Edwin Hancock
Marcello Pelillo (Eds.)

LNCS 7953

Similarity-Based Pattern Recognition

Second International Workshop, SIMBAD 2013
York, UK, July 2013
Proceedings

Springer

Edwin Hancock   Marcello Pelillo (Eds.)

Similarity-Based Pattern Recognition

Second International Workshop, SIMBAD 2013
York, UK, July 3-5, 2013
Proceedings

Springer

Volume Editors

Edwin Hancock
University of York
Department of Computer Science
Deramore Lane, York, YO10 5GH, UK
E-mail: erh@cs.york.ac.uk

Marcello Pelillo
Università Ca' Foscari
DAIS
Via Torino 155, 30172 Venice, Italy
E-mail: pelillo@dsi.unive.it

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-39139-2 e-ISBN 978-3-642-39140-8
DOI 10.1007/978-3-642-39140-8
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013941045

CR Subject Classification (1998): I.4, I.5, I.2.10, I.3.5, H.3, G.2, F.1, J.3

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

This volume contains the papers presented at the Second International Workshop on Similarity-Based Pattern Analysis and Recognition, held in York, UK, during July 3–5, 2013 (SIMBAD 2013). The aim of this series of workshops, the first edition of which was held in Venice, Italy, in September 2011, is to consolidate research efforts in the area of similarity-based pattern recognition and machine learning and to provide an informal discussion forum for researchers and practitioners interested in this important yet diverse subject. The idea of running these workshops originated from the EU FP7 Project SIMBAD (http://simbad-fp7.eu), which was devoted precisely to this theme.

The call for papers produced 33 submissions, resulting in the 18 contributed papers appearing in this volume, 10 of which were presented as long (40 min) talks and 8 as short ones (20 min). We make no distinction between these two types of contributions in the book. The papers cover a wide range of problems and perspectives, from supervised to unsupervised learning, from generative to discriminative models, from theoretical issues to real-world practical applications, and offer a timely picture of the state of the art in the field.

In addition to regular, original contributions, we also solicited papers that have been recently published, or accepted for publication, elsewhere. These papers underwent the same review process as regular ones, and the accepted ones were presented at the workshop either as a long or short talk. The workshop's program included the following non-original talks that, of course, are not contained in this book:

- Balcan, M.-F., Liang, Y.: Clustering under perturbation resilience. In: Proc. ICALP 2012, Warwick, UK, pp. 63–74 (2012)
- Bonev, B., Chuang, L., Escolano, F.: How do image complexity, task demands and looking biases influence human gaze behavior? Pattern Recognition Letters 34(7), 723–730 (2013)
- Chehreghani, M.H., Busse, L.M., Buhmann, J.M.: Information-theoretic analysis of clustering algorithms
- Lourenco, A., Rota Bulò, S., Rebagliati, N., Fred, A., Figueiredo, M., Pelillo, M.: Probabilistic consensus clustering using evidence accumulation. Machine Learning (2013, in press)
- Prabhakaran, S., Metzner, K., Boehm, A., Roth, V.: Recovering networks from distance data. In: JMLR: Workshop and Conference Proceedings, vol. 25, pp. 349–364 (2012)

Finally, the workshop also featured invited keynote talks by Avrim Blum, from Carnegie Mellon University, USA, Nello Cristianini, from the University of Bristol, UK, and Frank Nielsen, from Sony Computer Science Laboratories Inc., Japan.

We would like to take this opportunity to express our gratitude to all those who helped to organize the workshop. First of all, thanks are due to the members of the Scientific Committees and to the additional reviewers. Special thanks are due to the members of the Organizing Committee. In particular, Samuel Rota Bulò managed the workshop's website and the online review system, and Luca Rossi helped assemble the proceedings.

Finally, we offer our appreciation to the editorial staff at Springer in producing this book, and for supporting the event through publication in the LNCS series. We also thank all the authors and the invited speakers for helping to make this event a success, and producing a high-quality publication to document the event.

April 2013

Edwin Hancock
Marcello Pelillo

Organization

Program Chairs

Edwin R. Hancock	University of York, UK
Marcello Pelillo	University of Venice, Italy

Steering Committee

Joachim Buhmann	ETH Zurich, Switzerland
Robert Duin	Delft University of Technology, The Netherlands
Mário Figueiredo	Technical University of Lisbon, Portugal
Edwin Hancock	University of York, UK
Vittorio Murino	Italian Institute of Technology, Italy
Marcello Pelillo (Chair)	University of Venice, Italy

Program Committee

Maria-Florina Balcan	Georgia Institute of Technology, USA
Manuele Bicego	University of Verona, Italy
Avrim Blum	Carnegie Mellon University, USA
Joachim Buhmann	ETHZ Zurich, Switzerland
Terry Caelli	NICTA, Australia
Tiberio Caetano	NICTA, Australia
Umberto Castellani	University of Verona, Italy
Luca Cazzanti	University of Washington
Nello Cristianini	University of Bristol, UK
Robert Duin	Delft University of Technology, The Netherlands
Aykut Erdem	Hacettepe University, Turkey
Francisco Escolano	University of Alicante, Spain
Mario Figueiredo	Technical University of Lisbon, Portugal
Ana Fred	Technical University of Lisbon, Portugal
Marco Gori	University of Siena, Italy
Mehmet Gőnen	Aalto University, Finland
Bernard Haasdonk	University of Stuttgart, Germany
Edwin Hancock	University of York, UK
Robert Krauthgamer	Weizmann Institute of Science, Israel
Xuelong Li	Chinese Academy of Sciences, China
Marco Loog	Delft University of Technology, The Netherlands
Marina Meila	University of Washington, USA

Vittorio Murino	Italian Institute of Technology, Italy
Frank Nielsen	Sony Computer Science Laboratories Inc., Japan
Marcello Pelillo	University of Venice, Italy
Massimiliano Pontil	University College London, UK
Antonio Robles-Kelly	NICTA, Australia
Fabio Roli	University of Cagliari, Italy
Samuel Rota Bulò	University of Venice, Italy
Volker Roth	University of Basel, Switzerland
John Shawe-Taylor	University College London, UK
Andrea Torsello	University of Venice, Italy
Richard Wilson	University of York, UK
Lior Wolf	Tel Aviv University, Israel

Additional Reviewers

Ariu, Davide
Biggio, Battista
Piras, Luca
Ram, Parikshit
Sangineto, Enver
Sheffet, Or

Table of Contents

Pattern Learning and Recognition on Statistical Manifolds: An Information-Geometric Review

Frank Nielsen

Sony Computer Science Laboratories, Inc.
Tokyo, Japan
Frank.Nielsen@acm.org
www.informationgeometry.org

Abstract. We review the *information-geometric* framework for statistical pattern recognition: First, we explain the role of statistical similarity measures and distances in fundamental statistical pattern recognition problems. We then concisely review the main statistical distances and report a novel versatile family of divergences. Depending on their intrinsic complexity, the statistical patterns are learned by either atomic parametric distributions, semi-parametric finite mixtures, or non-parametric kernel density distributions. Those statistical patterns are interpreted and handled geometrically in *statistical manifolds* either as single points, weighted sparse point sets or non-weighted dense point sets. We explain the construction of the two prominent families of statistical manifolds: The Rao Riemannian manifolds with geodesic metric distances, and the Amari-Chentsov manifolds with dual asymmetric non-metric divergences. For the latter manifolds, when considering atomic distributions from the same exponential families (including the ubiquitous Gaussian and multinomial families), we end up with dually flat exponential family manifolds that play a crucial role in many applications. We compare the advantages and disadvantages of these two approaches from the algorithmic point of view. Finally, we conclude with further perspectives on how "geometric thinking" may spur novel pattern modeling and processing paradigms.

Keywords: Statistical manifolds, mixture modeling, kernel density estimator, exponential families, clustering, Voronoi diagrams.

1 Introduction

1.1 Learning Statistical Patterns and the Cramér-Rao Lower Bound

Statistical pattern recognition [1] is concerned with *learning* patterns from observations using sensors, and with *analyzing* and *recognizing* those patterns efficiently. We shall consider three kinds of statistical models for learning patterns depending on their intrinsic complexities:

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 1–25, 2013.

1. *parametric* models: A pattern is an atomic parametric distribution,
2. *semi-parametric models*: A pattern is a finite mixture of parametric distributions, and
3. *non-parametric models*: A pattern is a kernel density distribution.

Given a set of n observations $\{x_1, ..., x_n\}$, we may estimate the pattern parameter λ of the atomic distribution $p(x; \lambda)$ by using the *maximum likelihood principle*. The maximum likelihood estimator (MLE) proceeds by defining a function $L(\lambda; x_1, ..., x_n)$, called the *likelihood function* and maximizes this function with respect to λ. Since the sample is usually assumed to be *identically and independently distributed* (iid.), we have:

$$L(\lambda; x_1, ..., x_n) = \prod_i p(x_i; \lambda).$$

This maximization is equivalent (but mathematically often more convenient) to maximize the *log-likelihood function*:

$$l(\lambda; x_1, ..., x_n) = \log L(\lambda; x_1, ..., x_n) = \sum_i \log p(x_i; \lambda).$$

This maximization problem amounts to set the gradient to zero: $\nabla l(\lambda; x_1, ..., x_n) = 0$, and solve for the estimated quantity $\hat{\lambda}$ provided that it is well-defined (ie., that ML does not diverge to ∞). We can view the MLE as a function $\hat{\lambda}(X_1, ..., X_n)$ on a *random vector* and ask for its statistical performance. (Indeed, we can build a family of moment estimators by matching the sample l-th moments with the distribution l-th moments. This raises the question to compare them by analyzing, say, their variance characteristics.) Cramér [2], Fréchet [3] and Rao [4] independently proved a lower bound on the variance of any *unbiased* estimator $\hat{\lambda}$:

$$V[\hat{\lambda}] \succeq I(\lambda)^{-1},$$

where $\succeq$ denotes the Löwner partial ordering[1] on positive semidefinite matrices, and matrix $I(\lambda)$ is called the *Fisher information* matrix:

$$I(\lambda) = [I_{ij}(\lambda)], \quad I_{ij}(\lambda) = E[\partial_i l(x; \lambda) \partial_j l(x; \lambda)],$$

with ∂_k the shortcut notation: $\partial_k = \frac{\partial}{\partial \lambda_k}$. The Fisher information matrix [5] (FIM) is the variance of the score function $s(\lambda) = \nabla_\lambda \log p(\lambda; x)$: $I(\lambda) = V[s(\lambda)]$. This lower bound holds under very mild regularity conditions.

Learning finite mixtures of k atomic distributions is traditionally done using the *Expectation-Maximization* algorithm [6]. Learning a non-parametric distribution using a *kernel density estimator* (KDE) proceeds by choosing a kernel (e.g., Gaussian kernel), and by then fitting a kernel at each sample observation

[1] A symmetric matrix X is positive definite if and only if $\forall x \neq 0, x^\top X x > 0$, and $A \succeq B$ iff. $A - B \succ 0$. When the inequality is relaxed to include equality, we have the semi-positive definiteness property.

(controlling adaptively the kernel window is important in practice). Those three ML estimation/EM/KDE algorithms will be explained using the framework of information geometry in Section 5 when considering dually flat statistical *exponential family manifolds* (EFMs).

We now describe briefly the fundamental tasks of pattern recognition using eiher the *unsupervised setting* or the *supervised setting.* We recommend the introductory textbook [7] of Fukunaga for further explanations.

1.2 Unsupervised Pattern Recognition

Given a collection of n statistical patterns represented by their distributions (or estimated parameters $\lambda_1, ..., \lambda_n$), we would like to *categorize* them. That is, to identify groups (or clusters) of patterns inducing pattern categories. This is typically done using *clustering* algorithms. Observe that since patterns are represented by probability distributions, we need to have clustering algorithms suited to statistical distributions: Namely, clustering algorithms tailored for *information spaces.* We shall explain and describe the notions of statistical distances in information spaces in the following Section.

1.3 Supervised Pattern Recognition

When we are given beforehand a *training set* of properly *labeled* (or annotated) patterns, and seek to classify incoming online patterns, we may choose to label that *query pattern* with the label of its most similar annotated pattern in the database, or to vote by considering the k "nearest" patterns. Again, this requires a notion of statistical similarity that is described in Section 2.

1.4 Core Geometric Structures and Algorithmic Toolboxes

Since we are going to focus on two types of construction for defining *statistical manifolds of patterns*, let us review the wish list tools required by supervised or unsupervised pattern recognition. We need among others:

- Clustering (e.g., hard clustering à la k-means) with respect to statistical distances for unsupervised category discovery,
- To study the statistical Voronoi diagrams induced by the distinct category patterns,
- Data-structures for performing efficiently k-NN (nearest neighbor) search with respect to statistical distances (say, ball trees [8] or vantage point trees [9]),
- To study minimum enclosing balls (MEB) [10,11,12,13] (with applications in machine learning using vector ball machines [14])
- Etc.

1.5 Outline of the Paper

The paper is organized as follows: In Section 2, we review the main statistical divergences, starting from the seminal Kullback-Leibler divergence, and explain why and how the intractable *distribution intersection similarity* measure needs to be upper bounded. This allows to explain the genesis of the Bhattacharyya divergence, the Chernoff information and the family of α-divergences. Following this interpretation, we further present the novel concept of *quasi-arithmetic* α-*divergences* and *quasi-arithmetic Chernoff informations*. Section 3 recalls that geometry is grounded by notion of invariance, and introduces the concepts of statistical invariance with the class of Ali-Silvey-Csiszár f-divergences [15,16]. We then describe two classical statistical manifold constructions: In Section 4, we present the *Rao Riemannian manifold* and discuss on its algorithmic considerations. In Section 5, we describe the *dual affine Amari-Chentsov manifolds*, and explain the process of learning parametric/semi-parametric/non-parametric patterns on those manifolds. Finally, Section 6 wrap ups this review paper and hints at further perspectives in the realm of statistical pattern analysis and recognition.

2 Statistical Distances and Divergences

2.1 The Fundamental Kullback-Leibler Divergence

The Kullback-Leibler divergence between two probability distributions $P(x)$ and $Q(x)$ (with density $p(x)$ and $q(x)$ with respect to a measure ν) is equal to the cross-entropy $H^\times(P:Q)$ minus the Shannon entropy $H(P)$:

$$\mathrm{KL}(P:Q) = \int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}\nu(x) = H^\times(P:Q) - H(P) \geq 0,$$

with

$$H^\times(P:Q) = \int -p(x)\log q(x) \mathrm{d}\nu(x),$$

$$H(P) = \int -p(x) \log p(x) \mathrm{d}\nu(x) = H^\times(P:P).$$

In practice, the Kullback-Leibler divergence $\mathrm{KL}(\tilde{P}:P)$ [17] can be interpreted as the distance between the *estimated distribution* $\tilde{P}$ (derived from the observed samples) and the *true hidden* distribution P. The Kullback-Leibler divergence does not satisfy the metric axioms of symmetry and triangular inequality. Therefore we call this dissimilarity[2] measure a *divergence* as it is a smooth and differentiable distance function that satisfies the essential separability property: $\mathrm{KL}(P:Q)=0$ if and only if $P=Q$. Computing the Kullback-Leibler may not be tractable analytically (eg., for patterns modeled by mixtures or KDEs)

[2] Note that there are Finslerian distances [34] that preserve the triangular inequality without being symmetric.

and requires costly Monte-Carlo stochastic approximation algorithms to estimate. To bypass this computational obstacle, several alternative distances like the Cauchy-Schwarz divergences [18] have been proposed. Since the inception of the Kullback-Leibler divergence, many other statistical distances have been proposed. We shall review in the context of classification the most prominent divergences.

2.2 Genesis of Statistical Distances

How can we define a notion of "distance" between two probability distributions P_1 and P_2 sharing the same support $\mathcal{X}$ with respective density p_1 and p_2 with respect to a dominating measure ν? What is the meaning of defining statistical distances? A distance $D(\cdot,\cdot)$ can be understood as a non-negative *dissimilarity measure* $D(P_1, P_2) \geq 0$ that is related to the notion of a *similarity measure* $0 < S(P_1, P_2) \leq 1$. We present an overview of statistical distances based on the framework of *Bayesian binary hypothesis testing* [7].

Consider *discriminating* P_1 and P_2 with the following classification problem based on the mixture $P = \frac{1}{2}P_1 + \frac{1}{2}P_2$. To sample mixture P, we first toss an unbiased coin and choose to sample from P_1 if the coin fell on heads or to sample from P_2 if it fell on tails. Thus mixture sampling is a *doubly stochastic process.* Now, given a *random variate* x of P (i.e., an observation) we would like to *decide* whether x was sampled from P_1 or from P_2? It makes sense to label x as class C_1 if $p_1(x) > p_2(x)$ and as class C_2, otherwise (if $p_2(x) \geq p_1(x)$). Since the distribution supports of P_1 and P_2 coincide, we can *never* be certain, and shall find a decision rule to minimize the risk. We seek for the best decision rule that minimizes the *probability of error* P_e, that is, the probability of misclassification. Consider the decision rule based on the *log-likelihood ratio* $\log \frac{p_1(x)}{p_2(x)}$:

$$\log \frac{p_1(x)}{p_2(x)} \overset{C_2}{\underset{C_1}{\lesseqgtr}} 0.$$

The expected probability of error is:

$$P_e = E_P[\text{error}(x)] = \int_{x\in\mathcal{X}} \text{error}(x) p(x) \mathrm{d}\nu(x),$$

where $p(x) = \frac{1}{2}p_1(x) + \frac{1}{2}p_2(x)$ denotes the mixture density, and

$$\text{error}(x) = \min\left(\frac{1}{2}\frac{p_1(x)}{p(x)}, \frac{1}{2}\frac{p_2(x)}{p(x)}\right).$$

Indeed, suppose that at x (with probability $\frac{1}{2}$), $p_1(x) < p_2(x)$. Since we label x as C_2 then we misclassify with proportion $\frac{p_1(x)}{p(x)}$, and vice-versa [7]. Thus the probability of error $P_e = \frac{1}{2}S(P_1, P_2)$ where:

$$S(P_1, P_2) = \int \min(p_1(x), p_2(x)) \mathrm{d}\nu(x).$$

S is a similarity measure since $S(P_1, P_2) = 1$ if and only if $P_1 = P_2$. It is known in computer vision, in the discrete case, as the *histogram intersection* similarity [19].

In practice, computing S is not tractable[3], specially for multivariate distributions. Thus, we seek to *upper bound* S using mathematically convenient tricks purposely designed for large classes of probability distributions. Consider the case of exponential families [20] that includes most common distributions such as Poisson, Gaussian, Gamma, Beta, Dirichlet, etc. distributions. Their natural canonical density decomposition is:

$$p_i = p(x|\theta_i) = \exp(\langle \theta_i, t(x)\rangle - F(\theta_i) + k(x)),$$

where θ_i is the natural parameter belonging to natural parameter space Θ. Function F is strictly convex and characterize the family. $t(x)$ is the sufficient statistic and $k(x)$ is an auxiliary carrier term [20]. Table 1 summarizes the canonical decomposition and related results for the multinomial and Gaussian families, with $p_i = p(x|\lambda_i) = p(x|\theta(\lambda_i))$. We can upper bound the probability intersection similarity S using the fact that:

$$\min(p_1(x), p_2(x)) \leq \sqrt{p_1(x)p_2(x)}.$$

We get:

$$S(P_1, P_2) \leq \rho(P_1, P_2) = \int \sqrt{p_1(x)p_2(x)}\mathrm{d}\nu(x).$$

The right hand-side is called the *Bhattacharrya coefficient* or *Bhattacharrya affinity*. For distributions belonging to the same exponential family (e.g., P_1 and P_2 are multivariate Gaussians [20]), we have:

$$\rho(P_1, P_2) = e^{-J_F(\theta_1, \theta_2)},$$

where J_F is a *Jensen divergence* defined over the natural parameter space:

$$J_F(\theta_1, \theta_2) = \frac{F(\theta_1) + F(\theta_2)}{2} - F\left(\frac{\theta_1 + \theta_2}{2}\right) \geq 0.$$

Of course, the bound is not the tightest. Therefore, we may consider for $\alpha \in (0, 1)$ that $\min(p_1(x), p_2(x)) \leq p_1(x)^\alpha p_2(x)^{1-\alpha}$. It follows the *$\alpha$-skewed Bhattacharrya coefficient* upper bounding S:

$$S(P_1, P_2) \leq \rho_\alpha(P_1, P_2) = \int p_1(x)^\alpha p_2(x)^{1-\alpha}\mathrm{d}\nu(x).$$

[3] In fact, using the mathematical rewriting trick $\min(a, b) = \frac{a+b}{2} - \frac{1}{2}|b - a|$, the probability intersection similarity is related to computing the *total variation metric distance*: $S(P_1, P_2) = 1 - \mathrm{TV}(P_1, P_2)$, with $\mathrm{TV}(P_1, P_2) = \frac{1}{2}\int |p_1(x) - p_2(x)|\mathrm{d}\nu(x)$. Bayes error that relies on a *cost design matrix* [7] to account for the different correct/incorrect classification costs extends the concept of the probability of error. Similarly, Bayes error can also be expressed using total variation distance on scaled probabilities (with scales depending on the prior mixture weights and on the cost design matrix).

