mathbf{U}^{\mathsf{T}} + \boldsymbol{\Lambda}^{-1}, \tag{16}$$

$$\mathbf{B}_1^{-1} = \mathbf{V}\mathbf{V}^{\mathsf{T}}. \tag{17}$$

Applying the same analysis to the simplified PLDA leads to the following equations:

$$\mathbf{W}_2^{-1} = \boldsymbol{\Lambda}_f^{-1}, \tag{18}$$

$$\mathbf{B}_2^{-1} = \mathbf{S}\mathbf{S}^{\mathsf{T}}. \tag{19}$$

2.3 Calculating the Degrees of Freedom

We have seen that all the three models have the same structure, but their predictive powers differ because they have different number of independent parameters. It is a known fact that for a factor analysis model latent subspace has *rotational invariance* (see [4, Page 576]). If $\mathbf{R}$ is an arbitrary orthogonal matrix (that is, $\mathbf{R}\mathbf{R}^{\mathsf{T}} = \mathbf{R}^{\mathsf{T}}\mathbf{R} = \mathbf{I}$) then

$$\mathbf{B}_1^{-1} = \mathbf{V}\mathbf{V}^{\mathsf{T}} = \mathbf{V}(\mathbf{R}\mathbf{R}^{\mathsf{T}})\mathbf{V}^{\mathsf{T}} = (\mathbf{V}\mathbf{R})(\mathbf{V}\mathbf{R})^{\mathsf{T}}, \tag{20}$$

so that $\mathbf{V}$ and $\tilde{\mathbf{V}} = \mathbf{V}\mathbf{R}$ lead to the same covariance matrix and the same model. This ambiguity means that a particular solution is not unique.

In the two-covariance model both $\mathbf{W}_3^{-1}$ and $\mathbf{B}_3^{-1}$ are full and symmetric matrices so each of them has $D(D+1)/2$ degrees of freedom. In the case of standard PLDA, $\mathbf{W}_1^{-1} = \mathbf{U}\mathbf{U}^\mathsf{T} + \mathbf{\Lambda}^{-1}$ has $DM + D - M(M-1)/2$ degrees of freedom, where the second term is due to diagonal noise matrix and the last term is due to rotational invariance property. The same argument can be applied to the remaining matrices. Table 1 summarizes the degrees of freedom for each of the three models.

Table 1. Degrees of freedom for each model. Here, D is the dimensionality of the feature vectors, P and L are the number of basis vectors for between-individual subspaces of the corresponding models, M is a number of basis vectors for within-individual subspace, $\mathbf{B}_n$ and $\mathbf{W}_n$ are a between-individual and within-individual precision matrices for n-th model.

PLDA type	$\boldsymbol{\mu}$	$\mathbf{B}_n$	$\mathbf{W}_n$
$n=1$ (standard)	$D+$	$DP - \frac{P(P-1)}{2} +$	$DM + D - \frac{M(M-1)}{2}$
$n=2$ (simplified)	$D+$	$DL - \frac{L(L-1)}{2} +$	$\frac{D(D+1)}{2}$
$n=3$ (two-covariance)	$D+$	$\frac{D(D+1)}{2} +$	$\frac{D(D+1)}{2}$

Regarding the degrees of freedom, we conclude the following from Table 1:

1. When $L = D$ (factor loadings matrix is of full rank) the simplified PLDA is equivalent to the two-covariance model.
2. When $P = D$ and $M = D - 1$ the standard PLDA model is equivalent to the two-covariance model.
3. When $P = L$ and $M = D - 1$ the standard PLDA model is equivalent to the simplified PLDA.

To sum up, the standard PLDA is the most general model, and a two-covariance is the least general model.