This definition of affinity coefficient is still mathematically convenient for exponential families since we find that [20]:

$$\rho_\alpha(P_1, P_2) = e^{-J_F^{(\alpha)}(\theta_1,\theta_2)},$$

where $J_F^{(\alpha)}$ denotes a *skewed Jensen divergence* defined on the corresponding natural parameters:

$$J_F^{(\alpha)}(\theta_1, \theta_2) = \alpha F(\theta_1) + (1-\alpha)F(\theta_2) - F(\alpha\theta_1 + (1-\alpha)\theta_2) \geq 0,$$

with equality to zero if and only if $\theta_1 = \theta_2$ since F is a strictly convex and differentiable function. Setting $\alpha = \frac{1}{2}$, we get back the Bhattacharrya coefficient.

The upper bound can thus be "best" improved by optimizing over the α-range in $(0, 1)$:

$$S(P_1, P_2) \leq \min_{\alpha\in[0,1]} \rho_\alpha(P_1, P_2) = \rho_{\alpha^*}(P_1, P_2)$$

The optimal value α^* is called *best error exponent* in Bayesian hypothesis testing [7]. For an iid. sequence of n observations, the probability of error is thus bounded [21] by:

$$P_e^{(n)} \leq \frac{1}{2}\rho_{\alpha^*}^n(P_1, P_2)$$

Historically, those similarity or affinity coefficients upper bounding the probability intersection similarity yielded respective notions of *statistical distances*:

$$B_\alpha(P_1, P_2) = -\log \rho_\alpha(P_1, P_2) = J_F^{(\alpha)}(\theta_1, \theta_2),$$

the *skew Bhattacharyya divergences*. Let us rescale B_α by a factor $\frac{1}{\alpha(1-\alpha)}$, then we have for $\alpha \not\in \{0, 1\}$:

$$B'_\alpha(P_1, P_2) = \frac{1}{\alpha(1-\alpha)}B_\alpha(P_1, P_2) = \frac{1}{\alpha(1-\alpha)}J_F^{(\alpha)}(\theta_1, \theta_2) = {J'}_F^{(\alpha)}(\theta_1, \theta_2).$$

When $\alpha \to 1$ or $\alpha \to 0$, we have B'_α that tends to the direct or reverse Kullback-Leibler divergence. For exponential families, that means that the scaled skew Jensen divergences ${J'}_F^{(\alpha)}$ tends to the direct or reverse Bregman divergence [22]:

$$\lim_{\alpha\to 0} {J'_F}^{(\alpha)}(\theta_1, \theta_2) = B_F(\theta_1, \theta_2),$$

where a Bregman divergence is defined for a strictly convex and differentiable genetor F by:

$$B_F(\theta_1, \theta_2) = F(\theta_1) - F(\theta_2) - (\theta_1 - \theta_2)^\top \nabla F(\theta_2).$$

Furthermore, the *Chernoff divergence* (historically called *Chernoff information*) is defined by:

$$C(P_1, P_2) = \max_{\alpha\in[0,1]} -\log \rho_\alpha(P_1, P_2) = B_{\alpha^*}(P_1, P_2)$$

The mapping of a similarity coefficient by the monotonous function $-\log(\cdot)$ mimicked the unbounded property of the Kullback-Leibler divergence. However, we can also map a similarity coefficient $S \in (0,1]$ to a distance $D \in [0,1)$ by simply defining:

$$D(P_1,P_2) = 1 - S(P_1,P_2)$$

For example, we can define $d_\alpha(P_1,P_2) = 1 - \rho_\alpha(P_1,P_2)$. Since distances are used relatively to compare distributions and rank them as nearer or farther away, we can also rescale them. Another mathematical convenience is to scale d_α by $\frac{1}{\alpha(1-\alpha)}$ so that we get:

$$D_\alpha(P_1,P_2) = \frac{1-\rho_\alpha(P_1,P_2)}{\alpha(1-\alpha)} = \frac{1-\int p(x)^\alpha q(x)^{1-\alpha}\mathrm{d}\nu(x)}{\alpha(1-\alpha)}$$

This is known as the α-divergences of Amari that are the canonical divergences in information geometry [23]. When $\alpha \to 1$, we get the Kullback-Leibler divergence. When $\alpha \to 0$, we get the reverse Kullback-Leibler divergence. When $\alpha = \frac{1}{2}$, we find the (scaled) squared of the Hellinger distance. In information geometry, it is customary to set $\alpha \in [-\frac{1}{2}, \frac{1}{2}]$ instead of $[0,1]$ by remapping $\alpha \leftarrow \alpha - \frac{1}{2}$. For members P_1 and P_2 belonging to the same exponential family, we have the following closed-formula for the α-divergences:

$$A_\alpha(P:Q) = \frac{4}{1-\alpha^2}\left(1 - \int_{x\in\mathcal{X}} p^{\frac{1-\alpha}{2}}(x) q^{\frac{1+\alpha}{2}}\mathrm{d}x\right),$$

$$A_\alpha(P:Q) = \frac{4}{1-\alpha^2}\left(1 - e^{-J_F^{\left(\frac{1-\alpha}{2}\right)}}(\theta(P):\theta(Q))\right).$$

2.3 Novel Quasi-Arithmetic α-Divergences and Chernoff Information

Note that we can design many similar divergences by similarly upper bounding the probability intersection histogram similarity S. By definition, a *weighted mean* should have the property that it lies inside the range of its elements. Thus we can bound $\min(a,b)$ by *any* other kind of weighted means:

$$\min(a,b) \leq M(a,b;\alpha),$$

with $\alpha \in [0,1]$. Instead of bounding S by a geometric weighted mean, let us consider for a strictly monotonous function f the *quasi-arithmetic weighted means*:

$$M_f(a,b;\alpha) = f^{-1}(\alpha f(a) + (1-\alpha) f(b)).$$

We get:

$$S(P_1,P_2) \leq \rho_\alpha^{(f)}(P_1,P_2) = \int M_f(p_1(x), p_2(x);\alpha)\mathrm{d}\nu(x),$$

for $\alpha \in (0,1)$, since the extremities $\alpha = 0, 1$ are not discriminative:

$$\rho_0^{(f)}(P_1, P_2) = \rho_1^{(f)}(P_1, P_2) = 1.$$

When distributions coincide, notice that we have maximal affinity: $\rho_\alpha^{(f)}$ $(P, P) = 1$.

Similarly, we can also generalize the Chernoff information to *quasi-arithmetic f-Chernoff information* as follows:

$$C_f(P_1, P_2) = \max_{\alpha \in [0,1]} -\log \int M_f(p_1(x), p_2(x)) \mathrm{d}\nu(x).$$

For example, if we consider distributions not belonging to the exponential families like the univariate Cauchy distributions or the multivariate t-distributions (related to the unnormalized Pearson type VII elliptical distributions), in order to find a closed-form expression for $\int M_f(p_1(x), p_2(x))\mathrm{d}\nu(x)$, we may choose the *harmonic mean* with $f(x) = \frac{1}{x} = f^{-1}(x)$ instead of the geometric weighted mean.

To summarize, we have explained how the canonical α-divergences upper bounding the probability of error have been designed to include the sided (i.e., direct and reverse) Kullback-Leibler divergence, and explained the notion of probability separability using a binary classification task. We now turn our focus to build geometries for modeling statistical manifolds.

3 Divergence, Invariance and Geometry

In Euclidean geometry, we are familiar with the *invariant group* of *rigid transformations* (translations, rotations and reflections). The Euclidean distance $d(P_1, P_2)$ of two points P_1 and P_2 does not change if we apply such a rigid transformation T on their respective representations p_1 and p_2:

$$d(P_1, P_2) = d(p_1, p_2) = d(T(p_1), T(p_2)).$$

In fact, when we compute the distance between two points P_1 and P_2, we should not worry about the origin. Distance computations require numerical attributes that nevertheless should be invariant of the underlying geometry. Points exist beyond a specific coordinate system. This geometric invariance principle by a group of action has been carefully studied by Felix Klein in his *Erlangen* program.

A *divergence* is basically a smooth C_2 function (statistical distance) that may not be symmetric nor satisfy the triangular inequality of metrics. We denote by $D(P : Q)$ the divergence from distribution P (with density $p(x)$) to distribution Q (with density $q(x)$), where the ":" notation emphasizes the fact that this dissimilarity measure may not be symmetric: $D(P : Q) \neq D(Q : P)$.

It is proven that the only *statistical invariant* divergences [23,24] are the Ali-Silvey-Csiszár f-divergences D_f [15,16] that are defined for a functional convex generator f satisfying $f(1) = f'(1) = 0$ and $f''(1) = 1$ by:

Table 1. Summary of the canonical decompositions and its related results for the two prominent multivariate exponential families [20] met in statistical pattern recognition: The multinomial (one trial) and the Gaussian distributions. Observe that both families have their log-normalizer F *not* separable. For the MLE, we have that $\sqrt{n}(\hat{\theta} - \theta)$ that converges in distribution to $N(0, I^{-1}(\theta))$.

	Multinomial ($n = 1$ trial)	Multivariate Gaussian
density	$\frac{1}{x_1!\cdots x_k!}p_1^{x_1}\cdots p_k^{x_k}$	$\frac{1}{(2\pi)^{d/2}\lvert\Sigma\rvert^{1/2}}\exp\left(-\frac{(x-\mu)^\top\Sigma^{-1}(x-\mu)}{2}\right)$
support	$\{E_1, ..., E_k\}$	$\mathbb{R}^d$
base measure	Counting measure	Lebesgue measure
auxiliary carrier $k(x)$	$-\sum_{i=1}^k \log x_i!$	0
sufficient statistics $t(x)$	$(x_1, \cdots, x_{k-1})$	$(x, -xx^\top)$
$\theta(\lambda)$	$\left(\log\left(\frac{p_i}{p_k}\right)\right)_i$	$\left(\Sigma^{-1}\mu, \frac{1}{2}\Sigma^{-1}\right)$
Order D	$d-1$	$\frac{d}{d+3}2$
log-normalizer $F(\theta)$	$\log\left(1 + \sum_{i=1}^{k-1}\exp\theta_i\right)$	$\frac{1}{4}\mathrm{tr}(\theta_2^{-1}\theta_1\theta_1^\top) - \frac{1}{2}\log\lvert\theta_2\rvert + \frac{d}{2}\log\pi$
$\eta = \nabla F(\theta) = E[t(X)]$	$\left(\frac{\exp\theta_i}{1+\sum_{j=1}^{k-1}\exp\theta_j}\right)_i$	$\left(\frac{1}{2}\theta_2^{-1}\theta_1, -\frac{1}{2}\theta_2^{-1} - \frac{1}{4}(\theta_2^{-1}\theta_1)(\theta_2^{-1}\theta_1)^\top\right)$
$\theta = \nabla F^*(\eta)$	$\left(\log\left(\frac{\eta_i}{1-\sum_{j=1}^{k-1}\eta_j}\right)\right)_i$	$\left(-(\eta_2 + \eta_1\eta_1^\top)^{-1}\eta, -\frac{1}{2}(\eta_2 + \eta_1\eta_1^\top)^{-1}\right)$
$F^*(\eta) = \langle\theta, \eta\rangle - F(\theta)$	$\left(\sum_{i=1}^{k-1}\eta_i\log\eta_i\right) + \left(1 - \sum_{i=1}^{k-1}\eta_i\right)\log\left(1 - \sum_{i=1}^{k-1}\eta_i\right)$	$-\frac{1}{2}\log\left(1 + \eta_1^\top\eta_2^{-1}\eta_1\right) - \frac{1}{2}\log\lvert-\eta_2\rvert - \frac{d}{2}\log(2\pi e)$
Kullback-Leibler divergence	$p_{1,k}\log\frac{p_{1,k}}{p_{2,k}} - \sum_{i=1}^{k-1}p_{1,i}\log\frac{p_{2,i}}{p_{1,i}}$	$\frac{1}{2}\left(\log\left(\frac{\lvert\Sigma_2\rvert}{\lvert\Sigma_1\rvert}\right) + \mathrm{tr}\left(\Sigma_2^{-1}\Sigma_1\right)\right) + \frac{1}{2}\left((\mu_2 - \mu_1)^\top\Sigma_2^{-1}(\mu_2 - \mu_1) - d\right)$
Fisher information $I(\theta) = \nabla^2 F(\theta)$	$I_{ii} = \frac{1}{p_i} + \frac{1}{p_d}, \quad I_{ij} = \frac{1}{p_d}(i \neq j)$	$I_{ij} = \partial_i\mu^\top\Sigma^{-1}\partial_j\mu$
MLE $\hat{\eta} = \bar{t}$	$\hat{p}_i = \frac{n_i}{n}$	$\hat{\mu} = \frac{1}{n}\sum_{i=1}^n x_i \qquad \hat{\Sigma} = \frac{1}{n}\sum_{i=1}^n(x_i - \hat{\mu})(x_i - \hat{\mu})\top$
$\lambda(\theta)$	$\begin{cases} p_i = \frac{\exp\theta_i}{1+\sum_{j=1}^{k-1}\exp\theta_j} & \text{if } i < k \\ p_k = \frac{1}{1+\sum_{j=1}^{k-1}\exp\theta_j} \end{cases}$	$\left(\frac{1}{2}\theta_2^{-1}\theta, \frac{1}{2}\theta_2^{-1}\right)$
$\lambda(\eta)$	$\begin{cases} p_i = \eta_i & \text{if } i < k \\ p_k = 1 - \sum_{j=1}^{k-1}\eta_j \end{cases}$	$\left(\eta, -(\eta_2 + \eta\eta^\top)\right)$

$$D_f(P:Q)=\int_{x\in\mathcal{X}} p(x) f\left(\frac{q(x)}{p(x)}\right)\mathrm{d}\nu(x).$$

Indeed, under an invertible mapping function (with $\dim(\mathcal{X})=\dim(\mathcal{Y})=d$):

$$\begin{aligned} m:\mathcal{X}&\rightarrow\mathcal{Y}\\ x&\mapsto y=m(x)\end{aligned}$$

a probability density $p(x)$ is converted into another probability density $q(y)$ such that:

$$p(x)\mathrm{d}x=q(y)\mathrm{d}y,\qquad \mathrm{d}y=|M(x)|\mathrm{d}x,$$

where $|M(x)|$ denotes the determinant of the Jacobian matrix [23] of the transformation m (i.e., the partial derivatives):

$$M(x)=\begin{bmatrix}\frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_d}\\ \vdots & \ddots & \vdots\\ \frac{\partial y_d}{\partial x_1} & \cdots & \frac{\partial y_d}{\partial x_d}\end{bmatrix}.$$

It follows that we have:

$$q(y)=q(m(x))=p(x)|M(x)|^{-1}.$$

For any two densities p_1 and p_2, we have the f-divergence on the transformed densities q_1 and q_2 that can be rewritten mathematically as:

$$\begin{aligned} D_f(q_1:q_2) &= \int_{y\in\mathcal{Y}} q_1(y) f\left(\frac{q_2(y)}{q_1(y)}\right)\mathrm{d}y,\\ &= \int_{x\in\mathcal{X}} p_1(x)|M(x)|^{-1} f\left(\frac{p_2(x)}{p_1(x)}\right)|M(x)|\mathrm{d}x,\\ &= D_f(p_1:p_2).\end{aligned}$$

Furthermore, the f-divergences are the only divergences satisfying the *data-processing theorem* [25]. This theorem characterizes the property of *information monotonicity* [26]. Consider discrete distributions on an alphabet $\mathcal{X}$ of d letters. For any partition $\mathcal{B}=\mathcal{X}_1\cup\ldots\mathcal{X}_b$ of $\mathcal{X}$ that merge alphabet letters into $b\leq d$ bins, we have

$$0\leq D_f(\bar{p}_1:\bar{p}_2)\leq D_f(p_1:p_2),$$

where $\bar{p}_1$ and $\bar{p}_2$ are the discrete distribution induced by the partition $\mathcal{B}$ on $\mathcal{X}$. That is, we loose discrimination power by coarse-graining the support of the distributions. The most fundamental f-divergence is the Kullback-Leibler divergence [17] obtained for the generator $f(x)=x\log x$: In general, statistical invariance is characterized under *Markov morphisms* [27,24] (also called *sufficient stochastic kernels* [24]) that generalizes the deterministic transformations $y=m(x)$. Loosely speaking, a geometric parametric statistical manifold $\mathcal{F}=\{p_\theta(x)|\theta\in\Theta\}$ equipped with a f-divergence must also provide invariance by:

Non-singular Parameter Re-Parameterization. That is, if we choose a different coordinate system, say $\theta' = f(\theta)$ for an invertible transformation f, it should not impact the intrinsic distance between the underlying distributions. For example, whether we parametrize the Gaussian manifold by $\theta = (\mu, \sigma)$ or by $\theta' = (\mu^5, \sigma^4)$, it should preserve the distance.

Sufficient Statistic. When making statistical inference, we use statistics $T : \mathbb{R}^d \to \Theta \subseteq \mathbb{R}^D$ (e.g., the mean statistic $T_n(X) = \frac{1}{n}\sum_{i=1}^n X_i$ is used for estimating the parameter μ of Gaussians). In statistics, the concept of *sufficiency* was introduced by Fisher [28]:
Mathematically, the fact that all information should be aggregated inside the sufficient statistic is written as

$$\Pr(x|t, \theta) = \Pr(x|t).$$

It is not surprising that all statistical information of a parametric distribution with D parameters can be recovered from a set of D statistics. For example, the univariate Gaussian with $d = \dim(\mathcal{X}) = 1$ and $D = \dim(\Theta) = 2$ (for parameters $\theta = (\mu, \sigma)$) is recovered from the mean and variance statistics. A sufficient statistic is a set of statistics that compress *information without loss* for statistical inference.

4 Rao Statistical Manifolds: A Riemannian Approach

4.1 Riemannian Construction of Rao Manifolds

We review the construction first reported in 1945 by C.R. Rao [4]. Consider a family of parametric probability distribution $\{p_\theta(x)\}_\theta$ with $x \in \mathbb{R}^d$ (dimension of the support) and $\theta \in \mathbb{R}^D$ denoting the D-dimensional parameters of the distributions. It is called the order of the probability family. The *population parameter space* is defined by:

$$\Theta = \left\{ \theta \in \mathbb{R}^D \middle| \int p_\theta(x)\mathrm{d}x = 1 \right\}.$$

A given distribution $p_\theta(x)$ is interpreted as a corresponding point indexed by $\theta \in \mathbb{R}^D$. θ also encodes a coordinate system to identify probability models: $\theta \leftrightarrow p_\theta(x)$.

Consider now two infinitesimally close points θ and $\theta + \mathrm{d}\theta$. Their probability densities differ by their first order differentials: $\mathrm{d}p(\theta)$. The distribution of $\mathrm{d}p$ over all the support aggregates the consequences of replacing θ by $\theta + \mathrm{d}\theta$. Rao's revolutionary idea was to consider the *relative discrepancy* $\frac{\mathrm{d}p}{p}$ and to take the variance of this difference distribution to define the following *quadratic differential form*:

$$\begin{aligned}\mathrm{d}s^2(\theta) &= \sum_{i=1}^D \sum_{j=1}^D g_{ij}(\theta)\mathrm{d}\theta_i \mathrm{d}\theta_j,\\ &= (\nabla\theta)^\top G(\theta)\nabla\theta,\end{aligned}$$

with the matrix entries of $G(\theta) = [g_{ij}(\theta)]$ as

$$g_{ij}(\theta) = E_\theta \left[\frac{1}{p(\theta)} \frac{\partial p}{\partial \theta_i} \frac{1}{p(\theta)} \frac{\partial p}{\partial \theta_j} \right] = g_{ji}(\theta).$$

In differential geometry, we often use the symbol ∂_i as a shortcut to $\frac{\partial}{\partial \theta_i}$.

The elements $g_{ij}(\theta)$ form the quadratic differential form defining the elementary length of Riemannian geometry. The matrix $G(\theta) = [g_{ij}(\theta)] \succ 0$ is positive definite and turns out to be equivalent to the *Fisher information matrix*: $G(\theta) = I(\theta)$. The information matrix is invariant to monotonous transformations of the parameter space [4] and makes it a good candidate for a Riemannian metric as the concepts of the concepts of invariance in statistical manifolds[29,27] later was revealed.