2.4 Over-Complete Case

It is important to note that the above equations hold *only* when the dimensionality of the latent variables is less or equal to the dimensionality of the data. Otherwise, we have an over-complete basis for a latent variable subspace and we need an additional step before analysing the model. To this end, suppose that the matrix $\mathbf{V} \in \mathbb{R}^{D\times P}$ has more columns than D, then this matrix affects generative process (12) only in the form $\mathbf{V}\mathbf{V}^\mathsf{T}$. When $P > D$, this $D \times D$ matrix

has a rank D. As a symmetric positive-definite matrix, we may apply Cholesky decomposition to get,

$$\mathbf{V}\mathbf{V}^{\mathsf{T}} = \mathbf{L}\mathbf{L}^{\mathsf{T}}, \tag{21}$$

where $\mathbf{L} \in \mathbb{R}^{D \times D}$ is an upper triangular matrix. Without loss of generality, we can choose $\mathbf{V} = \mathbf{L}$ and transform an over-complete case to a complete one. The same argument holds for matrices $\mathbf{U}$ and $\mathbf{S}$.

2.5 Scoring

At verification stage we are given a pair of individual models: one created from the test features and the other from enrolment features of the claimed person and we need to decide whether these models belong to the same person. To do this in a PLDA approach we need to calculate a log-likelihood ratio between two hypothesis: both models share the same latent identity variable or they share a different identity variables. The scoring equations are the same for all models but due to lack of space we do not present them here. For an optimized scoring procedure please consult [13].

3 EM-Algorithms

The original EM-algorithm proposed in [2] has a serious drawback: at the E-step we need to invert a matrix whose size grows linearly with the number of samples per individual. For large datasets this algorithm becomes highly impractical. A number of solutions for this problem have been introduced. In [14], the authors utilize a special matrix structure of PLDA model and manually derive equations for the required matrix inversions. In [15], the authors proposed a special change of variables that lead to a diagonalized versions of the required matrices. The most detailed derivations are given in [16]. Our version was based on [14] and accelerated in a similar style as in [16]. The algorithm 1 summarizes it and the details are presented in the appendix A.

Incomplete algorithm (only E-step) for the two-covariance model is given in [7]. Here we present complete solution in the form of short summary (see algorithm 2). The details are available in the appendix B.

Technical notes:

- The rank of matrix $\mathbf{V}$ is equal to the rank of $\mathbf{T}_{\mathbf{y}}$, which is just the number of individuals in the training set. So, this is an upper bound for the number of columns of matrix $\mathbf{V}$ that we should choose.
- If in the algorithm 1 we set matrix $\mathbf{U}$ to zero and do not constrain noise precision matrix $\mathbf{\Lambda}$ to be diagonal we get EM-algorithm for the simplified PLDA model [17].
- For the two-covariance model the number of individuals in the training set should be bigger than the dimensionality of feature vectors (i-vectors, in our case).

Algorithm 1. Scalable PLDA learning algorithm

Input: $\mathbf{\Phi} = \{\boldsymbol{\phi}_{ij}\}_{i=1,j=1}^{K,n_i}$, where K is a total number of persons, and n_i is the number of samples for i-th person.
Output: Estimated matrices $\mathbf{V}$, $\mathbf{U}$ and $\mathbf{\Lambda}$.
Sort persons according to the number of samples $\{n_i\}$;
Find total number of samples N and center the data (eq. A.1 and A.2) ;
Compute data statistics $\{\mathbf{f}_i\}$ and $\mathbf{S}$ (eq. A.3 and A.4) ;
Initialize $\mathbf{V}$ and $\mathbf{U}$ with small random values, $\mathbf{\Lambda} \leftarrow N\mathbf{S}^{-1}$;
repeat
  E-step:
    Set $\mathbf{R} \leftarrow 0$;
    Compute auxiliary matrices $\mathbf{Q}$, $\mathbf{J}$ (eq. A.5 and A.6) ;
    for $i = 1$ *to* K **do**
      if $n_i \neq n_{i-1}$ **then** compute $\mathbf{M}_i$ (eq. A.7);
      else $\mathbf{M}_i \leftarrow \mathbf{M}_{i-1}$;
      Find $\mathbb{E}[\mathbf{y}_i]$ (eq. A.8) ;
      Update $\mathbf{R}_{\mathbf{yy}}$ (eq. A.13);
    Calculate $\mathbf{T}$, $\mathbf{R}_{\mathbf{yx}}$ and $\mathbf{R}_{\mathbf{xx}}$ (eq. A.12, A.14 and A.15) ;
  M-step:
    Find $\mathbf{V}$, $\mathbf{U}$, $\mathbf{\Lambda}$ (eq. A.16 and A.17) ;
  MD-step:
    Compute auxiliary matrices $\boldsymbol{\mathcal{Y}}$, $\mathbf{G}$, $\boldsymbol{\mathcal{X}}$ (eq. A.18, A.19 and A.20) ;
    Update $\mathbf{U}$, $\mathbf{V}$ (eq. A.21 and A.22) ;
until Convergence;

Algorithm 2. Two-covariance model learning algorithm

Input: $\mathbf{\Phi} = \{\boldsymbol{\phi}_{ij}\}_{i=1,j=1}^{K,n_i}$, where K is a total number of persons, and n_i is a number of samples for i-th person.
Output: Estimated matrices $\boldsymbol{\mu}$, $\mathbf{B}$ and $\mathbf{W}$.
Sort persons according to the number of samples $\{n_i\}$;
Compute data statistics N, $\{\mathbf{f}_i\}$ and $\mathbf{S}$ (eq. B.1, B.2 and B.3) ;
Initialize $\boldsymbol{\mu}$, $\mathbf{B}$, $\mathbf{W}$;
repeat
  E-step:
    Set $\mathbf{T} \leftarrow 0$, $\mathbf{R} \leftarrow 0$, $\boldsymbol{\mathcal{Y}} \leftarrow 0$;
    for $i = 1$ *to* K **do**
      if $n_i \neq n_{i-1}$ **then** compute $\mathbf{L}_i$ (eq. B.4);
      else $\mathbf{L}_i \leftarrow \mathbf{L}_{i-1}$;
      Find $\mathbb{E}[\mathbf{y}_i]$ and $\mathbb{E}[\mathbf{y}_i\mathbf{y}_i^{\mathsf{T}}]$ (eq. B.5, B.6) ;
      Update $\mathbf{T}$, $\mathbf{R}$ and $\boldsymbol{\mathcal{Y}}$ (eq. B.8, B.9 and B.10);
  M-step:
    Find $\boldsymbol{\mu}$, $\mathbf{B}$ and $\mathbf{W}$ (eq. B.11, B.12 and B.13) ;
until Convergence;

4 Experiments

4.1 System Setup

In modern speaker and language recognition, a speech utterance can be represented using its *i-vector* [8]. Briefly, variable-duration feature sequences are first mapped to utterance-specific Gaussian mixture models (GMMs). The i-vector is a low-dimensional latent representation of the corresponding GMM mean supervector [18], typical dimensionality varying from 400 to 600. This is sufficiently low to robustly estimate a full within-individual variation covariance matrix [6].

Our i-vector system uses standard Mel-frequency cepstral coefficient (MFCC) features involving RASTA filter, delta and double delta coefficients, energy-based speech activity detector [19] and utterance level cepstral mean and variance normalization (CMVN), in this order. Gender-dependent universal background models (UBMs) were trained with data from NIST 2004, 2005 and 2006 data and gender-dependent i-vector extractors from NIST 2004, 2005, 2006, Fisher and Switchboard. For more details, see [20]. For the experiments we used only female subset which has 578 train speakers, 21216 train segments, 459 test speakers, 10524 target trails and 6061824 non-target trials.

The UBM has 1024 Gaussians and i-vector dimensionality is set to 600. The i-vectors are whitened and length-normalized [9]. Speaker verification accuracy is measured through the equal error rate (EER) corresponding to the operating point with equal false acceptance and false rejection rates.