4.2 Rao Riemannian Geodesic Metric Distance

Let P_1 and P_2 be two points of the population space corresponding to the distributions with respective parameters θ_1 and θ_2. In Riemannian geometry, the geodesics are the *shortest paths*. The statistical distance between the two populations is defined by integrating the infinitesimal element lengths ds along the geodesic linking P_1 and P_2. Equipped with the Fisher information matrix tensor $I(\theta)$, the *Rao distance* $D(\cdot, \cdot)$ between two distributions on a statistical manifold can be calculated from the geodesic length as follows:

$$D(p_{\theta_1}(x), p_{\theta_2}(x)) = \min_{\substack{\theta(t) \\ \theta(0)=\theta_1, \theta(1)=\theta_2}} \int_0^1 \left(\sqrt{(\nabla\theta)^\top I(\theta) \nabla\theta} \right) \mathrm{d}t \qquad (1)$$

Therefore we need to calculate explicitly the geodesic linking $p_{\theta_1}(x)$ to $p_{\theta_2}(x)$ to compute Rao's distance. This is done by solving the following second order ordinary differential equation (ODE) [23]:

$$g_{ki}\ddot{\theta}_i + \Gamma_{k,ij}\dot{\theta}_i\dot{\theta}_j = 0,$$

where Einstein summation [23] convention has been used to simplify the mathematical writing by removing the leading sum symbols. The coefficients $\Gamma_{k,ij}$ are the Christoffel symbols of the first kind defined by:

$$\Gamma_{k,ij} = \frac{1}{2} \left(\frac{\partial g_{ik}}{\partial \theta_j} + \frac{\partial g_{kj}}{\partial \theta_i} - \frac{\partial g_{ij}}{\partial \theta_k} \right).$$

For a parametric statistical manifold with D parameters, there are D^3 Christoffel symbols. In practice, it is difficult to explicitly compute the geodesics of the Fisher-Rao geometry of arbitrary models, and one needs to perform a gradient descent to find a local solution for the geodesics [30]. This is a drawback of the Rao's distance as it has to be checked manually whether the integral admits a closed-form expression or not.

To give an example of the Rao distance, consider the smooth manifold of univariate normal distributions, indexed by the $\theta = (\mu, \sigma)$ coordinate system. The Fisher information matrix is

$$I(\theta) = \begin{bmatrix} \frac{1}{\sigma^2} & 0 \\ 0 & \frac{2}{\sigma^2} \end{bmatrix} \succ 0. \tag{2}$$

The infinitesimal element length is:

$$\begin{aligned} \mathrm{d}s^2 &= (\nabla\theta)^\top I(\theta)\nabla\theta, \\ &= \frac{\mathrm{d}\mu^2}{\sigma^2} + \frac{2\mathrm{d}\sigma^2}{\sigma^2}. \end{aligned}$$

After the minimization of the path length integral, the Rao distance between two normal distributions [4,31] $\theta_1 = (\mu_1, \sigma_1)$ and $\theta_2 = (\mu_2, \sigma_2)$ is given by:

$$D(\theta_1, \theta_2) = \begin{cases} \sqrt{2}\log\frac{\sigma_2}{\sigma_1} & \text{if } \mu_1 = \mu_2, \\ \frac{|\mu_1 - \mu_2|}{\sigma} & \text{if } \sigma_1 = \sigma_2 = \sigma, \\ \sqrt{2}\log\frac{\tan\frac{a_1}{2}}{\tan\frac{a_2}{2}} & \text{otherwise.} \end{cases}$$

where $a_1 = \arcsin\frac{\sigma_1}{b_{12}}$, $a_2 = \arcsin\frac{\sigma_2}{b_{12}}$ and

$$b_{12} = \sigma_1^2 + \frac{(\mu_1 - \mu_2)^2 - 2(\sigma_2^2 - \sigma_1^2)}{8(\mu_1 - \mu_2)^2}.$$

For univariate normal distributions, Rao's distance amounts to computing the hyperbolic distance for $\mathbb{H}(\frac{1}{\sqrt{2}})$, see [32].

The table below summarizes some types of Rao geometries:

Riemannian geometry	Fisher-Rao statistical manifold
Euclidean	Normal distributions with same covariance matrices
Spherical	Discrete distributions (multinomials)
Hyperbolic	Location-scale family (i.e, univariate normal, Cauchy)

4.3 Geometric Computing on Rao Statistical Manifolds

Observe that in any tangent plane T_x of the Rao statistical manifold, the inner product induces a squared Mahalanobis distance:

$$D_x(p, q) = (p - q)^\top I(x)(p - q).$$

Since matrix $I(x) \succ 0$ is positive definite, we can apply Cholesky decomposition on the Fisher information matrix $I(x) = L(x)L^\top(x)$, where $L(x)$ is a lower triangular matrix with strictly positive diagonal entries. By mapping the points p to $L(p)^\top$ in the tangent space T_p, the squared Mahalanobis amounts to computing the squared Euclidean distance $D_E(p, q) = \|p - q\|^2$ in the tangent planes:

$$D_x(p,q) = (p-q)^\top I(x)(p-q) = (p-q)^\top L(x)L^\top(x)(p-q) = D_E(L^\top(x)p, L^\top(x)q).$$

It follows that after applying the "Cholesky transformation" of objects into the tangent planes, we can solve geometric problems in tangent planes as one usually does in the Euclidean geometry. Thus we can use the classic toolbox of computational geometry in tangent planes (for extrinsic computing and mapping back and forth on the manifold using the Riemannian Log/Exp).

Let us consider the Rao univariate normal manifold that is equivalent to the hyperbolic plane. Classical algorithms like the clustering k-means do not apply straightforwardly because, in hyperpolic geometry, computing a center of mass e is not available in closed-form but requires a numerical scheme. To bypass this limitation, we rather consider non-Kärcher centroids called *model centroids* that can be easily built in hyperbolic geometry [33,34]. The computational geometry toolbox is rather limited even for the hyperbolic geometry. We proved that hyperbolic Voronoi diagrams is affine in the Klein model and reported an optimal algorithm based on power diagram construction [35,36]. We alo generalized the Euclidean minimum enclosing ball approximation algorithm using an iterative geodesic cut algorithm in [13]. This is useful for zero-centered multivariate normal distributions that has negative curvature and is guaranteed to converge.

In general, the algorithmic toolbox on generic Riemannian manifolds is very restricted due to the lack of closed-form expressions for the geodesics. One of the techniques consists in using the Riemannian Log/Exp mapping to go from/to the manifold to the tangent planes. See [37] for a review with applications on computational anatomy.

The next section explains the dual affine geometry induced by a convex function (with explicit dual geodesic parameterizations) and shows how to design efficient algorithms when consider the exponential family manifolds.

5 Amari-Chentsov Statistical Manifolds

5.1 Construction of Dually Flat Statistical Manifolds

The Legendre-Fenchel convex duality is at the core of information geometry: Any strictly convex and differentiable function F admits a dual convex conjugate F^* such that:

$$F^*(\eta) = \max_{\theta \in \Theta} \theta^\top \eta - F(\theta).$$

The maximum is attained for $\eta = \nabla F(\theta)$ and is unique since $F(\theta)$ is strictly convex ($\nabla^2 F(\theta) \succ 0$). It follows that $\theta = \nabla F^{-1}(\eta)$, where ∇F^{-1} denotes the functional inverse gradient. This implies that:

$$F^*(\eta) = \eta^\top (\nabla F)^{-1}(\eta) - F((\nabla F)^{-1}(\eta)).$$

The Legendre transformation is also called slope transformation since it maps $\theta \to \eta = \nabla F(\theta)$, where $\nabla F(\theta)$ is the gradient at θ, visualized as the slope of the support tangent plane of F at θ. The transformation is an involution for strictly convex and differentiable functions: $(F^*)^* = F$. It follows that gradient of convex conjugates are reciprocal to each other: $\nabla F^* = (\nabla F)^{-1}$. Legendre duality induces dual coordinate systems:

$$\eta = \nabla F(\theta),$$
$$\theta = \nabla F^*(\eta).$$

Furthermore, those dual coordinate systems are orthogonal to each other since,

$$\nabla^2 F(\theta) \nabla^2 F^*(\eta) = \mathrm{Id},$$

the identity matrix.

The Bregman divergence can also be rewritten in a canonical mixed coordinate form C_F or in the θ- or η-coordinate systems as

$$\begin{aligned} B_F(\theta_2 : \theta_1) &= F(\theta_2) + F^*(\eta_1) - \theta_2^\top \eta_1 = C_F(\theta_2, \eta_1) = C_{F^*}(\eta_1, \theta_2), \\ &= B_{F^*}(\eta_1 : \eta_2). \end{aligned}$$

Another use of the Legendre duality is to interpret the log-density of an exponential family as a dual Bregman divergence [38]:

$$\log p_{F,t,k,\theta}(x) = -B_{F^*}(t(x) : \eta) + F^*(t(x)) + k(x),$$

with $\eta = \nabla F(\theta)$ and $\theta = \nabla F^*(\eta)$.

5.2 Dual Geodesics: Exponential and Mixture Geodesics

Information geometry as further pioneered by Amari [23] considers dual affine geometries introduced by a pair of connections: the α-connection and $-\alpha$-connection instead of taking the Levi-Civita connection induced by the Fisher information Riemmanian metric of Rao. The ± 1-connections give rise to dually flat spaces [23] equipped with the Kullback-Leibler divergence [17]. The case of $\alpha = -1$ denotes the mixture family, and the exponential family is obtained for $\alpha = 1$. We omit technical details in this expository paper, but refer the reader to the monograph [23] for details.

For our purpose, let us say that the geodesics are defined not anymore as shortest path lengths (like in the metric case of the Fisher-Rao geometry) but rather as curves that ensures the parallel transport of vectors [23]. This defines the notion of "straightness" of lines. Riemannian geodesics satisfy both the straightness property and the minimum length requirements. Introducing dual connections, we do not have anymore distances interpreted as curve lengths, but the geodesics defined by the notion of straightness only.

In information geometry, we have dual geodesics that are expressed for the exponential family (induced by a convex function F) in the dual affine coordinate systems θ/η for $\alpha = \pm 1$ as:

$$\gamma_{12} : L(\theta_1, \theta_2) = \{\theta = (1-\lambda)\theta_1 + \lambda\theta_2 \mid \lambda \in [0,1]\},$$
$$\gamma_{12}^* : L^*(\eta_1, \eta_2) = \{\eta = (1-\lambda)\eta_1 + \lambda\eta_2 \mid \lambda \in [0,1]\}.$$

Furthermore, there is a *Pythagorean theorem* that allows one to define information-theoretic projections [23]. Consider three points p, q and r such that γ_{pq} is the θ-geodesic linking p to q, and γ_{qr}^* is the η-geodesic linking q to r. The geodesics are orthogonal at the intersection point q if and only if the Pythagorean relation is satisfied:

$$D(p:r) = D(p:q) + D(q:r).$$

In fact, a more general triangle relation (extending the law of cosines) exists:

$$D(p:q) + D(q:r) - D(p:r) = (\theta(p) - \theta(q))^\top (\eta(r) - \eta(q)).$$

Note that the θ-geodesic γ_{pq} and η-geodesic γ_{qr}^* are orthogonal with respect to the inner product $G(q)$ defined at q (with $G(q) = I(q)$ being the Fisher information matrix at q). Two vectors u and v in the tangent place T_q at q are said to be orthogonal if and only if their inner product equals zero:

$$u \perp_q v \Leftrightarrow u^\top I(q) v = 0.$$

Information geometry of dually flat spaces thus extend the traditional self-dual Euclidean geometry, obtained for the convex function $F(x) = \frac{1}{2}x^\top x$ (and corresponding to the statistical manifold of isotropic Gaussians).

The construction can be extended to dual constant curvature manifolds using Amari-Chentsov's affine α-connections. We omit those details here, but refer the reader to the textbook [23].

5.3 Learning Statistical Patterns

We mentioned in the introduction that statistical patterns can either be learned from (1) a parametric model, (2) a mixture model, or (3) a kernel density estimator. We concisely review algorithms to learn those statistical patterns by taking into consideration the *exponential family manifold* (EFM).

Parametric Distribution. Let $x_1, ..., x_n$ be n data points assumed to be iid. from an exponential family. The maximum likelihood estimator (MLE) yields [20]:

$$\eta(\hat{P}) = \frac{1}{n} t(x_i) = \bar{t}$$

The point $\hat{P}$ on the EFM with η-coordinates $\bar{t}$ is called the *observed point* in information geometry [23]. The MLE is guaranteed to exist [39,40] provided that matrix:

$$T = \begin{bmatrix} 1 \; t_1(x_1) \; ... \; t_D(x_1) \\ \vdots \; \vdots \qquad \vdots \; \vdots \\ 1 \; t_1(x_n) \; ... \; t_D(x_n) \end{bmatrix} \tag{3}$$

of dimension $n \times (D + 1)$ has rank $D + 1$ [40].

Furthermore, the log-likelihood achieved by the MLE can be expressed as:

$$l(\hat{\theta}; x_1, ..., x_n) = F^*(\hat{\eta}) + \frac{1}{n} \sum_{i=1}^{n} k(x_i)$$

For exponential families, the MLE is consistent and efficient (i.e., matches the Cramér-Rao lower bound) and has normal asymptotic distribution with covariance matrix the inverse of the Fisher information matrix:

$$\sqrt{n}(\hat{\theta} - \theta) \overset{\text{distribution}}{\longrightarrow} N(0, I^{-1}(\theta)).$$

Notice that to choose between two different exponential family models, say, parameterized by F_1 and F_2, we can evaluate their MLE log-likelihood using their respective convex conjugates F_1^* and F_2^*, and choose the model which yielded the highest likelihood.

Learning Finite Mixture Distributions. By using the duality between (regular) exponential families and (regular) Bregman divergences, Banerjee *et al.* [38] showed that the classical EM algorithm for learning mixtures of the same exponential families amount to a *soft Bregman clustering*. The EM maximizes the expected complete log-likelihood [7]. Recently, it has been shown that maximizing the complete log-likelihood (by labeling all observation data with their component number) for an exponential family mixture amounts to perform a k-means clustering for the dual Bregman divergence B_{F^*} on the sufficient statistic data: $\{y_i = t(x_i)\}_{i=1}^n$. Thus by using Lloyd batched k-means algorithm that optimizes the k-means loss, we obtain an algorithm for learning mixtures. This algorithm is called k-MLE [41] and outperforms computationally EM since it deals with hard membership. Furthermore, a generalization of k-MLE considers for each component a different exponential family and adds a step to choose the best exponential family of a cluster. This generalized k-MLE has been described specifically for learning generalized gaussian mixtures [42], gamma mixtures [43], and Wishart mixtures [44]. (The technical details focus on computing the dual convex conjugate F^* and on how to stratify an exponential family with $D > 1$ parameters as a family of exponential families of order $D - 1$.)

Learning Non-parametric Distributions with KDEs. For each datum x_i, we can associate a density with weight $\frac{1}{n}$ and mode matching x_i. This is the kernel density estimator [7] (KDE). For the kernel family, we can choose the univariate location-scale families or multivariate elliptical distributions. Normal

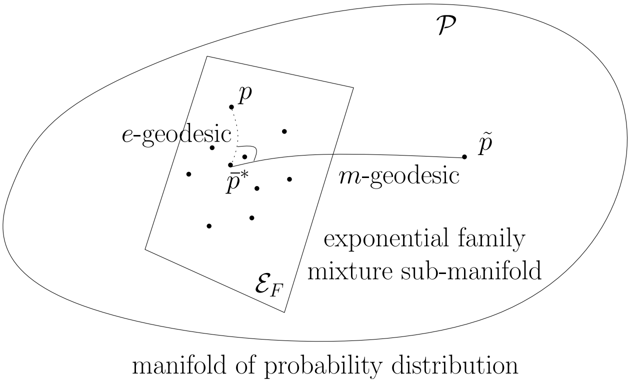


Fig. 1. Simplifying a statistical mixture of exponential families or KDE $\tilde{p}$ to a single component model amounts to perform a Kullback-Leibler projection of the mixture onto the exponential family manifold [45]. Optimality is proved using the Pythagorean theorem of dually flat geometries.

distributions belong both to the exponential families and the elliptical families. Since the mixture model is dense and has n components, we can simplify this representation to a sparse model by performing mixture simplification.

Simplifying KDEs and Mixtures. A statistical mixture or a KDE is represented on the exponential family manifold as a *weighted point set*. We simplify a mixture by clustering. This requires to compute centroids and barycenters with respect to information-theoretic distances. The Kullback-Leibler and Jeffreys centroid computations have been investigated in [46].

A neat geometric characterization of the mixture simplification is depicted in Figure 1. We project the mixture $\tilde{p}$ on the exponential family manifold using the m-geodesic. This amounts to compute a barycenter of the weighted parameter points on the manifold. See [45] for further details.

Instead of clustering groupwise, we can also consider hierarchical clustering to get a dendrogram [7] (a binary tree-structured representation): This yields a mixture representation with *levels of details* for modeling statistical mixtures [47]. We can extend the centroid computations to the wider class of skewed Bhattacharrya centroids [22] that encompasses the Kullback-Leibler divergence. In [48,49], we further consider the novel class of information-theoretic divergences called *total Bregman divergences*. The total Bregman divergence (and total Kullback-Leibler divergence when dealing with exponential family members) is defined by:

$$tB(P:Q) = \frac{B(P:Q)}{\sqrt{1+\|\nabla F(\theta(Q))\|^2}},$$

and yields *conformal geometry* [49]. We experimentally improved application performance for shape retrieval and diffusion tensor imaging.

5.4 Statistical Voronoi Diagrams

It is well-known that the k-means algorithm [7] is related to ordinary Voronoi diagrams since data points are associated to their closest centroid. Namely, the centroids play the role of Voronoi seeds. The Kullback-Leibler k-means intervenes in the description of the k-MLE or the mixture simplification algorithms. For distributions belonging to the same exponential families, those statistical Voronoi diagrams amount to perform Bregman Voronoi diagrams on the distribution parameters (using either the natural θ-coordinates, or the dual η-coordinates). The Bregman Voronoi diagrams and its extensions have been investigated in [50,51,52,53]. They can always be reduced to *affine diagrams* (i.e., hyperplane bisectors) which can be computed either as equivalent *power diagrams* or by generalizing the Euclidean paraboloid lifting procedure by choosing the potential function $(x, F(x))$ instead of the paraboloid [50]. Statistical Voronoi diagrams can also be used for *multiple* class hypothesis testing: Figure 2 illustrates a geometric characterization of the Chernoff distance of a set of n distributions belonging to the same exponential families. Refer to [54] for further explanations.

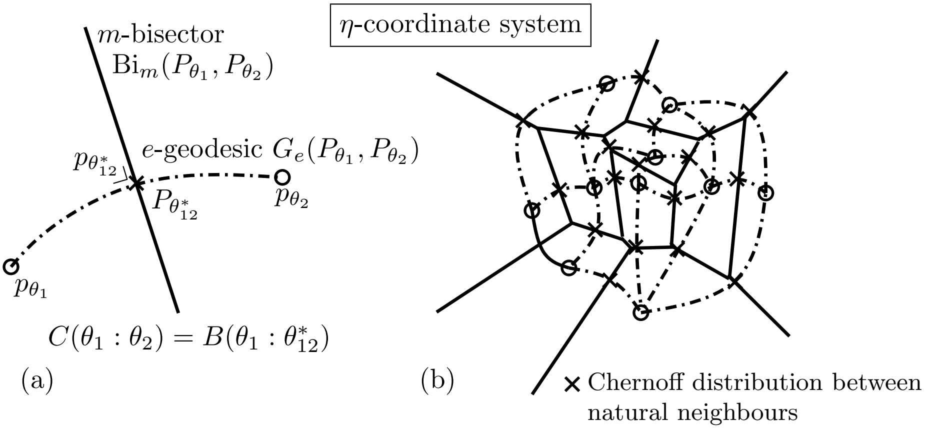

η-coordinate system
m-bisector
$\mathrm{Bi}_m(P_{\theta_1}, P_{\theta_2})$
e-geodesic $G_e(P_{\theta_1}, P_{\theta_2})$
$p_{\theta^*_{12}}$
$P_{\theta^*_{12}}$
p_{θ_2}
p_{θ_1}
$C(\theta_1 : \theta_2) = B(\theta_1 : \theta^*_{12})$
(a)
(b)
× Chernoff distribution between natural neighbours

Fig. 2. Geometry of the best error exponent in Bayesian classification [54]. Binary hypothesis (a): The Chernoff distance is equal to the Kullback-Leibler divergence from the midpoint distribution $P_{\theta^*_{12}}$ to the extremities, where the midpoint distribution $P_{\theta^*_{12}}$ (×) is obtained as the left-sided KL projection of the sites to their bisector [55]. (b) Multiple hypothesis testing: The Chernoff distance is the minimum of pairwise Chernoff distance that can be deduced from statistical Voronoi diagram by inspecting all Chernoff distributions (×) lying on $(d-1)$-faces. Both drawings illustrated in the η-coordinate system where m-bisectors are hyperplanes.

6 Conclusion and Perspectives

We concisely reviewed the principles of computational information geometry for pattern learning and recognition on statistical manifolds: We consider statistical patterns whose distributions are either represented by atomic distributions (parametric models, say, of an exponential family), mixtures thereof (semi-parametric models), or kernel density estimations (non-parametric models). Those statistical pattern representations need to be estimated from datasets. We presented a geometric framework to learn and process those statistical patterns by embedding them on statistical manifolds. A statistical pattern is then represented either by a single point (parametric model), a k-weighted point set or a n-point set on the statistical manifold. To discriminate between patterns, we introduced the notion of statistical distances, and presented a genesis that yielded the family of α-divergences. We described the two notions of statistical invariances on statistical manifolds: invariance by sufficient statistic and invariance by 1-to-1 reparameterization of distribution parameters. We then introduced two kinds of statistical manifolds that fulfills the statistical invariance: The Rao manifolds based on Riemannian geometry using the Fisher information matrix as the underlying metric tensor, and the Amari-Chentsov dually flat manifolds based on the convex duality induced by a convex functional generator. We then explained why the usual lack of closed-form geodesic expression for Rao manifolds yields a limited algorithmic toolbox. By contrast, the explicit dual geodesics of Amari-Chentsov manifolds provides a handy framework to extend the Euclidean algorithmic toolbox. We illustrated those concepts by reviewing the Voronoi diagrams (and dual Delaunay triangulations), and considered simplifying mixtures or KDEs using clustering techniques. In particular, in the Amari-Chentsov manifolds, we can compute using either the primal, dual, or mixed coordinate systems. This offers many strategies for efficient computing. For the exponential family manifolds, we explained the bijection between exponential families, dual Bregman divergences and quasi-arithmetic means [10].