4.2 Comparison of Different PLDA Configurations

We made a thorough comparison of different PLDA configurations. Since PLDA training uses random initialization, we made 10 independent runs for each tested configuration and averaged the EERs. Although usually PLDA models achieve the best performance when they are slightly under-trained, the number of iterations and relative increase in a log-likelihood at the optimal point are different for every configuration. That is why in this experiment we set the number of iterations to 50, that was more than enough for the convergence in all cases.

The averaged EERs are presented in Fig. 1. Here, we fix the number of columns of one subspace matrix and vary the other. Our training dataset has only 578 unique speakers that is why to compare standard and simplified PLDA to the two-covariance model we applied LDA to reduce the dimensionality to be 550.

The figures clearly show that for the 600-dimensional i-vectors channel subspace should be as large as possible whereas after LDA projection the channel variability is compensated and the best performance is achieved when matrix $\mathbf{U}$ is set to zero.

Another interesting finding is that usually deviations from the standard PLDA show better performance even when they are supposed to be theoretically equivalent. It could be the result of simpler EM-algorithms with less intermediate steps and matrices.

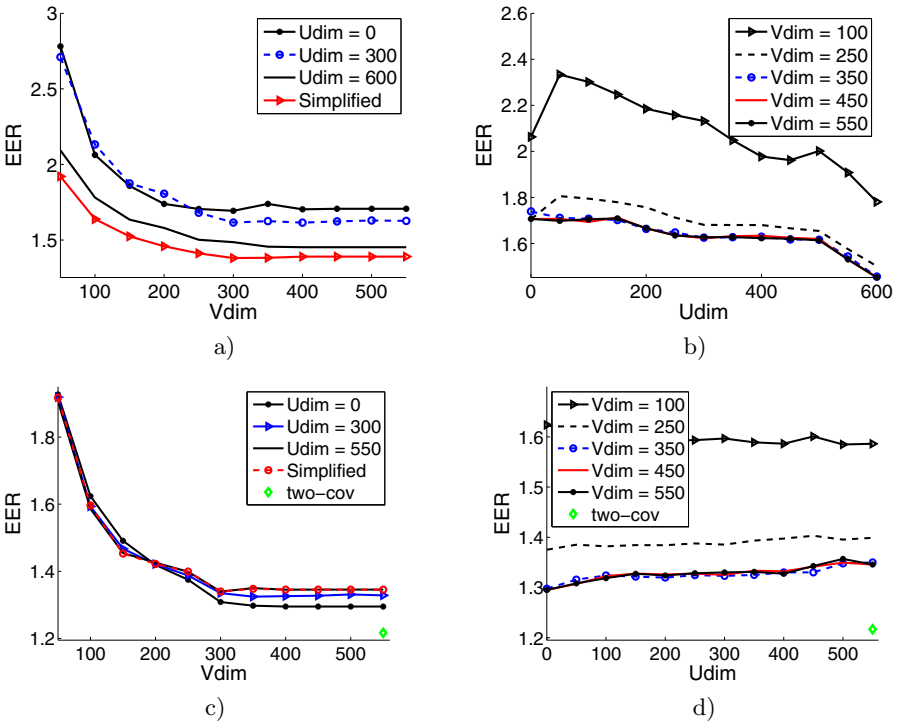


Fig. 1. Comparison of different configuration of the standard PLDA model with simplified and two-covariance models. Here, **Vdim** is a number of basis vectors for between-individual subspace (number of columns in matrix **V**), **Udim** is a number of basis vectors for within-individual subspace (number of columns in matrix **U**). Experiments a) and b) is done on the uncompressed i-vectors with 600 dimensions, c) and d) — on the LDA-projected 550-dimensional i-vectors.

5 Conclusion

We compared the standard, simplified and two-covariance PLDA variants. We have shown that the standard PLDA is the most general formulation and that, for certain configurations, it is equivalent to the other two models in terms of the predictive power. Our experimental results suggested that it is better to use the simplest possible model suited for the particular application. We presented the algorithms for all three models and shared their implementation online.

References

1. Jain, A.K., Ross, A., Pankati, S.: Biometrics: A tool for information security. IEEE-TIFS 1(2), 125–143 (2006)
2. Prince, S.J.D., Elder, J.H.: Probabilistic linear discriminant analysis for inferences about identity. In: IEEE 11th ICCV, pp. 1–8 (October 2007)

3. Kinnunen, T., Li, H.: An overview of text-independent speaker recognition: from features to supervectors. Speech communication 52(1), 12–40 (2010)
4. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus (2006)
5. Li, P., Fu, Y., Mohammed, U., Elder, J.H., Prince, S.J.D.: Probabilistic models for inference about identity. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(1), 144–157 (2012)
6. Kenny, P.: Bayesian speaker verification with heavy tailed priors. In: Proc. of the Odyssey Speak. and Lan. Recog. Workshop, Brno, Czech Republic (2010)
7. Brümmer, N., De Villiers, E.: The speaker partitioning problem. In: Proc. of the Odyssey Speak. and Lan. Recog. Workshop, Workshop, Brno, Czech Republic (2010)
8. Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech and Language Processing 19(4), 788–798 (2011)
9. Garcia-Romero, D., Espy-Wilson, C.: Analysis of i-vector length normalization in speaker recognition systems. In: Interspeech, pp. 249–252 (2011)
10. Vesnicer, B., Gros, J.Z., Pavešić, N., Štruc, V.: Face recognition using simplified probabilistic linear discriminant analysis. International Journal of Advanced Robotic Systems 9 (2012)
11. Rajan, P., Kinnunen, T., Hautamäki, V.: Effect of multicondition training on i-vector PLDA configurations for speaker recognition. In: Proc. Interspeech (2013)
12. Villalba, J.A., Brümmer, N.: Towards fully bayesian speaker recognition: Integrating out the between-speaker covariance. In: Interspeech (2011)
13. Rajan, P., Afanasyev, A., Hautamäki, V., Kinnunen, T.: From single to multiple enrollment i-vectors: Practical PLDA scoring variants for speaker verification. In: Digital Signal Processing (to appear, 2014)
14. Jiang, Y., Lee, K.A., Tang, Z., Ma, B., Larcher, A., Li, H.: PLDA modeling in i-vector and supervector space for speaker verification. In: Interspeech (2012)
15. Shafey, E.L., McCool, C., Wallace, R., Marcel, S.: A scalable formulation of probabilistic linear discriminant analysis: applied to face recognition. IEEE Trans. Pattern Anal. Mach. Intell. 35(7), 1788–1794 (2013)
16. Brümmer, N.: EM for probabilistic LDA. Technical report, Agnitio Research, Cape Town (2010), `http://sites.google.com/site/nikobrummer/`
17. Minka, T.: Old and new matrix algebra useful for statistics. Technical report. MIT (2000),
`http://research.microsoft.com/en-us/um/people/minka/papers/matrix/`
18. Campbell, W.M., Sturim, D.E., Reynolds, D.A., Solomonoff, A.: SVM based speaker verification using a GMM supervector kernel and NAP variability compensation. In: Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), vol. 1, pp. 97–100. IEEE (2006)
19. Kinnunen, T., Rajan, P.: A practical, self-adaptive voice activity detector for speaker verification with noisy telephone and microphone data. In: Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP) (2013)
20. Saeidi, R., Lee, K.A., Kinnunen, T., et al.: I4U submission to NIST SRE 2012: A large-scale collaborative effort for noise-robust speaker verification. In: Proc. Interspeech (2013)
21. Brümmer, N.: The EM algorithm and minimum divergence. Technical report, Agnitio Research, Cape Town (2009), `http://sites.google.com/site/nikobrummer/`
22. Luttinen, J., Ilin, A.: Transformations in variational bayesian factor analysis to speed up learning. Neurocomputing 73(79), 1093–1102 (2010)