We would like to conclude with perspectives for further work. To begin with, let us say that there are several advantages to think "geometrically":

- First, it allows to use simple concepts like line segments, balls, projections to describe properties or algorithms. The language of geometry gives special affordances for human thinking. For example, to simplify a mixture of exponential families to a single component amount to project the mixture model onto the exponential family manifold (depicted in Figure 1). Algorithmically, this projection is performed by computing a barycenter.
- Second, sometimes we do not have analytical solution but nevertheless we can still describe geometrically exactly where the solution is. For example, consider the Chernoff information of two distributions: It is computed as the Kullback-Leibler divergence from the mid-distribution to the extremities (depicted in Figure 2). The mid-distribution is the unique distribution that is at the intersection of the exponential geodesic with the mixture bisector.

We implemented those various algorithms in the jMEF[4] [56] or PyMEF[5] [57] software libraries.

To quote mathematician Jules H. Poincaré: "One geometry cannot be more true than another; it can only be more convenient". We have exemplified this quote by showing that geometry is not absolute nor ultimate: Indeed, we have shown two kinds of geometries for handling statistical manifolds: Rao Riemannian manifolds and Amari-Chentsov dual affine manifolds. We also presented several *mathematical tricks* that yielded computational convenience: Bounding the intersection similarity measure with quasi-arithmetic means extends the α-divergences. Besides the Rao and Amari-Chentsov manifolds, we can also consider Finsler geometry [58] or Hilbert spherical geometry in infinite dimensional spaces to perform statistical pattern recognition. Non-extensive entropy pioneered by Tsallis also gave birth to *deformed exponential families* that have been studied using conformal geometry. See also the infinite-dimensional exponential families and Orlicz spaces [59], the optimal transport geometry [60], the symplectic geometry, Kähler manifolds and Siegel domains [61], the Geometry of proper scoring rules [62], the quantum information geometry [63], etc, etc. This raises the question of knowing which geometry to choose? For a specific application, we can study and compare experimentally say Rao vs. Amari-Chentsov manifolds. However, we need deeper axiomatic understandings in future work to (partially) answer this question. For now, we may use Rao manifolds if we require metric properties of the underlying distance, or if we want to use the triangular inequality to improve k-means clustering or nearest-neighbor searches. Some applications require to consider symmetric divergences: We proposed a parametric family of symmetric divergences [64] including both the Jeffreys divergence and the Jensen-Shannon divergence, and described the centroid computations with respect to that class of distances.

Geometry offers many more possibilities to explore in the era of big data analytics as we are blinded with numbers and need to find rather qualitative invariance of the underlying space of data. There are many types of geometries to explore or invent as mothers of models. Last but not least, we should keep in mind statistician George E. P. Box quote: "Essentially, all models are wrong, but some are useful." When it comes to data spaces, we also believe that all geometries are wrong, but some are useful.

References

1. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: A review. IEEE Trans. Pattern Anal. Mach. Intell. 22, 4–37
2. Cramér, H.: Mathematical Methods of Statistics. Princeton Landmarks in mathematics (1946)
3. Fréchet, M.: Sur l'extension de certaines évaluations statistiques au cas de petits échantillons. Review of the International Statistical Institute 11, 182–205 (1939) (published in IHP Lecture)

[4] http://www.lix.polytechnique.fr/~nielsen/MEF/

[5] http://www.lix.polytechnique.fr/~schwander/pyMEF/

4. Rao, C.R.: Information and the accuracy attainable in the estimation of statistical parameters. Bulletin of the Calcutta Mathematical Society 37, 81–89
5. Nielsen, F.: In : Connected at Infinity II: A selection of mathematics by Indians. Cramér-Rao lower bound and information geometry (Hindustan Book Agency (Texts and Readings in Mathematics, TRIM)) arxiv 1301.3578
6. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B (Methodological) 39, 1–38
7. Fukunaga, K.: Introduction to statistical pattern recognition, 2nd edn. Academic Press Professional, Inc. (1990); (1st edn. 1972)
8. Piro, P., Nielsen, F., Barlaud, M.: Tailored Bregman ball trees for effective nearest neighbors. In: European Workshop on Computational Geometry (EuroCG), LORIA, Nancy, France. IEEE (2009)
9. Nielsen, F., Piro, P., Barlaud, M.: Bregman vantage point trees for efficient nearest neighbor queries. In: Proceedings of the 2009 IEEE International Conference on Multimedia and Expo (ICME), pp. 878–881 (2009)
10. Nock, R., Nielsen, F.: Fitting the smallest enclosing bregman balls. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 649–656. Springer, Heidelberg (2005)
11. Nielsen, F., Nock, R.: On the smallest enclosing information disk. Inf. Process. Lett. 105, 93–97
12. Nielsen, F., Nock, R.: On approximating the smallest enclosing Bregman balls. In: ACM Symposium on Computational Geometry (SoCG). ACM Press (2006)
13. Arnaudon, M., Nielsen, F.: On approximating the Riemannian 1-center. Computational Geometry 46, 93–104
14. Nielsen, F., Nock, R.: Approximating smallest enclosing balls with applications to machine learning. Int. J. Comput. Geometry Appl. 19, 389–414
15. Ali, S.M., Silvey, S.D.: A general class of coefficients of divergence of one distribution from another. Journal of the Royal Statistical Society, Series B 28, 131–142
16. Csiszár, I.: Information-type measures of difference of probability distributions and indirect observation. Studia Scientiarum Mathematicarum Hungarica 2, 229–318
17. Cover, T.M., Thomas, J.A.: Elements of information theory. Wiley Interscience, New York (1991)
18. Nielsen, F.: Closed-form information-theoretic divergences for statistical mixtures. In: International Conference on Pattern Recognition, ICPR (2012)
19. Wu, J., Rehg, J.M.: Beyond the Euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In: ICCV (2009)
20. Nielsen, F., Garcia, V.: Statistical exponential families: A digest with flash cards. arXiv.org:0911.4863 (2009)
21. Hellman, M.E., Raviv, J.: Probability of error, equivocation and the Chernoff bound. IEEE Transactions on Information Theory 16, 368–372
22. Nielsen, F., Boltz, S.: The Burbea-Rao and Bhattacharyya centroids. IEEE Transactions on Information Theory 57, 5455–5466
23. Amari, S., Nagaoka, H.: Methods of Information Geometry. Oxford University Press (2000)
24. Qiao, Y., Minematsu, N.: A study on invariance of f-divergence and its application to speech recognition. Transactions on Signal Processing 58, 3884–3890
25. Pardo, M.C., Vajda, I.: About distances of discrete distributions satisfying the data processing theorem of information theory. IEEE Transactions on Information Theory 43, 1288–1293

26. Amari, S.: Alpha-divergence is unique, belonging to both f-divergence and Bregman divergence classes. IEEE Transactions on Information Theory 55, 4925–4931
27. Morozova, E.A., Chentsov, N.N.: Markov invariant geometry on manifolds of states. Journal of Mathematical Sciences 56, 2648–2669
28. Fisher, R.A.: On the mathematical foundations of theoretical statistics. Philosophical Transactions of the Royal Society of London A 222, 309–368
29. Chentsov, N.N.: Statistical Decision Rules and Optimal Inferences. Transactions of Mathematics Monograph, numero 53 (1982) (published in Russian in 1972)
30. Peter, A., Rangarajan, A.: A new closed-form information metric for shape analysis, vol. 1, pp. 249–256
31. Atkinson, C., Mitchell, A.F.S.: Rao's distance measure. Sankhya A 43, 345–365
32. Lovric, M., Min-Oo, M., Ruh, E.A.: Multivariate normal distributions parametrized as a Riemannian symmetric space. Journal of Multivariate Analysis 74, 36–48
33. Schwander, O., Nielsen, F.: Model centroids for the simplification of kernel density estimators. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 737–740
34. Arnaudon, M., Nielsen, F.: Medians and means in Finsler geometry. CoRR abs/1011.6076 (2010)
35. Nielsen, F., Nock, R.: Hyperbolic Voronoi diagrams made easy, vol. 1, pp. 74–80. IEEE Computer Society, Los Alamitos
36. Nielsen, F., Nock, R.: The hyperbolic voronoi diagram in arbitrary dimension. CoRR abs/1210.8234 (2012)
37. Pennec, X.: Statistical computing on manifolds: From riemannian geometry to computational anatomy. In: Nielsen, F. (ed.) ETVC 2008. LNCS, vol. 5416, pp. 347–386. Springer, Heidelberg (2009)
38. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with Bregman divergences. Journal of Machine Learning Research 6, 1705–1749
39. Barndorff-Nielsen, O.E.: Information and exponential families: In statistical theory. Wiley series in probability and mathematical statistics: Tracts on probability and statistics. Wiley (1978)
40. Bogdan, K., Bogdan, M.: On existence of maximum likelihood estimators in exponential families. Statistics 34, 137–149
41. Nielsen, F.: k-MLE: A fast algorithm for learning statistical mixture models. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). IEEE (2012) (preliminary, technical report on arXiv)
42. Schwander, O., Nielsen, F., Schutz, A., Berthoumieu, Y.: k-MLE for mixtures of generalized Gaussians. In: International Conference on Pattern Recognition, ICPR (2012)
43. Schwander, O., Nielsen, F.: Fast learning of Gamma mixture models with k-MLE. In: Hancock, E., Pelillo, M. (eds.) SIMBAD 2013. LNCS, vol. 7953, pp. 235–249. Springer, Heidelberg (2013)
44. Saint-Jean, C., Nielsen, F.: A new implementation of k-MLE for mixture modelling of Wishart distributions. In: Geometric Sciences of Information, GSI (2013)
45. Schwander, O., Nielsen, F.: Learning Mixtures by Simplifying Kernel Density Estimators. In: Bhatia, Nielsen (eds.) Matrix Information Geometry, pp. 403–426
46. Nielsen, F., Nock, R.: Sided and symmetrized Bregman centroids. IEEE Transactions on Information Theory 55, 2882–2904
47. Garcia, V., Nielsen, F., Nock, R.: Levels of details for Gaussian mixture models, vol. 2, pp. 514–525

48. Vemuri, B., Liu, M., Amari, S., Nielsen, F.: Total Bregman divergence and its applications to DTI analysis. IEEE Transactions on Medical Imaging (2011) 10.1109/TMI.2010.2086464
49. Liu, M., Vemuri, B.C., Amari, S., Nielsen, F.: Shape retrieval using hierarchical total Bregman soft clustering. Transactions on Pattern Analysis and Machine Intelligence (2012)
50. Boissonnat, J.-D., Nielsen, F., Nock, R.: Bregman Voronoi diagrams. Discrete Comput. Geom. 44, 281–307
51. Nielsen, F., Boissonnat, J.-D., Nock, R.: On Bregman Voronoi diagrams. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, pp. 746–755. Society for Industrial and Applied Mathematics, Philadelphia
52. Nielsen, F., Boissonnat, J.-D., Nock, R.: Visualizing Bregman Voronoi diagrams. In: Proceedings of the Twenty-Third Annual Symposium on Computational Geometry, SCG 2007, pp. 121–122. ACM, New York
53. Nielsen, F., Nock, R.: Jensen-Bregman Voronoi diagrams and centroidal tessellations. In: International Symposium on Voronoi Diagrams (ISVD), pp. 56–65.
54. Nielsen, F.: Hypothesis testing, information divergence and computational geometry. In: Geometric Sciences of Information, GSI (2013)
55. Nielsen, F.: An information-geometric characterization of Chernoff information. IEEE Signal Processing Letters (SPL) 20, 269–272
56. Garcia, V., Nielsen, F.: Simplification and hierarchical representations of mixtures of exponential families. Signal Processing (Elsevier) 90, 3197–3212
57. Schwander, O., Nielsen, F.: PyMEF - A framework for exponential families in Python. In: IEEE/SP Workshop on Statistical Signal Processing, SSP (2011)
58. Shen, Z.: Riemann-Finsler geometry with applications to information geometry. Chinese Annals of Mathematics 27B, 73–94
59. Cena, A., Pistone, G.: Exponential statistical manifold. Annals of the Institute of Statistical Mathematics 59, 27–56
60. Gangbo, W., McCann, R.J.: The geometry of optimal transportation. Acta Math. 177, 113–161
61. Barbaresco, F.: Interactions between Symmetric Cone and Information Geometries: Bruhat-Tits and Siegel Spaces Models for High Resolution Autoregressive Doppler Imagery. In: Nielsen, F. (ed.) ETVC 2008. LNCS, vol. 5416, pp. 124–163. Springer, Heidelberg (2009)
62. Dawid, A.P.: The geometry of proper scoring rules. Annals of the Institute of Statistical Mathematics 59, 77–93
63. Grasselli, M.R., Streater, R.F.: On the uniqueness of the Chentsov metric in quantum information geometry. Infinite Dimensional Analysis, Quantum Probability and Related Topics 4, 173–181, arXiv.org:math-ph/0006030
64. Nielsen, F.: A family of statistical symmetric divergences based on Jensen's inequality. CoRR abs/1009.4004 (2010)

Dimension Reduction Methods for Image Pattern Recognition

Hayato Itoh[1], Tomoya Sakai[2], Kazuhiko Kawamoto[3], and Atsushi Imiya[4]

[1] School of Advanced Integration Science, Chiba University,
1-33 Yayoicho, Inage-ku, Chiba, 263-8522, Japan
[2] Graduate School of Engineering, Nagasaki University,
1-14 Bunkyo-cho, Nagasaki, 852-8521, Japan
[3] Academic Link Center, Chiba University,
1-33 Yayoicho, Inage-ku, Chiba, 263-8522, Japan
[4] Institute of Media and Information Technology, Chiba University,
1-33 Yayoicho, Inage-ku, Chiba, 263-8522, Japan

Abstract. In this paper, we experimentally evaluate the validity of dimension-reduction methods for the computation of the similarity in pattern recognition. Image pattern recognition uses pattern recognition techniques for the classification of image data. For the numerical achievement of image pattern recognition techniques, images are sampled using an array of pixels. This sampling procedure derives vectors in a higher-dimensional metric space from image patterns. For the accurate achievement of pattern recognition techniques, the dimension reduction of data vectors is an essential methodology, since the time and space complexities of data processing depend on the dimension of data. However, dimension reduction causes information loss of geometrical and topological features of image patterns. The desired dimension-reduction method selects an appropriate low-dimensional subspace that preserves the information used for classification.

1 Introduction

Pattern recognition techniques are applied to various areas such as face recognition [1], character recognition[2], spatial object recognition[3], fingerprint classification [4] and iris recognition[5]. These applications deal with image patterns. In image pattern recognition, images are sampled so that they can be embeded in a vector space. Kernel methods are promised to analyse a relational data with more complex structure[6,7]. For practical computation, we embedding the image in a vector space too. Furthermore, dimension reduction is operated to reduce the dimensions of image patterns.

In practice, as shown in Fig. 1, two methods are used for dimension reduction. One method reduces the dimension of data in a sampled image space using image compression methods such as the pyramid transform, wavelet transform and low-pass filtering. The other method is data compression in a vector space after vectorisation of sampled image patterns using operations such as random

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 26–42, 2013.

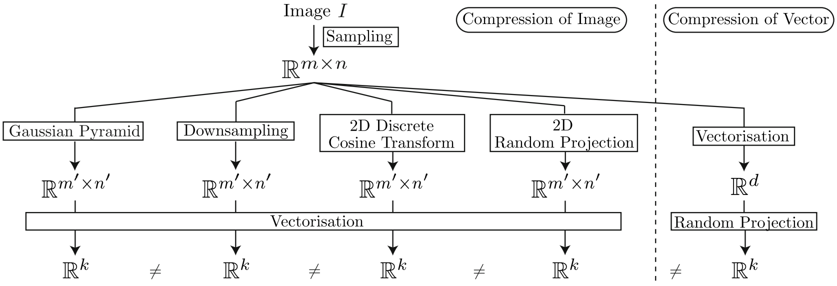


Image I
Sampling
$\mathbb{R}^{m \times n}$
Compression of Image
Compression of Vector
Gaussian Pyramid
Downsampling
2D Discrete Cosine Transform
2D Random Projection
Vectorisation
$\mathbb{R}^{m' \times n'}$
$\mathbb{R}^{m' \times n'}$
$\mathbb{R}^{m' \times n'}$
$\mathbb{R}^{m' \times n'}$
$\mathbb{R}^{d}$
Vectorisation
Random Projection
$\mathbb{R}^{k} \neq \mathbb{R}^{k} \neq \mathbb{R}^{k} \neq \mathbb{R}^{k} \neq \mathbb{R}^{k}$

Fig. 1. Differences in the dimension-reduction path among downsampling, the Gaussian pyramid transform, two-dimensional discrete transformation, the two-dimensional random projection and random projection. After the sampling of an original image, dimension-reduction methods mainly follow two paths. In the first path, after the reduction of the image, the reduced image will be converted to a vector. In the second path, after vectorisation, the feature vector is be reduced. Here, $m, m', n, n', d, k \in \mathbb{Z}$ and $n' < n, m' < m, k < d$.

projection. The reduction and vectorisation operations are generally noncommutative as shown in Fig. 1. The pyramid transform is a nonexpansion mapping. As shown in this paper, a nonexpansion mapping affects the similarity, while a random projection is a stochastically unitary operation which preserves the metric between original image patterns and compressed image patterns.

In this paper, we evaluate the effects and performance of these two properties of data compression. We adopted the following dimension-reduction techniques: downsampling of the pixels, the Gaussian-based pyramid transform, the two-dimensional discrete cosine transform and random projection. For classification, we adopted the subspace method, mutual subspace method, constraint mutual subspace method and two-dimensional tensorial subspace method. We tested each pair of these dimension-reduction techniques and classifiers for face recognition, spatial object recognition and character recognition.

2 Related Works

The local preserving projection (LPP) was introduced as a linear approximation of the Laplacian eigenmap of a nonflat discrete data manifold[8]. The method locally preserves the distance relation among data.

Principal component analysis (PCA) was introduced for the linear approximation of a subspace of Chinease characters and spatial data[9,10]. PCA selects the subspace in which the covariance of class data is maximised. To improve the accuracy of the eigenspace computed using learning data, Leonardis *et al.* dealt with a locally low-dimensional structure for appearance-based image matching[3]. The constant normalisation in PCA[9] subtracts the constant bias, since each image pattern contains a constant bias. This process is a nonexpansion mapping.

The classical subspace method computes the orthogonal projection of inputs to each category. As an extension, the mutual subspace method[11,12] computes the orthogonal projection of the subspaces spanned by inputs with perturbations. A combination of a generalisation of the constant normalisation and the mutual subspace method is proposed in ref. [12]. The method subtracts the elements in the common linear subspace of many categories.

Two-dimensional PCA (2DPCA)[1,13] is a special case of tensor PCA[14], since 2DPCA deals with images, which are linear two-dimensional arrays, as a tensor of order two. 2DPCA considers only the row distribution of images[13] although there is a method which considers both the column and row distributions of images[15].

Linear discriminant analysis (LDA) and the related Fisher's linear discriminant are methods which reduce the dimension of data by maximising the ratio between the inter- and intraclass distances[16,4].

As a nonlinear pattern recogntion method, the kernel method is a promising techniques[6,7]. The kernel method is extended from metric data to combinatrial data, such as graph structural data. This extension provides a powerful method to data mining for biochemistry. The graph kernel is an discrete version of diffusion based data, which produce the combinatrial structure.

These methods are not able to deal with images with too high resolution. Therefore, we need a dimension-reduction methods for preprocessing.

3 Dimension Reduction Methods

3.1 Gaussian-Based Pyramid Transform

We define the image reduction method as

$$g(x,y) = Rf(x,y) = \int\int_{\mathbf{R}^2} w_1(u)w_1(v)f(2x-u, 2y-v)dudv, \qquad (1)$$

$$w_1(x) = \begin{cases} \frac{1}{2}(1-\frac{|x|}{2}), & |x| \leq 2 \\ 0, & |x| > 2 \end{cases}. \qquad (2)$$

The dual operation of R is

$$Eg(x,y) = 4\int\int_{\mathbf{R}^2} w_1(u)w_1(v)g(\frac{x-u}{2}, \frac{y-v}{2})dudv. \qquad (3)$$

For $g = Rf$, the derivative of g satisfies the relations

$$g_x = \frac{1}{2}Rf_x, \quad g_y = \frac{1}{2}Rf_y. \qquad (4)$$

Therefore, we have the following relation.

Theorem 1. *For $g = Rf$, we have the relation*

$$\begin{pmatrix} g_{xx}, g_{xy} \\ g_{yx}, g_{yy} \end{pmatrix} = \frac{1}{2^2}\begin{pmatrix} Rf_{xx}, Rf_{xy} \\ Rf_{yx}, Rf_{yy} \end{pmatrix}.$$

Furthermore, the pyramid transform globally preserves the geometry of the terrain $z = f(x, y)$ same as the case of the Gaussian scale space transform [1].

For L_1 and L_2, the norm is defined as

$$\|f\|_2 = \left(\int \int_{\mathrm{R}^2} |f(x, y)|^2 \, dxdy \right)^{\frac{1}{2}}, \tag{5}$$

$$\|g\|_1 = \int \int_{\mathrm{R}^2} |g(x, y)| \, dxdy. \tag{6}$$

For the convolution of $f(x, y)$ and $g(x, y)$ such as $h(x, y) = g(x, y) * f(x, y)$, we have the following proposition.

Proposition 1. *For the energy of a convolution, we have the following property:*

$$\|h(x, y)\|_2 = \|g(x, y) * f(x, y)\|_2 \leq \|g\|_1 \|f\|_2. \tag{7}$$

For the linear operator R, if $Rf = 0$, we have the following theorem.

Theorem 2. *For both for $f \in L_2$ and $g \in L_2$, the relation*

$$\|Rf - Rg\|_2 \leq \|f - g\|_2 \tag{8}$$

is satisfied.

This theorem implies that if g exists in the neighbourhood of f, that is, $\|f - g\|_2 < \epsilon, \epsilon \ll 1$, then Rg exists in the neighbourhood of Rf, and $\|Rf - Rg\|_2 < \epsilon', \epsilon' \ll \epsilon \ll 1$. Therefore, Rf preserves the local topology of the pattern space. For the nonexpansion mapping ϕ such that

$$\|\phi(f) - \phi(g)\|_2 \leq r\|f - g\|_2, \, 0 \leq r \leq 1, \tag{9}$$

with the condition $\phi(f) \leq rf$, we have the following property. Figure 2 illustrates the following theorem.

Theorem 3. *Setting $\angle(f, g)$ to be the angle between f and g in the Hilbert space H, the relation*

$$\angle(\phi(f), \phi(g)) \leq \angle(f, g) \tag{10}$$

is satisfied.

(*Proof*) From the assumptions for the norms, we have the relations $\|\phi(f)\|_2 \leq \|f\|_2$, $\|\phi(g)\|_2 \leq \|g\|_2$ and $\|\phi(f) - \phi(g)\|_2 \leq \|f - g\|_2$. Furthermore, $\phi(f) \neq \lambda f$, $\phi(g) \neq \mu g$ and $\phi(f - g) \neq \nu(f - g)$. These relations imply the relation

$$\frac{(f, g)}{\|f\|_2 \|g\|_2} \leq \frac{(\phi(f), \phi(g))}{\|\phi(f)\|_2 \|\phi(g)\|_2}. \tag{11}$$

[1] The pyramid transform preserves the local geometry and topology of an image pattern, whereas the random projection preserves the local topology of the vectors of an image pattern.

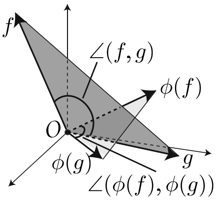


Fig. 2. Angle between two functions and the nonexpansion map. $f, g \in H$ are set to be functions and ϕ is set to be a nonexpansion map. Here, $\angle(f, g)$ represents the angle between f and g.

Setting $\theta = \angle(f, g)$, $\cos\theta = \frac{(f,g)}{\|f\|_2\|g\|_2}$. Therefore, $\angle(\phi(f), \phi(g)) = \theta_\phi \leq \theta$ (*Q.E.D.*)
From Theorem 1 and Theorem 2, image reduction by the pyramid transform reduces the angle between two images preserving their geometric properties.

For the sampled function $f_{ij} = f(i, j)$, the pyramid transform R and its dual transform E [17] are expressed as

$$Rf_{mn} = \sum_{i,j=-1}^{1} w_i w_j f_{2m-i,\,2n-j}, \quad Ef_{mn} = 4 \sum_{i,j=-2}^{2} w_i w_j f_{\frac{m-i}{2},\frac{n-j}{2}}, \tag{12}$$

where $w_{\pm 1} = \frac{1}{4}$ and $w_0 = \frac{1}{2}$. Moreover, the summation is carried out for integers $(m - i)$ and $(n - j)$. These two operations involves the reduction and expansion of the image size. As a nonexpansion mapping, the pyramid transform compresses the nth-order tensor with $\mathcal{O}(1/2^n)$, preserving the differential geometric structure of the tensor data.

3.2 Random Projection

Let R be a $k \times d$ matrix whose k row vectors span a k-dimensional linear subspace in $\mathbb{R}^d$ $(k < d)$. We obtain a low-dimensional representation $\hat{x}$ for each $\boldsymbol{x}_i \in X$ as

$$\hat{\boldsymbol{x}}_i = \sqrt{\frac{d}{k}} \boldsymbol{R}\boldsymbol{x}_i. \tag{13}$$

Figure 3(a) shows the basic idea of the random projection[18]. For the random projection, we have the following embedding property from the Johnson-Lindenstrauss lemma[19,20].

Theorem 4. *(Johnson-Lindenstrauss embeddings). For any* $0 < \epsilon$*, set* X *of* N *points* $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ *and* $k < d$*, one can map* X *to* $\hat{X} = \{\hat{\boldsymbol{x}}_1, \dots, \hat{\boldsymbol{x}}_N \in \mathbb{R}^{\hat{d}}\}$ *by the random projection in Eq. (13) with probability* $(1 - \mathrm{e}^{-\mathcal{O}(k\epsilon^2)})$ *when*

$$(1 - \epsilon)\|\boldsymbol{x}_j - \boldsymbol{x}_i\|_2 \leq \|\hat{\boldsymbol{x}}_j - \hat{\boldsymbol{x}}_i\|_2 \leq (1 + \epsilon)\|\boldsymbol{x}_j - \boldsymbol{x}_i\|_2. \tag{14}$$

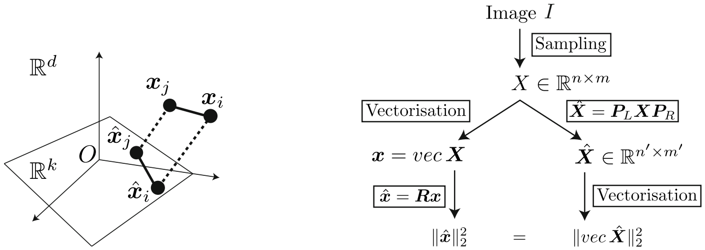

$\mathbb{R}^d$
$\boldsymbol{x}_j$
$\boldsymbol{x}_i$
$\hat{\boldsymbol{x}}_j$
O
$\mathbb{R}^k$
$\hat{\boldsymbol{x}}_i$
Image I
Sampling
$X \in \mathbb{R}^{n\times m}$
Vectorisation
$\hat{\boldsymbol{X}} = \boldsymbol{P}_L\boldsymbol{X}\boldsymbol{P}_R$
$\boldsymbol{x} = vec\,\boldsymbol{X}$
$\hat{\boldsymbol{X}} \in \mathbb{R}^{n'\times m'}$
$\hat{\boldsymbol{x}} = \boldsymbol{R}\boldsymbol{x}$
Vectorisation
$\|\hat{\boldsymbol{x}}\|_2^2$
$=$
$\|vec\,\hat{\boldsymbol{X}}\|_2^2$

Fig. 3. (a) Random projection. Let $\boldsymbol{x}_i \in X$ be a point and $\hat{\boldsymbol{x}}_i = \boldsymbol{R}\boldsymbol{x}_i$. The distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is preserved in the projected space $\mathbb{R}^k$. (b) Differences in two random projection paths.

The random projection preserves the local topological structure of the vectors of an image pattern.

An efficient random projection is proposed as an improved version of the random projection[20]. Using spectrum spreading and circular convolution, we can speed up the random projection.

3.3 Two-Dimensional Random Projection

For a set of two-dimensional arrays $\{\boldsymbol{X}_i | \boldsymbol{X}_i \in \mathbb{R}^{m\times n}\}_{i=1}^N$ such that $E_i(\boldsymbol{X}_i) = 0$, setting $\boldsymbol{R}_L \in \mathbb{R}^{k_1\times m}$ and $\boldsymbol{R}_R \in \mathbb{R}^{k_2\times n}$ to be random projection matrices, we define the transform

$$\hat{\boldsymbol{X}}_i = \boldsymbol{R}_L \boldsymbol{X}_i \boldsymbol{R}_R^\top. \tag{15}$$

For the set $\hat{X} = \{\hat{\boldsymbol{X}}_i\}_{i=1}^N$, we have the following theorem.

Theorem 5. *$\hat{\boldsymbol{X}}_i \in \hat{X}$ and $\boldsymbol{X}_i \in X$ satisfy the Johnson-Lindenstrauss property.*

(*Proof*) From $\hat{\boldsymbol{X}}_i = \boldsymbol{R}_L \boldsymbol{X}_i \boldsymbol{R}_R$, we have the relation

$$vec\hat{\boldsymbol{X}}_i = (\boldsymbol{R}_L \otimes \boldsymbol{R}_R) vec\boldsymbol{X}_i \tag{16}$$

where $\boldsymbol{R}_L \otimes \boldsymbol{R}_R = \boldsymbol{R} \in \mathbb{R}^{k\times d}$ is a random projection matrix. Here, $k = k_1 \times k_2$ and $d = m \times n$. Therefore, for any $0 < \epsilon$ and set of X of N images $\{\boldsymbol{X}_1, \dots, \boldsymbol{X}_N\}$, $\hat{\boldsymbol{X}}_i$ and $\hat{\boldsymbol{X}}_j$ satisfy the property

$$(1-\epsilon)\|\boldsymbol{X}_j - \boldsymbol{X}_i\|_2 \le \|\hat{\boldsymbol{X}}_j - \hat{\boldsymbol{X}}_i\|_2 \le (1+\epsilon)\|\boldsymbol{X}_j - \boldsymbol{X}_i\|_2. \tag{17}$$

Here, setting $\|\boldsymbol{A}\|_2$ to be the Frobenius norm of matrix $\boldsymbol{A}$, the relation

$$\|\boldsymbol{x}_i\|^2 = \|vec\boldsymbol{X}_i\|^2 = \|\boldsymbol{X}_i\|_2^2 \tag{18}$$

is satisfied for $\boldsymbol{x}_i = vec\boldsymbol{X}_i$. Therefore, by replacing the Euclidean norm of $vec\boldsymbol{X}_i$ with the Frobenius norm of $\boldsymbol{X}_i$, we have the statement of the theorem. (*Q.E.D.*)

Considering the two-dimensional array as a second-order tensor, we can reduce the dimension of the tensorial data to an arbitrary dimension. The random projection preserves the topology of the tensor in the function space, since the Frobenius norm of a tensor is preserved.

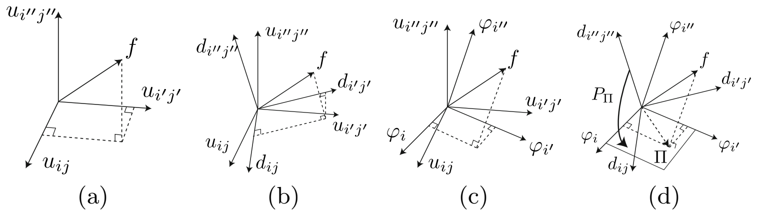


Fig. 4. Image representations in three coordinates. (a) $u_{ij}, u_{i'j'}, u_{i''j''}$ are the bases which represent each pixel of an image. (b) $d_{ij}, d_{i'j'}, d_{i''j''}$ are the bases of the DCT. (c) $\varphi_i, \varphi_{i'}, \varphi_{i''}$ are the bases of the PCA. (d) There is a projection P_Π which projects the image f to the linear subspace $\Pi = \{\varphi_i, \varphi_{i'}\}$ from the space spanned by the cosine bases.

3.4 Two-Dimensional Discrete Cosine Transform

For a real image, the discrete Fourier transformation can be replaced with the discrete cosine transform (DCT). Furthermore, the eigenfunction and eigendistribution of the DCT approximately coincide with those of the Karhunen-Loeve expansion for images. Moreover, in special cases, the reduction using the DCT is equal to the reduction using the PCA. Figure 4 illustrates the representation of an image by the DCT and PCA and the special case. The DCT and PCA are unitary transforms; therefore, these bases are related to a rotation transformation.

4 Classification Methods

4.1 Subspace Method

Setting H to be the space of patterns, we assume that in H the inner product (f, g) is defined. Furthermore, we define the Schatten product $\langle f, g \rangle$, which is an operator from H to H. Let $f \in H$ and P_k, $i = 1, \dots, N$ be a pattern and an operator for the ith class where the ith class is defined as

$$\mathcal{C}_i = \{f \mid P_i f = f,\ P_i^* P_i = I\}. \tag{19}$$

Since patterns have perturbations, we define the ith class as

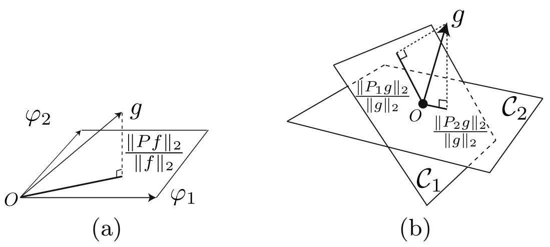
$$\mathcal{C}_i(\delta) = \{f \mid \|P_i f - f\|_2 \ll \delta,\ P_i^* P_i = I\}, \tag{20}$$

where δ is a small perturbation of the pattern and a small value, respectively. For input $g \in H$ and class $\mathcal{C}_i$, we define the similarity and classification criteria as

$$\theta_i = \angle(C_i(\delta), g),\ 0 < \theta_i < {}^{\exists}\theta_0 \rightarrow g \in \mathcal{C}_i(\delta), \tag{21}$$

since we define the angle between input pattern g and the space of the pattern as

$$\theta_i = \cos^{-1} \frac{\|P_i g\|_2}{\|g\|_2}. \tag{22}$$


φ_2
g
$\frac{\|Pf\|_2}{\|f\|_2}$
O
φ_1
g
$\frac{\|P_1 g\|_2}{\|g\|_2}$
O
$\mathcal{C}_2$
$\frac{\|P_2 g\|_2}{\|g\|_2}$
$\mathcal{C}_1$

(a) (b)

Fig. 5. (a) Geometric property of the SM. Let φ_1 and φ_2 be the bases of a class pattern. For input g, similarity is defined as the orthogonal projection to the pattern space. (b) Multiclass recognition using the SM. Let P_1 and P_2 be operators for subspace $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. Input g is labeled as being in the 1st class, since the subspaces $\mathcal{C}_1$ has the longest projection length of g.

The angle between the input pattern and pattern space represents their similarity.

For input $g \in H$, we construct

$$\mathcal{C}_g = \{g \mid Qg = g,\, Q^*Q = I\}, \tag{23}$$

$$\mathcal{C}_g(\delta) = \{g \mid \|Qg - g\|_2 \ll \delta,\, Q^*Q = I\}. \tag{24}$$

Then, we define the generalisation of Eq. (21) as

$$\theta_i = \angle(\mathcal{C}_i(\delta), \mathcal{C}_g(\delta)),\ \theta < \theta_i < {}^{\exists}\theta_0 \to \mathcal{C}_g(\delta) \in \mathcal{C}_i(\delta), \tag{25}$$

where $\sharp|\mathcal{C}_g \backslash C_k(\delta) \cap \mathcal{C}_g(\delta)| \ll \delta$.

We construct an operator P_i for $f_i \in \mathcal{C}_i$ such that

$$E(\|f - P_i f\|_2) \to \min, \quad P_i^* P_i = I, \tag{26}$$

where $f \in \mathcal{C}_i$, I is the identity operator and E is the expectation over H.

For practical calculation, we set $\{\varphi_j\}_{j=1}^n$ to be the eigenfunction of $M = E\langle f, f\rangle$. We define the eigenfunction of M as $\|\varphi_j\|_2 = 1$ for eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_j \geq \cdots \geq \lambda_n$. Therefore, operator P is defined as $P_n = \sum_{j=1}^n \langle \varphi_j, \varphi_j\rangle$.

Figure 5(a) shows the basic idea of the subspace method (SM). To identify whether the input data are in the subspace of the classes or not, we calculate the angle between the input data and the subspace of the classes. If g belongs to the space, the length of the orthogonal projection is close to 1. Figure 5(b) shows multiclass recognition using the SM.

4.2 Mutual Subspace Method

Let P_i and Q be operators for $\mathcal{C}_i$ and $\mathcal{C}_g$, respectively. If a pattern is expressed as an element of the linear subspace $\mathcal{C}_g = \{f | Qf = f, Q^*Q = I\}$, we are required to compute the angle between $\mathcal{C}_g$ and C_i as the extension of the classical pattern

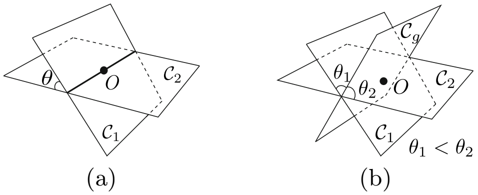


Fig. 6. (a) Angle between two linear subspaces $\mathcal{C}_1$ and $\mathcal{C}_2$. The minimal angle between the two subspace is 0. However, in the MSM, we adopt the angle θ to indicate the similarity between two subspaces. (b) Multiclass recognition using the MSM. For input subspace $\mathcal{C}_g$, let θ_1 and θ_2 be its angles relative to $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. The input subspace $\mathcal{C}_g$ is labeled as being in the 1 class since $\theta_1 < \theta_2$.

recognition such that $\operatorname{rank} Q = 1$ and $\dim \mathcal{C}_g = 1$. Then, the angle between P_i and Q is computed by

$$\cos\theta_i = \max\, E\left(\frac{\|QP_if\|_2}{\|f\|_2}\right) = \max\, E\left(\frac{\|P_iQf\|_2}{\|f\|_2}\right), \tag{27}$$

where f satisfies $\|f\|_2 \neq 0$. Figure 6(a) shows the angle between two subspaces.

For practical calculation, we adopt the following theorem[21].

Theorem 6. *The angle between $\mathcal{C}_i$ and $\mathcal{C}_g$ is calculated as the maximum eigenvalue of P_iQP_i and QP_iQ.*

Figure 6(b) shows multiclass recognition using the mutual subspace method (MSM).

4.3 Constraint Mutual Subspace Method

We next define a common subspace. For $f \neq g$, in a common subspace $AP_Cf = P_Cg$ and $\|AP_Cf\|_2 = \|P_Cg\|_2$ are satisfied, where A and P_C are an appropriate equi-affine operation and orthogonal projection, respectively. All patterns in a common subspace are written in terms of the equi-affine transform. For the projections $\{P_i\}_{i=1}^N$ to the class $\{\mathcal{C}_i\}_{i=1}^N$, we have the operator for the common subspace

$$P_C = \prod_{i=1}^{N} P_i. \tag{28}$$

Therefore, we define the constraint subspace as the operator $Q_C = I - P_C$, where I is the identity operator. Using the operator Q_C, we can calculate the angle in the constraint subspace by Eq. (27). The orthogonal projection for the constraint subspace is a nonexpansion mapping.

In the constraint subspace, the angle $\theta_{C,i}$ between the projected reference subspace $\mathcal{C}_{C,i}, i = 1, \ldots, N$ and the projected input subspace $\mathcal{C}_{C,g}$ is defined as

$$\cos\theta_{C,i} = \max\, E\left(\frac{\|Q_cQQ_cP_if\|_2}{\|f\|_2}\right) = \max\, E\left(\frac{\|Q_cP_iQ_cQf\|_2}{\|f\|_2}\right), \tag{29}$$

where f satisfies $\|f\|_2 \neq 0$.

The generalised difference subspace $\mathcal{D}_k$ is defined for the constraint mutual subspace method (CMSM) as the constraint subspace[12]. For the construction of the operator Q_c for $\mathcal{D}_k$, setting $\{\psi_j\}_{j=1}^{N_C}$ to be the eigenfunction of $G = \sum_{i=1}^{N} P_i$, we define the eigenfunction of G as

$$G\psi_j = \lambda_j \psi_j, \quad \|\psi_j\|_2 = 1, \tag{30}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_j \geq \cdots \geq \lambda_{N_c}$ and $N_c = n \times N$. Using $\{\psi_j\}_{j=1}^{N}$, the operator Q_c is defined as

$$Q_C = \sum_{j=1}^{k} \langle \psi_{N_c-(j-1)}, \psi_{N_c-(j-1)} \rangle, \tag{31}$$

where $k < N_C$. The dimension k of the difference subspace is selected experimentally.

If the dimension of the common space is unity and the base of this space corresponds to the first eigenfunction, which is associated with the largest eigenvalue of the covariance of the space of the pattern space, the operation is called the constant normalisation of patterns.