A EM-Algorithm for Standard/Simplified PLDA

Suppose that we have K individuals in total and the i-th person has n_i enrolment samples $\{\boldsymbol{\phi}_{ij}\}_{j=1}^{n_i}$. It is more convenient to subtract the global mean from the data before learning the model. Let

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i,j}\boldsymbol{\phi}_{ij}\,, \tag{A.1}$$

where $N = \sum_{i=1}^{K} n_i$ is a global zero-order moment (total number of PLDA training vectors). We centralize the data

$$\boldsymbol{\varphi}_{ij} = \boldsymbol{\phi}_{ij} - \boldsymbol{\mu} \tag{A.2}$$

and define the first-order moment for the i-th person as

$$\mathbf{f}_i = \sum_{j=1}^{n_i}\boldsymbol{\varphi}_{ij}, \tag{A.3}$$

and the global second-order moment as

$$\mathbf{S} = \sum_{ij}\boldsymbol{\varphi}_{ij}\boldsymbol{\varphi}_{ij}^{\mathsf{T}}. \tag{A.4}$$

In the E-step we first pre-compute the following matrices:

$$\mathbf{Q} = (\mathbf{U}^{\mathsf{T}}\boldsymbol{\Lambda}\mathbf{U} + \mathbf{I})^{-1} \tag{A.5}$$

$$\mathbf{J} = \mathbf{U}^{\mathsf{T}}\boldsymbol{\Lambda}\mathbf{V} \tag{A.6}$$

$$\mathbf{M}_i = (n_i\mathbf{V}^{\mathsf{T}}\boldsymbol{\Lambda}(\mathbf{V} - \mathbf{U}\mathbf{Q}\mathbf{J}) + \mathbf{I})^{-1}\,, \tag{A.7}$$

where the matrices $\mathbf{U}$, $\mathbf{V}$ and $\boldsymbol{\Lambda}$ as defined in (1) and (4). After that we can easily find the first moments of the latent variables:

$$\mathbb{E}[\mathbf{y}_i] = \mathbf{M}_i(\mathbf{V} - \mathbf{U}\mathbf{Q}\mathbf{J})^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{f}_i \tag{A.8}$$

$$\mathbb{E}[\mathbf{x}_{ij}] = \mathbf{Q}(\mathbf{U}^{\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{\varphi}_{ij} - \mathbf{J}\mathbb{E}[\mathbf{y}_i]) \tag{A.9}$$

Let us define $\mathbf{z}_{ij}^{\mathsf{T}} = \left[\mathbf{y}_i^{\mathsf{T}}\ \mathbf{x}_{ij}^{\mathsf{T}}\right]$. In the M-step, we need an aggregated second moment of the compound variables $\mathbf{z}_{ij}$:

$$\mathbf{R} = \sum_{ij}\mathbb{E}\left[\begin{pmatrix}\mathbf{y}_i\\ \mathbf{x}_{ij}\end{pmatrix}\left(\mathbf{y}_i^{\mathsf{T}}\ \mathbf{x}_{ij}^{\mathsf{T}}\right)\right] = \begin{bmatrix}\mathbf{R}_{\mathbf{yy}} & \mathbf{R}_{\mathbf{yx}}\\ \mathbf{R}_{yx}^{\mathsf{T}} & \mathbf{R}_{\mathbf{xx}}\end{bmatrix} \tag{A.10}$$

where

$$\mathbb{E}[\mathbf{z}_{ij}\mathbf{z}_{ij}^{\mathsf{T}}] = \begin{bmatrix}\mathbf{M}_i & -\mathbf{M}_i\mathbf{J}^{\mathsf{T}}\mathbf{Q}^{\mathsf{T}}\\ -\mathbf{Q}\mathbf{J}\mathbf{M}_i & \mathbf{Q} + \mathbf{Q}\mathbf{J}\mathbf{M}_i\mathbf{J}^{\mathsf{T}}\mathbf{Q}^{\mathsf{T}}\end{bmatrix} + \mathbb{E}[\mathbf{z}_{ij}]\mathbb{E}[\mathbf{z}_{ij}]^{\mathsf{T}} \tag{A.11}$$