According to Theorem 3, if an input pattern has a high similarity to a class in the MSM, the input pattern has higher similarity to the class in the CMSM than one in the MSM. However, the CMSM does not guarantee the preservation of dissimilarity according to [12]. The projection onto the constraint subspace is a nonexpansion mapping, therefore the angle between the two subspaces becomes small.

4.4 Two-Dimensional Tensorial Subspace Method

As an extension of the subspace method for vector data, we introduce a linear subspace method for a bilinear array as two-dimensional tensorial subspace method (2DTSM). For a bilinear array $\boldsymbol{X}$, setting $\boldsymbol{P}_L$ and $\boldsymbol{P}_R$ to be orthogonal projections, we call the operation

$$\boldsymbol{Y} = \boldsymbol{P}_L \boldsymbol{X} \boldsymbol{P}_R \tag{32}$$

the orthogonal projection of $\boldsymbol{X}$ to $\boldsymbol{Y}$. Therefore, using this expression for a collection of bilinear forms $\{\boldsymbol{X}\}_{i=1}^{n}$, such that $E_i(\boldsymbol{X}_i) = 0$, the solutions of

$$J(\boldsymbol{P}_L, \boldsymbol{P}_R) = E_i \left(\frac{\|\boldsymbol{P}_L \boldsymbol{X}_i \boldsymbol{P}_R\|_2}{\|\boldsymbol{X}_i\|_2} \right) \rightarrow \max, \ w.r.t. \ \boldsymbol{P}_L^* \boldsymbol{P}_L = \boldsymbol{I}, \ \boldsymbol{P}_R^* \boldsymbol{P}_R = \boldsymbol{I} \tag{33}$$

define a bilinear subspace which approximates $\{\boldsymbol{X}\}_{i=1}^{n}$. Here, norm $\|\boldsymbol{X}\|_2$ for matrix $\boldsymbol{X}$ represents the Frobenius norm. Therefore, using the solutions of Eq. (33), if an input data array $\boldsymbol{G}$ satisfies the condition

$$\arg \left(\max_i \frac{\|\boldsymbol{P}_{Li} \boldsymbol{G} \boldsymbol{P}_{Ri}\|_2}{\|\boldsymbol{G}\|_2} \right) = \{\boldsymbol{P}_{Lk}, \boldsymbol{P}_{Rk}\}, \tag{34}$$

Table 1. Details of each database

	# class	# data /class	image size [pixel]	vectorised size	reduced dimension	reduced dimensions of image [pixel]
Yale B	38	64	192×168	32,256	1024	32×32
ETH80	30	41	128×128	16,384	1024	32×32
MNIST	10	7,000	28×28	784	225	15×15

(a) (b) (c)

Fig. 7. Examples of data. (a) Yale B. (b) ETH80. (c) MNIST.

we conclude that $\boldsymbol{G} \in \mathcal{C}_k(\delta)$ when $\mathcal{C}_k = \{\boldsymbol{X} \mid \|P_{Lk}\boldsymbol{X}P_{Rk} - \boldsymbol{X}\|_2 \ll \delta\}$. In practical computation to find the projections P_L and P_R, we adopt the marginal eigenvalue (MEV)[15].

5 Experiments

We evaluate the performance of the dimension-reduction methods using cropped versions of the extended Yale B database[22], the ETH80 database[23] and the MNIST dataset[24]. Table 1 lists the details of the three databases. Figure 7 shows examples of images for each database. We adopt downsampling (DS), the Gaussian-based pyramid transform (PT), the two-dimensional discrete cosine transform (2DDCT), random projection (RP) and two-dimensional random projection (2DRP) as the dimension-reduction methods. We calculate the recognition rate for each pair of dimension reduction methods and classifiers. The RP is applied to images after their vectorisation. The other reduction methods are applied before the vectorisation of images. For the Yale B and ETH80 databases,

Table 2. Dimensions of the class subspace in classification

	# query	# basis	dimension of constraint subspace	
Yale B	SM	1	1∼32	-
	MSM	3,5,7,9	1∼10	-
	CMSM	3	3	938,950,960,. . . ,1000,1024
	2DTSM	1	1×1 ∼ 32×32	-
ETH 80	SM	1	1∼21	-
	MSM	3,5,7,9	1∼10	-
	CMSM	5	5	938,950,960,. . . ,1000,1024
	2DTSM	1	1×1 ∼ 32×32	-
MNIST	SM	1	1∼225	-
	MSM	3,5,7,9	1∼10	-
	CMSM	3	3	10,20,. . . ,220,225
	2DTSM	1	1×1 ∼ 15×15	-

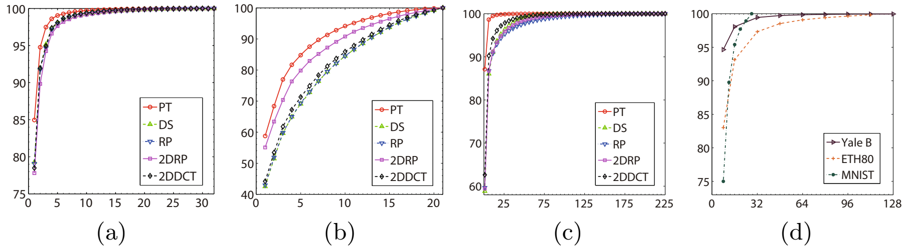


Fig. 8. Cumulative contribution ratios. (a) Yale B. (b) ETH80. (c) MNIST. (d) Preservation of power in the 2DDCT. In (a), (b) and (c), red, green, blue, magenta and black curves represent the cumulative contribution ratio of PT, DS, 2DRP and 2DDCT, respectively. x and y axes represent cumulative contribution ratios and ith eigenvectors, respectively. In (d), raisin, carrot and mint curves represent Yale B, ETH80 and MNIST database, respectively. x and y axes represent power preservation rates, and sizes of width and height for image, respectively.

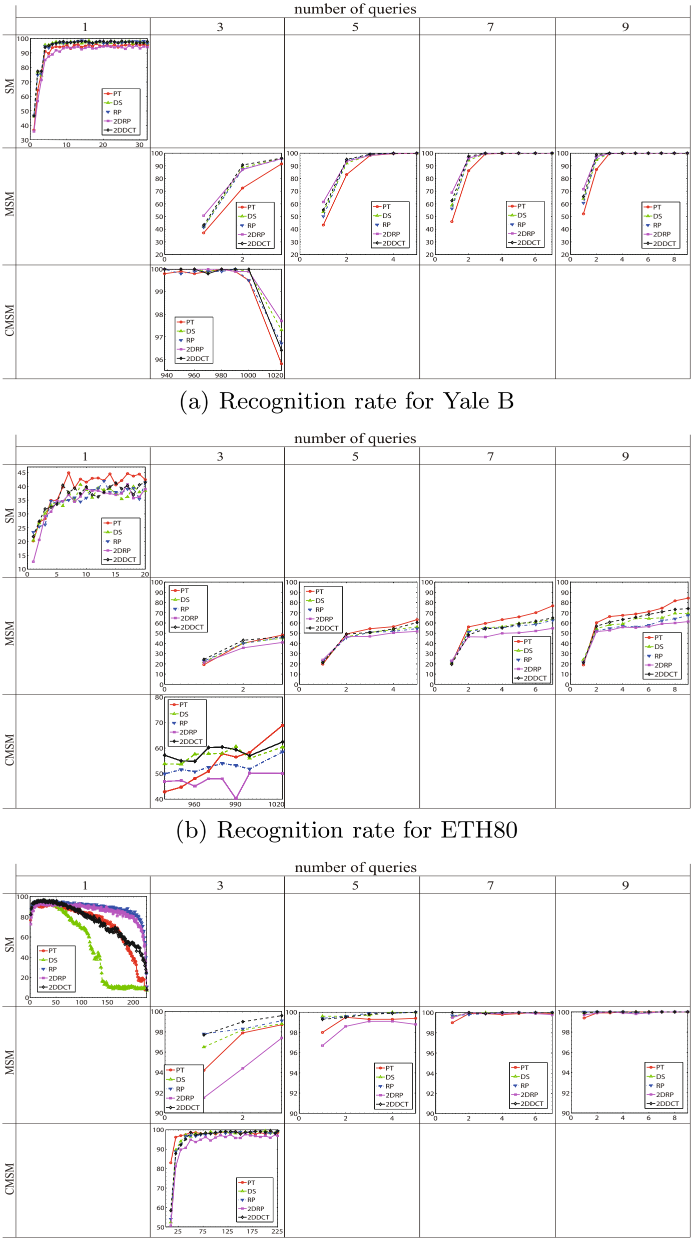
images labelled with even numbers are used as training data and the others are used as test data. The MNIST dataset is divided into training and test data in advance. The recognition rates are the successful label-estimation ratios of 1000 iterations in the estimations. In each estimation, queries are randomly chosen from the test data. For recognition, we use the SM, MSM, CMSM and 2DTSM as classifiers. The 2DTSM adopts the matrix representing the image as a feature. The other methods adopt the vector representing the image as a feature. Tables 2 illustrates the dimension of the class subspace used in the recognition for each database.

For the three databases, Figs. 8(a), (b) and (c) show the cumulative contribution ratios of the eigenvalues for a each class. The blue, red and green curves represent the ratios for the Yale B, ETH80 and MNIST databases, respectively. Figure 8(d) illustrates the mean preservation ratio of the power of the 2DDCT for images in the three databases. For the three databases, Figs. 9(a), (b) and (c) show the recognition rates of the SM, MSM and CMSM, respectively. In these figures, the red, green, blue, magenta and black curves represent the recognition rates of the PT, DS, RP, 2DRP and 2DDCT, respectively. Figure 10 shows the recognition rate of the 2DTSM for the three databases. In Fig. 10, the red, green, magenta and black curves represent the recognition rates of the PR, DS, 2DRP and 2DDCT, respectively.

As shown in Fig. 8, for all databases, the PT has the highest cumulative contribution in a low-dimensional linear subspace. Since the PT is an nonexpansion mapping, distances among data become small. Figure 8(d) shows that the low-frequency 32×32 bases have almost the same power as the 192×168 pixel image in the Yale B and ETH80 databases. For the MNIST database, the low-frequency 24 × 24 bases have almost the same power as a 28×28 pixel image. That is, the images are potentially compressible.

For the Yale B database, Fig. 8(a) shows that image patterns are efficiently approximated in a low-dimensional linear subspace. Figure 9(a) shows that the recognition rates of all classifiers are larger than 95% because discriminative features exist in the low-dimensional linear subspace and furthermore, the spectrum of these texture concentrates in low-frequency band. For a two-dimensional image with a small perturbation, using three eigenvectors we can approximately represent the image[9,11]. As shown in middle row of Fig. 9(a), using three or more bases, MSM has almost 100% classification. As shown in the bottom row of Fig. 9(a), the CMSM possesses a higher recognition rate than the MSM with a smaller number of bases. That is, the CMSM detects the common subspace for all classes, since the human face basically contains a common structures. Using three or more bases, the recognition rates for the three dimension-reduction methods are almost the same. For the ETH80 database, in contrast with the Yale B database, Fig. 8(b) shows that a discriminative low-dimensional linear subspace does not exist, since the cumulative contribution ratio is smaller than 95% in a low-dimensional subspace. Figure 9(b) illustrates that the SM has a recognition rate of less than 50%. The recognition rate of the MSM is smaller than 90%. In this case, we cannot obtain a discriminative low-dimensional linear subspace. The CMSM has a smaller recognition rate than the MSM, since the CMSM cannot find the optimal common structure of the linear subspace of all classes. Among the three classifiers, the PT has a larger recognition rate than the DS and RP. For the MNIST database, Fig. 8(c) shows that a discriminative low-dimensional subspace exists. Figure 9(c) illustrates that the recognition rates are larger than 95% for all classifiers. The CMSM has a smaller recognition rate than the MSM, since the CMSM cannot find a common subspace for all classes. Using 3 to 50 bases, the three dimension-reduction methods possess almost the same recognition rate. For the three databases, the 2DTSM gives almost the same results. The recognition rates of the PT, DS and 2DDCT are almost same. Using the 2DRP, none of the any classification methods can recognise the classes in any of the datasets. The width and height of images are too small to reduce the dimensions with a random projection, therefore, the distances among randomly projected images are not preserved. The 2DTSM has a smaller recognition rate than the SM, MSM and CMSM.

From these experiments, we observe that the Gaussian-based pyramid transform has a different recognition rate from the other methods for the SM, MSM and CMSM, since the pyramid transform is a nonexpansion mapping. As shown in the middle row of Fig. 9(a), the PT has the smallest recognition rate, whereas in the middle of Fig. 9(b), the PT has the highest recognition rate. In the bottom row of Fig. 9(b), the shape of the recognition rate for PT is different from those of the others. In the 2DTSM, all method have almost the same results. These results imply that RP works well comparing other methods if we have no *a priori* information for input data.

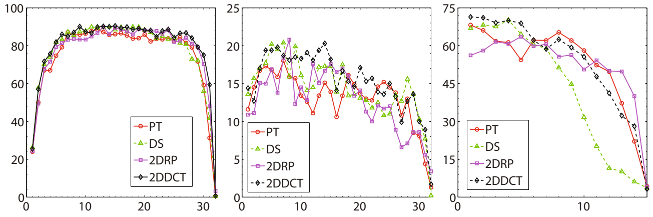


(a) Recognition rate for Yale B

(b) Recognition rate for ETH80

(c) Recognition rate for MNIST

Fig. 9. Recognition rates for each pair of dimension-reduction method and classifier. In each graph, x and y axes represent the number of bases and recognition rate[%], respectively. In each graph, red, green, magenta and brack curves represent the recognition rate of PT, DS, 2DRP and 2DDCT, respectively.



(a) (b) (c)

Fig. 10. Recognition rates for the 2DTSM. x and y axes represent sizes of images and recognition rates [%], respectively. (a) Yale B. (b) ETH80. (c) MNIST. In each graph, red, green, magenta and black curves represent the recognition rate of PT, DS, 2DRP and 2DDCT, respectively.

6 Conclusions

We experimentally evaluated the validity of dimension-reduction methods for image pattern recognition. The desired dimension-reduction method selects an appropriate low-dimensional subspace that preserves the information for classification.

By experimental evaluation of the reduction operation, we clarified the following properties. First, for three databases, the Gaussian-based pyramid transform has a higher cumulative contribution ratio than ones of the random projection and downsampling. Second, using feature vectors for recognition, the pyramid transform has the same or a higher recognition rate than the random projection and downsampling. The pyramid transform preserves the local geometry and topology of a image. However, it changes distances and angles among the vectors used as data since it is a nonexpansion mapping. Third, the using features of images for recognition, the pyramid transform, downsampling, two-dimensional random projection and the two-dimensional discrete transform have almost the same recognition rate. These reduction methods preserve the geometry and topology of images. Fourth, using the feature vectors results in a higher recognition rate than using a feature of images. From the fourth property, the classification should be computed in a vector space. Therefore, pyramid transform must not be used for classification since it changes the topology of the vector space. In contrast, the random projection preserves the topology of the vector space. These results imply that RP works well comparing other methods if we have no *a priori* information for input data. Therefore, for the application to remote exploration and field robot vision, the RP has theoretical and practical priorities, since the camera captures sceneries and sequences without ground truth in these applications.

References

1. Xu, Y., Zhang, D., Yang, J., Yang, J.Y.: An approach for directly extracting features from matrix data and its application in face recognition. Neurocomputing 71, 1857–1865 (2008)
2. Su, T.H., Zhang, T.W., Guan, D.J., Huang, H.J.: Off-line recognition of realistic chinese handwriting using segmentation-free strategy. Pattern Recognition 42, 167–182 (2009)
3. Leonardis, A., Bischof, H., Maver, J.: Multiple eigenspaces. Pattern Recognition 35, 2613–2627 (2002)
4. Park, C.H., Park, H.: Fingerprint classification using fast Fourier transform and nonlinear discriminant analysis. Pattern Recognition 38, 495–503 (2005)
5. Park, H., Park, K.: Iris recognition based on score level fusion by using SVM. Pattern Recognition Letters 28, 2019–2028 (2007)
6. Borgwardt, K.M.: Graph Kernels. PhD thesis, Ludwig-Maximilians-Universität München (2007), http://edoc.ub.uni-muenchen.de/7169/1/Borgwardt_KarstenMichael.pdf
7. Gärtnera, T., Le, Q.V., Smola, A.J.: A short tour of kernel methods for graphs. Technical report (2006), http://cs.stanford.edu/~quocle/srl-book.pdf
8. Xu, Y., Zhong, A., Yang, J., Zhang, D.: LPP solution schemes for use with face recognition. Pattern Recognition 43, 4165–4176 (2010)
9. Iijima, T.: Theory of pattern recognition. Electronics and Communications in Japan, 123–134 (1963)
10. Murase, H., Nayar, S.K.: Illumination planning for object recognition using parametric eigenspace. IEEE Trans. Pattern Analysis and Machine Intelligence 16, 1219–1227 (1994)
11. Maeda, K.: From the subspace methods to the mutual subspace method. In: Cipolla, R., Battiato, S., Farinella, G.M. (eds.) Computer Vision. SCI, vol. 285, pp. 135–156. Springer, Heidelberg (2010)
12. Fukui, K., Yamaguchi, O.: Face recognition using multi-viewpoint patterns for robot vision. Robotics Research 15, 192–201 (2005)
13. Yang, J., Zhang, D., Frangi, A., Yang, J.Y.: Two-dimensional PCA: a new approach to appearance-based face representation and recognition. IEEE Trans. Pattern Analysis and Machine Intelligence 26, 131–137 (2004)
14. Benito, M., Peña, D.: A fast approach for dimensionality reduction with image data. Pattern Recognition 38, 2400–2408 (2005)
15. Otsu, N.: Mathematical Studies on Feature Extraction in Pattern Recognition. PhD thesis, Electrotechnical Laboratory (1981)
16. Shen, L., Bai, L.: Mutual boost learning for selecting gabor features for face recognition. Pattern Recognition Letters 27, 1758–1767 (2006)
17. Burt, P.J., Andelson, E.H.: The Laplacian pyramid as a compact image code. IEEE Trans. Communications 31, 532–540 (1983)
18. Vempala, S.S.: The Random Projection Method. DIMACS, 65 (2004)
19. Johnson, W., Lindenstrauss, J.: Extensions of Lipschitz maps into a Hilbert space. Contemporary Mathematics 26, 189–206 (1984)
20. Sakai, T., Imiya, A.: Practical algorithms of spectral clustering: Toward large-scale vision-based motion analysis. In: Wang, L., Zhao, G., Cheng, L., Pietikäinen, M. (eds.) Machine Learning for Vision-Based Motion Analysis, pp. 3–26. Springer (2011)

21. Björck, A., Golub, G.H.: Numerical methods for computing angles between linear subspaces. Mathematics of Computation 27, 579–594 (1975)
22. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Analysis and Machine Intelligence 23, 643–660 (2001)
23. Leibe, B., Schiele, B.: Analyzing appearance and contour based methods for object categorization. Proc. of Computer Vison and Pattern Recognition 2, 409–415 (2003)
24. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. of the IEEE 86, 2278–2324 (1998)

Efficient Regression in Metric Spaces via Approximate Lipschitz Extension[*]

Lee-Ad Gottlieb[1], Aryeh Kontorovich[2,**], and Robert Krauthgamer[3,***]

[1] Ariel University
[2] Ben-Gurion University of the Negev
[3] Weizmann Institute of Science

Abstract. We present a framework for performing efficient regression in general metric spaces. Roughly speaking, our regressor predicts the value at a new point by computing a Lipschitz extension — the smoothest function consistent with the observed data — while performing an optimized structural risk minimization to avoid overfitting. The offline (learning) and online (inference) stages can be solved by convex programming, but this naive approach has runtime complexity $O(n^3)$, which is prohibitive for large datasets. We design instead an algorithm that is fast when the doubling dimension, which measures the "intrinsic" dimensionality of the metric space, is low. We make dual use of the doubling dimension: first, on the statistical front, to bound fat-shattering dimension of the class of Lipschitz functions (and obtain risk bounds); and second, on the computational front, to quickly compute a hypothesis function and a prediction based on Lipschitz extension. Our resulting regressor is both asymptotically strongly consistent and comes with finite-sample risk bounds, while making minimal structural and noise assumptions.

Keywords: metric space, regression, convex program.

1 Introduction

The classical problem of estimating a continuous-valued function from noisy observations, known as *regression*, is of central importance in statical theory with a broad range of applications, see e.g. [BFOS84, Nad89, GKKW02]. When no structural assumptions concerning the target function are made, the regression problem is termed *nonparametric*. Informally, the main objective in the study of nonparametric regression is to understand the relationship between the regularity conditions that a function class might satisfy (e.g., Lipschitz or Hölder continuity, or sparsity in some representation) and the minimax risk convergence

[*] A full version appears as arXiv:1111.4470 [GKK11].

[**] This research was partially supported by the Israel Science Foundation (grant #1141/12) and the Lynne and William Frankel Center for Computer Science.

[***] This work was supported in part by a US-Israel BSF grant #2010418, and by the Citi Foundation.

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 43–58, 2013.

rates [Tsy04, Was06]. A further consideration is the computational efficiency of constructing the regression function.

The general (univariate) nonparametric regression problem may be stated as follows. Let $(\mathcal{X}, \rho)$ be a metric space, namely $\mathcal{X}$ is a set of points and ρ a distance function, and let $\mathcal{H}$ be a collection of functions ("hypotheses") $h : \mathcal{X} \to [0, 1]$. (Although in general, h is not explicitly restricted to have bounded range, typical assumptions on the diameter of $\mathcal{X}$ and the noise distribution amount to an effective truncation.) The space $\mathcal{X} \times [0, 1]$ is endowed with some fixed, unknown probability distribution μ, and the learner observes n iid draws $(X_i, Y_i) \sim \mu$. The learner then seeks to fit the observed data with some hypothesis $h \in \mathcal{H}$ so as to minimize the *risk*, usually defined as the expected loss $\mathbf{E}\,|h(X) - Y|^q$ for $(X, Y) \sim \mu$ and some $q \geq 1$.

Two limiting assumptions have traditionally been made when approaching this problem: (i) the space $\mathcal{X}$ is Euclidean and (ii) $Y_i = h^*(X_i) + \xi_i$, where h^* is the target function and ξ_i is an iid noise process, often taken to be Gaussian. Although our understanding of nonparametric regression under these assumptions is quite elaborate, little is known about nonparametric regression in the absence of either assumption.

The present work takes a step towards bridging this gap. Specifically, we consider nonparametric regression in an arbitrary metric space, while making no assumptions on the distribution of the data or the noise. Our results rely on the structure of the metric space only to the extent of assuming that the metric space has a low "intrinsic" dimensionality. The dimension in question is the *doubling dimension* of $\mathcal{X}$, denoted $\mathrm{ddim}(\mathcal{X})$, which was introduced by [GKL03] based on earlier work of [Cla99], and has been since utilized in several algorithmic contexts, including networking, combinatorial optimization, and similarity search, see e.g. [KSW09, KL04, BKL06, HM06, CG06, Cla06]. Following the work in [GKK10] on classification problems, our risk bounds and algorithmic runtime bounds are stated in terms of the doubling dimension of the ambient space and the Lipschitz constant of the regression hypothesis, although neither of these quantities need be known in advance.

Our Results. We consider two kinds of risk: L_1 (mean absolute) and L_2 (mean square). More precisely, for $q \in \{1, 2\}$ we associate to each hypothesis $h \in \mathcal{H}$ the empirical L_q-risk

$$R_n(h) = R_n(h, q) = \frac{1}{n} \sum_{i=1}^{n} |h(X_i) - Y_i|^q \tag{1}$$

and the (expected) L_q-risk

$$R(h) = R(h, q) = \mathbf{E}\,|h(X) - Y|^q = \int_{\mathcal{X} \times [0,1]} |h(x) - y|^q \, \mu(dx, dy). \tag{2}$$

It is well-known that $h(x) = \mathbf{M}[Y \,|\, X = x]$ (where $\mathbf{M}$ is the median) minimizes $R(\cdot, 1)$ over all integrable $h \in [0, 1]^{\mathcal{X}}$ and $h(x) = \mathbf{E}[Y \,|\, X = x]$ minimizes $R(\cdot, 2)$.

However, these expressions are of little use as neither is computable without knowledge of μ. To circumvent this difficulty, we minimize the empirical L_q-risk and assert that the latter is a good approximation of the expected risk, provided $\mathcal{H}$ meets certain regularity conditions.

To this end, we define the following random variable, termed *uniform deviation*:

$$\Delta_n(\mathcal{H}) = \Delta_n(\mathcal{H}, q) = \sup_{h \in \mathcal{H}} |R_n(h) - R(h)| \,. \tag{3}$$

It is immediate that

$$R(h) \leq R_n(h) + \Delta_n(\mathcal{H}) \tag{4}$$

holds for all $h \in \mathcal{H}$ (i.e., the expected risk of any hypothesis does not exceed its empirical risk by much), and it can further be shown [BBL05] that $R(\hat{h}) \leq R(h^*) + 2\Delta_n(\mathcal{H})$, where $\hat{h} \in \mathcal{H}$ is a minimizer of the empirical risk and $h^* \in \mathcal{H}$ is a minimizer of the expected risk (i.e., the expected risk of $\hat{h}$ does not exceed the risk of the best admissible hypothesis by much).

Our contribution is twofold: statistical and computational. The algorithm in Theorem 3.1 computes an η-additive approximation to the empirical risk minimizer in time $\eta^{-O(\mathrm{ddim}(\mathcal{X}))} n \log^3 n$. This hypothesis can be evaluated on new points in time $\eta^{-O(\mathrm{ddim}(\mathcal{X}))} \log n$. The expected risk of this hypothesis decays as the empirical risk plus $1/\mathrm{poly}(n)$. Our bounds explicitly depend on the doubling dimension, but the latter may be efficiently estimated from the data, see e.g. [KL04, CG06, GK10, GKK13].

Related Work. There are many excellent references for classical Euclidean nonparametric regression assuming iid noise, see for example [GKKW02, BFOS84, DGL96]. For metric regression, a simple risk bound follows from classic VC theory via the pseudo-dimension, see e.g. [Pol84, Vap95, Ney06]. However, the pseudo-dimension of many non-trivial function classes, including Lipschitz functions, grows linearly with the sample size, ultimately yielding a vacuous bound. An approach to nonparametric regression based on empirical risk minimization, though only for the Euclidean case, may already be found in [LZ95]; see the comprehensive historical overview therein. Indeed, Theorem 5.2 in [GKKW02] gives a kernel regressor for Lipschitz functions that achieves the minimax rate. Note however that (a) the setting is restricted to Euclidean spaces; and (b) the cost of evaluating the hypothesis at a new point grows linearly with the sample size (while our complexity is roughly logarithmic). As noted above, another feature of our approach is its ability to give efficiently computable finite-sample bounds, as opposed to the asymptotic convergence rates obtained in [GKKW02, LZ95] and elsewhere.

More recently, risk bounds in terms of doubling dimension and Lipschitz constant were given in [Kpo09], assuming an additive noise model, and hence these results are incomparable to ours; for instance, these risk bounds worsen with an increasingly smooth regression function. Following up, a regression technique based on random partition trees was proposed in [KD11], based on mappings between Euclidean spaces and assuming an additive noise model. Another recent advance in nonparametric regression was Rodeo [LW08], which escapes the curse of dimensionality by adapting to the sparsity of the regression function.

Our work was inspired by the paper of von Luxburg and Bousquet [vLB04], who were apparently the first to make the connection between Lipschitz classifiers in metric spaces and large-margin hyperplanes in Banach spaces, thereby providing a novel generalization bound for nearest-neighbor classifiers. They developed a powerful statistical framework whose core idea may be summarized as follows: to predict the behavior at new points, find the smoothest function consistent with the training sample. Their work raises natural algorithmic questions like how to estimate the risk for a given input, how to perform model selection (Structural Risk Minimization) to avoid overfitting, and how to perform the learning and prediction quickly. Follow-up work [GKK10] leveraged the doubling dimension simultaneously for statistical and computational efficiency, to design an efficient classifier for doubling spaces. Its key feature is an efficient algorithm to find the optimal balance between the empirical risk and the penalty term for a given input. Minh and Hoffman [MH04] take the idea in [vLB04] in a more algebraic direction, establishing a representer theorem for Lipschitz functions on compact metric spaces.

2 Bounds on Uniform Deviation via Fat Shattering

In this section, we derive tail bounds on the uniform deviation $\Delta_n(\mathcal{H})$ defined in (3) in terms of the the smoothness properties of $\mathcal{H}$ and the doubling dimension of the underlying metric space $(\mathcal{X}, \rho)$.

2.1 Preliminaries

We rely on the powerful framework of fat-shattering dimension developed by [ABCH97], which requires us to incorporate the value of a hypothesis and the loss it incurs on a sample point into a single function. This is done by associating to any family of hypotheses $\mathcal{H}$ mapping $\mathcal{X} \mapsto [0,1]$, the induced family $\mathcal{F} = \mathcal{F}_{\mathcal{H}}^q$ of functions mapping $\mathcal{X} \times [0,1] \mapsto [0,1]$ as follows: for each $h \in \mathcal{H}$ the corresponding $f = f_h^q \in \mathcal{F}_{\mathcal{H}}^q$ is given by

$$f_h^q(x,y) = |h(x) - y|^q \,, \qquad q \in \{1,2\} \,. \tag{5}$$

In a slight abuse of notation, we define the uniform deviation of a class $\mathcal{F}$ of $[0,1]$-valued functions over $\mathcal{X} \times [0,1]$:

$$\Delta_n(\mathcal{F}) = \sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} f(X_i, Y_i) - \mathbf{E} f(X,Y) \right| , \tag{6}$$

where the expectation is over μ, as in (2). Obviously, $\Delta_n(\mathcal{F}_{\mathcal{H}}^q) = \Delta_n(\mathcal{H}, q)$.

2.2 Basic Generalization Bounds

Let us write

$$\mathcal{H}_L = \left\{ h \in [0,1]^{\mathcal{X}} : \|h\|_{\mathrm{Lip}} \le L \right\} \tag{7}$$

to denote the collection of $[0,1]$-valued L-Lipschitz functions on $\mathcal{X}$. We proceed to bound the γ-fat-shattering dimension of $\mathcal{F}^q_{\mathcal{H}_L}$.

Theorem 2.1. *Let $\mathcal{H}_L$ be defined on a metric space $(\mathcal{X}, \rho)$, where $\operatorname{diam}(\mathcal{X}) = 1$. Then*

$$\operatorname{fat}_\gamma(\mathcal{F}^q_{\mathcal{H}_L}) \le \left(1 + \frac{1}{\gamma^{(q+1)/2}}\right)\left(\frac{L}{\gamma^{(q+1)/2}}\right)^{\operatorname{ddim}(\mathcal{X})+1}$$

holds for $q \in \{1,2\}$ and all $0 < \gamma \le \frac{1}{2}$.

Proof. (Sketch) Fix a $\gamma > 0$ and recall what it means for $\mathcal{F}^q_{\mathcal{H}_L}$ to γ-shatter a set

$$S = (T, Z) = \{(t, z) : t \in \mathcal{X}, z \in [0,1]\}$$

(where $T \in \mathcal{X}^{|S|}$ and $Z \in [0,1]^{|S|}$): there exists some function $r \in \mathbb{R}^S$ such that for each label assignment $b \in \{-1,1\}^S$ there is an $f \in \mathcal{F}^q_{\mathcal{H}_L}$ satisfying $b(s)(f(s) - r(s)) \ge \gamma$ for all $s \in S$.

Put $K = \lceil \gamma^{-(q+1)/2} \rceil$ and define the map $\pi : S \to \{0, 1, \ldots, K\}$ by

$$\pi(s) = \pi(t,z) = \lfloor Kz \rfloor .$$

Thus, we may view S as being partitioned into $K+1$ buckets:

$$S = \bigcup_{k=0}^{K} \pi^{-1}(k). \tag{8}$$

Consider two points, $s = (t,z)$ and $s' = (t', z')$, belonging to some fixed bucket $\pi^{-1}(k)$. By construction, the following hold:

(i) $|z - z'| \le K^{-1} \le \gamma^{(q+1)/2}$

(ii) since $\mathcal{F}^q_{\mathcal{H}_L}$ γ-shatters S (and recalling (5)), there is an $h \in \mathcal{H}_L$ satisfying $|h(t) - z|^q \le r - \gamma$ and $|h(t') - z'|^q \ge r' + \gamma$ for some $\gamma \le r \le r' < 1 - \gamma$.

Conditions (i) and (ii) imply that

$$|h(t) - h(t')| \ge (r' + \gamma)^{1/q} - (r - \gamma)^{1/q} - |z - z'| \ge \gamma^{(q+1)/2}. \tag{9}$$

The fact that h is L-Lipschitz implies that $\rho(t,t') \ge |h(t) - h(t')|/L \ge \gamma^{(q+1)/2}/L$ and hence

$$\left|\pi^{-1}(k)\right| \le \left(\frac{L}{\gamma^{(q+1)/2}}\right)^{\operatorname{ddim}(\mathcal{X})+1} \tag{10}$$

for each $k \in \left\{0, 1, \ldots, \lceil \gamma^{-(q+1)/2} \rceil\right\}$. Together (8) and (10) yield our desired bound on $|S|$, and hence on the fat shattering dimension of $\mathcal{F}^q_{\mathcal{H}_L}$. □

The following generalization bound, implicit in [ABCH97], establishes the learnability of continuous-valued functions in terms of their fat-shattering dimension.

Theorem 2.2. *Let $\mathcal{F}$ be any admissible function class mapping $\mathcal{X} \times [0,1]$ to $[0,1]$ and define $\Delta_n(\mathcal{F})$ as in (6). Then for all $0 < \varepsilon < 1$ and all $n \geq 2/\varepsilon^2$,*

$$P(\Delta_n(\mathcal{F}) > \varepsilon) \leq 24n \left(\frac{288n}{\varepsilon^2}\right)^{d \log(24en/\varepsilon)} \exp(-\varepsilon^2 n/36)$$

where $d = \mathrm{fat}_{\varepsilon/24}(\mathcal{F})$.

Corollary 2.1. *Fix an $1 > \varepsilon > 0$ and $q \in \{1, 2\}$. Let $\mathcal{H}_L$ be defined on a metric space $(\mathcal{X}, \rho)$ and recall the definition of $\Delta_n(\mathcal{H}_L, q)$ in (3). Then for all $n \geq 2/\varepsilon^2$,*

$$P(\Delta_n(\mathcal{H}_L, q) > \varepsilon) \leq 24n \left(\frac{288n}{\varepsilon^2}\right)^{d \log(24en/\varepsilon)} \exp(-\varepsilon^2 n/36) \qquad (11)$$

where

$$d = \left(1 + \frac{1}{(\varepsilon/24)^{(q+1)/2}}\right) \left(\frac{L}{(\varepsilon/24)^{(q+1)/2}}\right)^{\mathrm{ddim}(\mathcal{X})+1}.$$

We can conclude from Corollary 2.1 that there exists $\epsilon(n, L, \delta)$ such that with probability at least $1 - \delta$,

$$\Delta_n(\mathcal{H}_L, q) \leq \epsilon(n, L, \delta), \qquad (12)$$

and by essentially inverting (11), we have

$$\epsilon(n, L, \delta) \leq O\left(\max\left\{\sqrt{\frac{\log(n/\delta)}{n}}, \left(\frac{L^{\mathrm{ddim}(\mathcal{X})+1}}{n} \log^2 n\right)^{\frac{1}{2+\frac{q+1}{2}(\mathrm{ddim}(\mathcal{X})+1)}}\right\}\right) \qquad (13)$$

(For simplicity, the dependence of $\epsilon(\cdot)$ on $\mathrm{ddim}(\mathcal{X})$ is suppressed.) This implies via (4) that

$$R(h) \leq R_n(h) + \epsilon(n, L, \delta)$$

uniformly for all $h \in \mathcal{H}_L$ with high probability.

2.3 Simultaneous Bounds for Multiple Lipschitz Constants

So far, we have established the following. Let $(\mathcal{X}, \rho)$ be a doubling metric space and $\mathcal{H}_L$ a collection of L-Lipschitz $[0,1]$-valued functions on $\mathcal{X}$. Then Corollary 2.1 guarantees that for all $\varepsilon, \delta \in (0,1)$ and $n \geq n_0(\varepsilon, \delta, L, \mathrm{ddim}(\mathcal{X}))$, we have

$$P(\Delta_n(\mathcal{H}_L) > \varepsilon) \leq \delta, \qquad (14)$$

where $\Delta_n(\mathcal{H}_L)$ is the uniform deviation defined in (3). Since our computational approach in Section 3 requires optimizing over Lipschitz constants, we will need a bound such as (14) that holds for many function classes of varying smoothness simultaneously. This is easily accomplished by stratifying the confidence parameter δ, as in [SBWA98]. We will need the following theorem:

Theorem 2.3. *Let*

$$\mathcal{H}^{(1)} \subset \mathcal{H}^{(2)} \subset \dots$$

be a sequence of function classes taking $\mathcal{X}$ to $[0,1]$ and let $p_k \in [0,1]$, $k = 1, 2, \dots$, be a sequence summing to 1. Suppose that $\epsilon : \mathbb{N} \times \mathbb{N} \times (0,1) \to [0,1]$ is a function such that for each $k \in \mathbb{N}$, with probability at least $1 - \delta$, we have

$$\Delta_n^q(\mathcal{H}^{(k)}) \leq \epsilon(n, k, \delta).$$

Then, whenever some $h \in \bigcup_{k \in \mathbb{N}} [\mathcal{H}^{(k)}]_\eta$ achieves empirical risk $R_n(h)$ on a sample of size n, we have that with probability at least $1 - \delta$,

$$R(h) \leq R_n(h) + \epsilon(n, k, \delta p_k) \quad \forall k. \tag{15}$$

Proof. An immediate consequence of the union bound. □

The structural risk minimization principle implied by Theorem 2.3 amounts to the following model selection criterion: choose an $h \in \mathcal{H}^{(k)}$ for which the right-hand side of (15) is minimized.

In applying Theorem 2.3 to Lipschitz classifiers in Section 3 below, we impose a discretization on the Lipschitz constant L to be multiples of $\frac{\eta}{24q}$. Formally, we consider the stratification $\mathcal{H}^{(k)} = \mathcal{H}_{L_k}$,

$$\mathcal{H}_{L_1} \subset \mathcal{H}_{L_2} \subset \dots,$$

where $L_k = k\eta$ with corresponding $p_k = 2^{-k}$ for $k = 1, 2, \dots$. This means that whenever we need a hypothesis that is an L-Lipschitz regression function, we may take $k = \lceil L\eta \rceil$ and use $\epsilon(n, k, \delta 2^{-k})$ as the generalization error bound. Note that all possible values of L are within a factor of 2 of the discretized sequence L_k.

3 Structural Risk Minimization

In this section, we address the problem of efficient model selection when given n observed samples. The algorithm described below computes a hypothesis that approximately attains the minimum risk over all hypotheses. Since our approximate Lipschitz extension algorithm will evaluate hypotheses up to an additive error, we define an η-perturbation $[\mathcal{H}]_\eta$ of a given hypothesis class $\mathcal{H}$ by

$$[\mathcal{H}]_\eta = \left\{ h' \in \mathbb{R}^{\mathcal{X}} : \exists h \in \mathcal{H} \text{ s.t. } \|h - h'\|_\infty \leq \eta \right\}. \tag{16}$$

Recall the risk bound achieved as a consequence of Theorem 2.3. In the full paper [GKK11], we extend this result to perturbations, showing that whenever some $h \in \bigcup_{k \in \mathbb{N}} \left[\mathcal{H}^{(k)}\right]_\eta$ achieves empirical risk $R_n(h)$ on a sample of size n, we have the following bound on $R(h)$, the true risk of h:

$$R(h) \leq R_n(h) + \epsilon(n, k, \delta p_k) + 24q\eta, \tag{17}$$

with probability at least $1-\delta$ (where the diameter of the point set has been taken as 1, and $\epsilon(n,k,\delta p_k) \geq \sqrt{2/n}$ is the minimum value of ϵ for which the right-hand side of (11) is at most δ). In the rest of this section, we devise an algorithm that computes a hypothesis that approximately minimizes our bound from (17) on the true risk, denoted henceforth

$$\tilde{R}_\eta(h) = R_n(h) + \epsilon(n,k,\delta p_k) + 24q\eta.$$

Notice that on the right-hand side, the first two terms depend on L, but only the first term depends on the choice of h, and only the third term depends on η.

Theorem 3.1. *Let (X_i, Y_i) for $i = 1,\dots,n$ be an iid sample drawn from μ, let $\eta \in (0,\frac{1}{4})$, and let h^* be a hypothesis that minimizes $\tilde{R}_\eta(h)$ over all $h \in \bigcup_{k\in\mathbb{N}} \left[\mathcal{H}^{(k)}\right]_\eta$. There is an algorithm that, given the n samples and η as input, computes in time $\eta^{-O(\mathrm{ddim}(\mathcal{X}))} n \log^3 n$ a hypothesis $h' \in \bigcup_{k\in\mathbb{N}} \left[\mathcal{H}^{(k)}\right]_\eta$ with*

$$\tilde{R}_\eta(h') \leq 2\tilde{R}_\eta(h^*). \tag{18}$$

Remark. We show in Theorem 4.1 how to quickly evaluate the hypothesis h' on new points.