$$\mathbf{T} = \sum_{ij} \mathbb{E}[\mathbf{z}_{ij}] \boldsymbol{f}_i^{\mathsf{T}} = \sum_{ij} \mathbb{E} \begin{bmatrix} \mathbf{y}_i \\ \mathbf{x}_{ij} \end{bmatrix} \boldsymbol{f}_i^{\mathsf{T}} = \begin{bmatrix} \mathbf{T_y} \\ \mathbf{T_x} \end{bmatrix} = \begin{bmatrix} \mathbf{T_y} \\ \mathbf{Q}(\mathbf{U}^{\mathsf{T}}\boldsymbol{\Lambda}\mathbf{S} - \mathbf{J}\mathbf{T_y}) \end{bmatrix} \quad \text{(A.12)}$$

$$\mathbf{R_{yy}} = \sum_i n_i (\mathbf{M}_i + \mathbb{E}[\mathbf{y}_i]\mathbb{E}[\mathbf{y}_i]^{\mathsf{T}}) \quad \text{(A.13)}$$

$$\mathbf{R_{yx}} = (\mathbf{T_y}\boldsymbol{\Lambda}\mathbf{U} - \mathbf{R_{yy}}\mathbf{J}^{\mathsf{T}})\mathbf{Q} \quad \text{(A.14)}$$

$$\mathbf{R_{xx}} = \mathbf{Q}(\mathbf{U}^{\mathsf{T}}\boldsymbol{\Lambda}\mathbf{S}\boldsymbol{\Lambda}\mathbf{U} - \mathbf{U}^{\mathsf{T}}\boldsymbol{\Lambda}\mathbf{T}_{\mathbf{y}}^{\mathsf{T}}\mathbf{J}^{\mathsf{T}} - \mathbf{J}\mathbf{T_y}\boldsymbol{\Lambda}\mathbf{U} + \mathbf{J}\mathbf{R_{yy}}\mathbf{J}^{\mathsf{T}})\mathbf{Q} + N\mathbf{Q} \quad \text{(A.15)}$$

At the M-step we update the matrices $\mathbf{V}$, $\mathbf{U}$ and $\boldsymbol{\Lambda}$ as following

$$\begin{bmatrix} \mathbf{V}\ \mathbf{U} \end{bmatrix} = \mathbf{T}^{\mathsf{T}}\mathbf{R}^{-1} \quad \text{(A.16)}$$

$$\boldsymbol{\Lambda}^{-1} = \frac{1}{N}\mathrm{diag}\left\{\mathbf{S} - \begin{bmatrix} \mathbf{V}\ \mathbf{U} \end{bmatrix} \mathbf{T}\right\} \quad \text{(A.17)}$$

To speed up convergence it is highly recommended to apply a so-called *minimum-divergence* (MD) step as well [21], [22]. During this step we assume that a prior for the latent variables $\{\mathbf{y}_i\}$ and $\{\mathbf{x}_{ij}\}$ could be in a non-standard Gaussian form, maximize w.r.t. its parameters and then find equivalent representation but with a standard prior. This step is very efficient against saddle-points. For MD-step we need a number of auxiliary matrices:

$$\boldsymbol{\mathcal{Y}} = \frac{1}{K}\sum_i (\mathbf{M}_i + \mathbb{E}[\mathbf{y}_i]\mathbb{E}[\mathbf{y}_i]^{\mathsf{T}}), \quad \text{(A.18)}$$

$$\mathbf{G} = \mathbf{R}_{\boldsymbol{yx}}^{\mathsf{T}}\mathbf{R}_{\boldsymbol{yy}}^{-1}, \quad \text{(A.19)}$$

$$\boldsymbol{\mathcal{X}} = \frac{1}{N}(\mathbf{R_{xx}} - \mathbf{G}\mathbf{R}_{\boldsymbol{yx}}). \quad \text{(A.20)}$$

After that it is enough to apply the following transformations:

$$\mathbf{U} \leftarrow \mathbf{U}\mathrm{chol}(\boldsymbol{\mathcal{X}}), \quad \text{(A.21)}$$

$$\mathbf{V} \leftarrow \mathbf{V}\mathrm{chol}(\boldsymbol{\mathcal{Y}}) + \mathbf{U}\mathbf{G}. \quad \text{(A.22)}$$

where $\mathrm{chol}(\boldsymbol{\mathcal{X}})$ is a Cholesky decomposition of the matrix $\boldsymbol{\mathcal{X}}$. The algorithm 1 presents a compact version of the derivations above.

B EM-Algorithm for Two-Covariance Model

As before we have K individuals in total and the i-th person has n_i enrolment samples $\{\boldsymbol{\phi}_{ij}\}_{j=1}^{n_i}$. Let's define a global zero-order moment:

$$N = \sum_{i=1}^{K} n_i, \quad \text{(B.1)}$$

the first-order moment for the i-th person as

$$\mathbf{f}_i = \sum_{j=1}^{n_i} \boldsymbol{\phi}_{ij}, \quad \text{(B.2)}$$

and the global second-order moment as

$$\mathbf{S} = \sum_{ij} \boldsymbol{\phi}_{ij} \boldsymbol{\phi}_{ij}^{\mathsf{T}} \,. \tag{B.3}$$

In the E-step we first pre-compute the following matrices

$$\mathbf{L}_i = \mathbf{B} + n_i \mathbf{W} \,, \tag{B.4}$$

where the matrices $\mathbf{B}$ and $\mathbf{W}$ are defined in (8) and (9). After that we can easily find the first and second moments of the latent variables:

$$\mathbb{E}[\mathbf{y}_i] = \mathbf{L}^{-1} \boldsymbol{\gamma} \,, \tag{B.5}$$

$$\mathbb{E}[\mathbf{y}_i \mathbf{y}_i^{\mathsf{T}}] = \mathbf{L}_i^{-1} + \mathbb{E}[\mathbf{y}_i] \mathbb{E}[\mathbf{y}_i]^{\mathsf{T}}, \tag{B.6}$$

where

$$\boldsymbol{\gamma} = \mathbf{B}\boldsymbol{\mu} + \mathbf{W}\mathbf{f}_i \,. \tag{B.7}$$

At the M-step we need to compute the following matrices

$$\mathbf{T} = \sum_i \mathbb{E}[\mathbf{y}_i] \mathbf{f}_i^{\mathsf{T}} \,, \tag{B.8}$$

$$\mathbf{R} = \sum_i n_i \mathbb{E}[\mathbf{y}_i \mathbf{y}_i^{\mathsf{T}}] \,, \tag{B.9}$$

$$\boldsymbol{\mathcal{Y}} = \sum_i n_i \mathbb{E}[\mathbf{y}_i] \,. \tag{B.10}$$

After that we update the parameters $\boldsymbol{\mu}$, $\mathbf{B}$ and $\mathbf{W}$ as follows

$$\boldsymbol{\mu} = \frac{1}{N} \boldsymbol{\mathcal{Y}} \,, \tag{B.11}$$

$$\mathbf{B}^{-1} = \frac{1}{N} (\mathbf{R} - 2\boldsymbol{\mu}\boldsymbol{\mathcal{Y}}^{\mathsf{T}}) + \boldsymbol{\mu}\boldsymbol{\mu}^{\mathsf{T}} \,, \tag{B.12}$$

$$\mathbf{W}^{-1} = \frac{1}{N} (\mathbf{S} - 2\mathbf{T} + \mathbf{R}) \,. \tag{B.13}$$

The algorithm 2 presents a compact version of the derivations above.

Author Index