The rest of Section 3 is devoted to describing an algorithm that realizes the bounds of Theorem 3.1 for $q=1$ (Sections 3.1 and 3.2) and $q=2$ (Section 3.3). In proving the theorem, we will find it convenient to compare the output h' to a hypothesis $\bar{h}$ that is smooth (i.e. Lipschitz but unperturbed). Indeed, let h^* be as in the theorem, and $\bar{h} \in \bigcup_{k\in\mathbb{N}} \mathcal{H}^{(k)}$ be a hypothesis that minimizes $\tilde{R}_\eta(\bar{h})$. Then $R_n(h^*) \leq R_n(\bar{h}) \leq R_n(h^*) + \eta$, and we get $\tilde{R}_\eta(h^*) \leq \tilde{R}_\eta(\bar{h}) \leq \tilde{R}_\eta(h^*) + \eta$. Accordingly, the analysis below will actually prove that $\tilde{R}_\eta(h') \leq 2\tilde{R}_\eta(\bar{h}) - 2\eta$, and then (18) will follow easily, essentially increasing the additive error by 2η. Moreover, once (18) is proved, we can use the above to conclude that $\tilde{R}_\eta(h') \leq 2\tilde{R}_0(\bar{h}) + O(\eta)$, which compares the risk bound of our algorithm's output h' to what we could possibly get using smooth hypotheses.

In the rest of this section we consider the n observed samples as fixed values, given as input to the algorithm, so we will write x_i instead of X_i.

3.1 Motivation and Construction

Suppose that the Lipschitz constant of an optimal *unperturbed* hypothesis $\bar{h}$ were known to be $L = \bar{L}$. Then $\epsilon(n,k,\delta p_k)$ is fixed, and the problem of computing both $\bar{h}$ and its empirical risk $R_n(\bar{h})$ can be described as the following optimization program with variables $f(x_i)$ for $i \in [n]$ to represent the assignments $h(x_i)$. Note it is a Linear Program (LP) when $q=1$ and a quadratic program when $q=2$.

$$\boxed{\begin{array}{lll} \text{Minimize} & \sum_{i\in[n]} |y_i - f(x_i)|^q & \\ \text{subject to} & |f(x_i) - f(x_j)| \leq L\cdot\rho(x_i,x_j) & \forall i,j \in [n] \\ & 0 \leq f(x_i) \leq 1 & \forall i \in [n] \end{array}} \tag{19}$$

It follows that $\bar{h}$ could be computed by first deriving $\bar{L}$, and then solving the above program. However, it seems that computing these exactly is an expensive computation. This motivates our search for an approximate solution to risk minimization. We first derive a target Lipschitz constant L' that "approximates" $\bar{L}$, in the sense that there exists an h' with Lipschitz constant L' which minimizes the objective $\max\{R_n(h'), \epsilon(n,k,\delta p_k)\}$. Notice that $R_n(h')$ may be computed by solving LP (19) using the given value L' for L. We wish to find such L' via a binary search procedure, which requires a method to determine whether a candidate L satisfies $L \le L'$, but since our objective need not be a monotone function of L, we cannot rely on the value of the objective at the candidate L. Instead, recall that the empirical risk term $R_n(h')$ is monotonically non-increasing, and the penalty term $\epsilon(n,k,\delta p_k)$ is monotonically non-decreasing, and therefore we can take L' to be the minimum value L for which $R_n(h') \le \epsilon(n,k,\delta p_k)$ (notice that both terms are right-continuous in L). Our binary search procedure can thus determine whether a candidate L satisfies $L \le L'$ by checking instead whether $R_n(h') \le \epsilon(n,k,\delta p_k)$.

Were the binary search on L to be carried out indefinitely (that is, with infinite precision), it would yield L' and a smooth hypothesis h' satisfying $\tilde{R}_\eta(h') \le 2\tilde{R}_\eta(\bar{h})$, where the factor 2 originates from the gap between maximum and summation. In fact, a slightly stronger bound holds:

$$\tilde{R}_\eta(h') - 24q\eta \le 2\max\{R_n(h'), \epsilon(n,k,\delta p_k)\} \le 2\big(R_n(\bar{h}) + \epsilon(n,k,\delta p_k)\big) \le 2\big(\tilde{R}_\eta(\bar{h}) - 24q\eta\big).$$

(In our actual LP solver below, h' will not be necessarily smooth, but rather a perturbation of a smooth hypothesis.) However, to obtain a tractable runtime, we fix an additive precision of η to the Lipschitz constant, and restrict the target Lipschitz constant to be a multiple of η. Notice that $\tilde{R}_\eta(\bar{h}) \le 2$ for sufficiently large n (since this bound can even be achieved by a hypothesis with Lipschitz constant 0), so by (13) it must be that $\bar{L} \le n^{O(1)}$, since $\bar{L}$ is the optimal Lipschitz constant. It follows that the binary search will consider only $O(\log(n/\eta))$ candidate values for L'.

To bound the effect of discretizing the target L' to multiples of η, we shall show the existence of a hypothesis $\hat{h}$ that has Lipschitz constant $\hat{L} \le \max\{\bar{L} - \eta, 0\}$ and satisfies $\tilde{R}_\eta(\hat{h}) \le \tilde{R}_\eta(\bar{h}) + \eta$. To see this, assume by translation that the minimum and maximum values assigned by $\bar{h}$ are, respectively 0 and $a \le 1$. Thus, its Lipschitz constant is $\bar{L} \ge a$ (recall we normalized $\mathrm{diam}(\mathcal{X}) = 1$). Assuming first the case $a \ge \eta$, we can set $\hat{h}(x) = (1 - \frac{\eta}{a}) \cdot \bar{h}(x)$, and it is easy to verify that its Lipschitz constant is at most $(1 - \frac{\eta}{a})\bar{L} \le \bar{L} - \eta$, and $\tilde{R}_\eta(\hat{h}) \le \tilde{R}_\eta(\bar{h}) + \eta$. The case $a < \eta$ is even easier, as now there is trivially a function $\hat{h}$ with Lipschitz constant 0 and $\tilde{R}_\eta(\hat{h}) \le \tilde{R}_\eta(\bar{h}) + \eta$. It follows that when the binary search is analyzed using this $\hat{h}$ instead of $\bar{h}$, we actually get

$$\tilde{R}_\eta(h') \le 2\tilde{R}_\eta(\hat{h}) - 24q\eta \le 2\tilde{R}_\eta(\bar{h}) - 22q\eta \le 2\tilde{R}_\eta(h^*) - 20q\eta.$$

It now remains to show that given L', program (19) may be solved quickly (within certain accuracy), which we do in Sections 3.2 and 3.3.

3.2 Solving the Linear Program

We show how to solve the linear program, given the target Lipschitz constant L'.

Fast LP-Solver Framework. To solve the linear program, we utilize the framework presented by Young [You01] for LPs of the following form: Given non-negative matrices P, C, vectors p, c and precision $\beta > 0$, find a non-negative vector x such that $Px \leq p$ and $Cx \geq c$. Young shows that if there exists a feasible solution to the input instance, then a solution to a relaxation of the input program (specifically, $Px \leq (1+\beta)p$ and $Cx \geq c$) can be found in time $O(md(\log m)/\beta^2)$, where m is the number of constraints in the program and d is the maximum number of constraints in which a single variable may appear.

In utilizing this framework for our problem, we encounter a difficulty that both the input matrices and output vector must be non-negative, while our LP (19) has difference constraints. To bypass this limitation, for each LP variable $f(x_i)$ we introduce a new variable $\tilde{x}_i$ and two new constraints:

$$\begin{aligned} f(x_i) + \tilde{x}_i &\leq 1 \\ f(x_i) + \tilde{x}_i &\geq 1 \end{aligned}$$

By the guarantees of the LP solver, we have that in the returned solution $1 - f(x_i) \leq \tilde{x}_i \leq 1 - f(x_i) + \beta$ and $\tilde{x}_i \geq 0$. This technique allows us to introduce negated variables $-f(x_i)$ into the linear program, at the loss of additive precision.

Reduced Constraints. A central difficulty in obtaining a near-linear runtime for the above linear program is that the number of constraints in LP (19) is $\Theta(n^2)$. We show how to reduce the number of constraints to near-linear in n, namely, $\eta^{-O(\mathrm{ddim}(\mathcal{X}))}n$. We will further guarantee that each of the n variables $f(x_i)$ appears in only $\eta^{-O(\mathrm{ddim}(\mathcal{X}))}$ constraints. Both these properties will prove useful for solving the program quickly.

Recall that the purpose of the $\Theta(n^2)$ constraints is solely to ensure that the target Lipschitz constant is not violated between any pair of points. We will show below that this property can be approximately maintained with many fewer constraints: The spanner described in our full paper [GKK11], has stretch $1+\delta$, degree $\delta^{-O(\mathrm{ddim}(\mathcal{X}))}$ and hop-diameter $c' \log n$ for some constant $c' > 0$, that can be computed quickly. Build this spanner for the observed sample points $\{x_i : i \in [n]\}$ with stretch $1+\eta$ (i.e., set $\delta = \eta$) and retain a constraint in LP (19) if and only if its two variables correspond to two nodes that are connected in the spanner. It follows from the bounded degree of the spanner that each variable appears in $\eta^{-O(\mathrm{ddim}(\mathcal{X}))}$ constraints, which implies that there are $\eta^{-O(\mathrm{ddim}(\mathcal{X}))}n$ total constraints.

Modifying Remaining Constraints. Each spanner-edge constraint $|f(x_i) - f(x_j)| \leq L' \cdot \rho(x_i, x_j)$ is replaced by a set of two constraints

$$\begin{aligned} f(x_i) + \tilde{x}_j &\leq 1 + L' \cdot \rho(x_i, x_j) \\ f(x_j) + \tilde{x}_i &\leq 1 + L' \cdot \rho(x_i, x_j) \end{aligned}$$

By the guarantees of the LP solver we have that in the returned solution, each spanner edge constraint will satisfy

$$\begin{aligned}|f(x_i) - f(x_j)| &\leq -1 + (1+\beta)[1 + L' \cdot \rho(x_i, x_j)] \\ &= \beta + (1+\beta)L' \cdot \rho(x_i, x_j).\end{aligned}$$

Now consider the Lipschitz condition for two points not connected by a spanner edge: Let $x_1, \ldots, x_{k+1}$ be a $(1+\eta)$-stretch $(k \leq c' \log n)$-hop spanner path connecting points $x = x_1$ and $x' = x_{k+1}$. Then the spanner stretch guarantees that

$$\begin{aligned}|f(x) - f(x')| &\leq \textstyle\sum_{i=1}^{k} [\beta + (1+\beta)L' \cdot \rho(x_i, x_{i+1})] \\ &\leq \beta c' \log n + (1+\beta)L' \cdot (1+\eta)\rho(x, x').\end{aligned}$$

Choosing $\beta = \frac{\eta^2}{24qc' \log n}$, and noting that $(1+\beta)(1+\eta) < (1+2\eta)$, we have that for all point pairs

$$|f(x) - f(x')| < \tfrac{\eta^2}{24q} + (1+2\eta)L' \cdot \rho(x, x').$$

We claim that the above inequality ensures that the computed hypothesis h' (represented by variables $f(x_i)$ above) is a 6η-perturbation of some hypothesis with Lipschitz constant L'. To prove this, first note that if $L' = 0$, then the statement follows trivially. Assume then that (by the discretization of L'), $L' \geq \eta$. Now note that a hypothesis with Lipschitz constant $(1+3\eta)L'$ is a 3η-perturbation of some hypothesis with Lipschitz constant L'. (This follows easily by scaling down this hypothesis by a factor of $(1+3\eta)$, and recalling that all values are in the range $[0,1]$.) Hence, it suffices to show that the computed hypothesis h' is a 3η-perturbation of some hypothesis $\tilde{h}$ with Lipschitz constant $(1+3\eta)L'$. We can construct $\tilde{h}$ as follows: Extract from the sample points $S = \{x_i\}_{i \in [n]}$ a (η/L')-net N, then for every net-point $z \in N$ set $\tilde{h}(z) = h'(z)$, and extend this function $\tilde{h}$ from N to all of S without increasing Lipschitz constant by using the McShane-Whitney extension theorem [McS34, Whi34] for real-valued functions. Observe that for every two net-points $z \neq z' \in N$,

$$|\tilde{h}(z) - \tilde{h}(z')| \leq \frac{\eta^2}{24q} + (1+2\eta)L' \cdot \rho(z, z') < (1+3\eta)L' \cdot \rho(z, z').$$

It follows that $\tilde{h}$ (defined on all of S) has Lipschitz constant $\tilde{L} \leq 1 + 3\eta$. Now, consider any point $x \in S$ and its closest net-point $z \in N$; then $\rho(x, z) \leq \eta/L'$. Using the fact $\tilde{h}(z) = h'(z)$, we have that $|h'(x) - \tilde{h}(x)| \leq |h'(x) - h'(z)| + |\tilde{h}(z) - \tilde{h}(x)| \leq \left[\frac{\eta^2}{24q} + (1+2\eta)L' \cdot \rho(x, z)\right] + (1+3\eta)L' \cdot \rho(x, y) \leq \frac{\eta^2}{24q} + (2+5\eta)\eta \leq 3\eta$. We conclude that h' is 3η-perturbation of $\tilde{h}$, and a 6η-perturbation of some hypothesis with Lipschitz constant L'.

Objective Function. We now turn to the objective function $\frac{1}{n}\sum_i |y_i - f(x_i)|$. We use the same technique as above for handling difference constraints: For each term $|y_i - f(x_i)|$ in the objective function we introduce the variable w_i and the constraint

$$f(x_i) + w_i \geq y_i$$

Note that the solver imposes the constraint that $w_i \geq 0$, so we have that $w_i \geq \max\{0, y_i - f(x_i)\}$. Now consider the term $f(x_i) + 2w_i$, and note that the minimum feasible value of this term in the solution of the linear program is exactly equal to $y_i + |y_i - f(x_i)|$: If $f(x_i) \geq y_i$ then the minimum feasible value of w_i is 0, which yields $f(x_i) + 2w_i = f(x_i) = y_i + (f(x_i) - y_i) = y_i + |y_i - f(x_i)|$. Otherwise we have that $f(x_i) < y_i$, so the minimum feasible value of w_i is $y_i - f(x_i)$, which yields $f(x_i) + 2w_i = 2y_i - f(x_i) = y_i + |y_i - f(x_i)|$.

The objective function is then replaced by the constraint

$$\tfrac{1}{n} \textstyle\sum_i (f(x_i) + 2w_i) \leq r,$$

which by the above discussion is equal to $\frac{1}{n} \sum_i (y_i + |y_i - f(x_i)|) \leq r$, and hence is a direct bound on the empirical error of the hypothesis. We choose bound r via binary search: Recalling that $\tilde{R}_n(h') \leq 1$ (since even a hypothesis with Lipschitz constant 0 can achieve this bound), we may set $r \leq 1$. By discretizing r in multiples of η (similar to what was done for L'), we have that the binary search will consider only $O(\log \eta^{-1})$ guesses for r. Note that for guess r', the solver guarantees only that the returned sum is less than $(1+\beta)r' \leq r' + \beta < r' + \eta$. It follows that the discretization of r and its solver relaxation of r introduce, together, at most an additive error of 2η in the LP objective, i.e., in $R_n(h')$ and in $\tilde{R}_\eta(h')$.

Correctness and Runtime Analysis. The fast LP solver ensures that h' computed by the above-described algorithm is a 6η-perturbation of a hypothesis with Lipschitz constant L'. As for $\tilde{R}(h')$, which we wanted to minimize, an additive error of 2η is incurred by comparing h' to $\bar{h}$ instead of to h^*, another additive error of 2η arises from discretizing $\bar{L}$ into L' (i.e., comparing to $\hat{h}$ instead of $\bar{h}$), and another additive error 4η introduced through the discretization of r and its solver relaxation. Overall, the algorithm above computes a hypothesis $h' \in \bigcup_{k \in \mathbb{N}} \left[\mathcal{H}^{(k)}\right]_{6\eta}$ with $\tilde{R}_\eta(h') \leq 2\tilde{R}_\eta(h^*) - 16\eta$. The parameters in Theorem 3.1 are achieved by scaling down η to $\frac{\eta}{6}$ and the simple manipulation $\tilde{R}_{\eta/6}(h) = \tilde{R}_\eta(h) - 20q\eta$.

Finally, we turn to analyze the algorithmic runtime. The spanner may be constructed in time $O(\eta^{-O(\mathrm{ddim}(\mathcal{X}))} n \log n)$. Young's LP solver [You01] is invoked $O(\log \frac{n}{\eta} \log \frac{1}{\eta})$ times, where the $\log \frac{n}{\eta}$ term is due to the binary search for L', and the $\log \frac{1}{\eta}$ term is due to the binary search for r. To determine the runtime per invocation, recall that each variable of the program appears in $d = \eta^{-O(\mathrm{ddim}(\mathcal{X}))}$ constraints, implying that there exist $m = \eta^{-O(\mathrm{ddim}(\mathcal{X}))} n$ total constraints. Since we set $\beta = O(\eta^2 / \log n)$, we have that each call to the solver takes time $O(md(\log m)/\beta^2) \leq \eta^{-O(\mathrm{ddim}(\mathcal{X}))} n \log^2 n$, for a total runtime of $\eta^{-O(\mathrm{ddim}(\mathcal{X}))} n \log^2 n \log \frac{n}{\eta} \log \frac{1}{\eta} \leq \eta^{-O(\mathrm{ddim}(\mathcal{X}))} n \log^3 n$. This completes the proof of Theorem 3.1 for $q = 1$.

3.3 Solving the Quadratic Program

Above, we considered the case when the loss function is linear. Here we modify the objective function construction to cover the case when the loss function is quadratic, that is $\frac{1}{n}\sum_i |y_i - f(x_i)|^2$. We then use the LP solver to solve our quadratic program. (Note that the spanner-edge construction above remains as before, and only the objective function construction is modified.)

Let us first redefine w_i by the constraints

$$\begin{gathered} f(x_i) + w_i \leq 1 \\ f(x_i) + w_i \geq 1 \end{gathered}$$

It follows from the guarantees of the LP solver that in the returned solution, $1 - f(x_i) \leq w_i \leq 1 - f(x_i) + \beta$ and $w_i \geq 0$.

Now note that a quadratic inequality $v \geq x^2$ can be approximated for $x \in [0,1]$ by a set of linear inequalities of the form

$$v \geq 2(j\eta)x - (j\eta)^2$$

for $0 \leq j \leq \frac{1}{\eta}$; these are just a collection of tangent lines to the quadratic function. Note that the slope of the quadratic function in the stipulated range is at most 2, so this approximation introduces an additive error of at most 2η.

Since $|y_i - f(x_i)|^2$ takes values in the range $[0,1]$, we will consider an equation set of the form

$$v_i \geq 2(j\eta)|y_i - f(x_i)| - (j\eta)^2 + 2\eta$$

which satisfies that the minimum feasible value of v_i is in the range $[|y_i - f(x_i)|^2, |y_i - f(x_i)|^2 + 2\eta]$. It remains to model these difference constraints in the LP framework: When $f(x_i) \leq y_i$, the equation set

$$v_i + 2(j\eta)f(x_i) \geq 2(j\eta)y_i - (j\eta)^2 + 2\eta$$

exactly models the above constraints. When $f(x_i) > y_i$, the lower bound of this set may not be tight, and instead the equation set

$$v_i + 2(j\eta)w_i \geq -2(j\eta)y_i - (j\eta)^2 + 2\eta + 2(j\eta)(1+\beta)$$

models the above constraints, though possibly increasing the value of v_i by $2(j\eta)\beta < \eta$. (Note that when $f(x_i) < y_i$, the lower bound of the second equation set may not be tight, so the first equation set is necessary. Also, note that whenever the right hand side of an equation is negative, the equation is vacuous and may be omitted.)

The objective function is then replaced by the inequality

$$\tfrac{1}{n}\textstyle\sum_i v_i \leq r,$$

where r is chosen by binary search as above.

Turning to the runtime analysis, the replacement of a constraint by $O(1/\eta)$ new constraints does not change the asymptotic runtime. For the analysis of the approximation error, first note that a solution to this program is a feasible solution to the original quadratic program. Further, given a solution to the original quadratic program, a feasible solution to the above program can be found by perturbing the quadratic program solution by at most 3η (since additive terms of 2η and η are lost in the above construction). The proof of Theorem 3.1 for $q = 2$ follows by an appropriate scaling of η.

4 Approximate Lipschitz Extension

In this section, we show how to evaluate our hypothesis on a new point. More precisely, given a hypothesis function $f : S \to [0,1]$, we wish to evaluate a minimum Lipschitz extension of f on a new point $x \notin S$. That is, denoting $S = \{x_1, \ldots, x_n\}$, we wish to return a value $y = f(x)$ that minimizes $\max_i\{\frac{|y-f(x_i)|}{\rho(x,x_i)}\}$. Necessarily, this value is not greater than the Lipschitz constant of the classifier, meaning that the extension of f to the new point does not increase the Lipschitz constant of f and so Theorem 2.3 holds for the single new point. (By this local regression analysis, it is not necessary for newly evaluated points to have low Lipschitz constant with respect to each other, since Theorem 2.3 holds for each point individually.)

First note that the Lipschitz extension label y of $x \notin S$ will be determined by two points of S. That is, there are two points $x_i, x_j \in S$, one with label greater than y and one with a label less than y, such that the Lipschitz constant of (x, y) relative to each of these points (that is, $L = \frac{f(x_i)-y}{\rho(x,x_i)} = \frac{y-f(x_j)}{\rho(x,x_j)}$) is maximum over the Lipschitz constant of (x, y) relative to any point in S. Hence, y cannot be increased or decreased without increasing the Lipschitz constant with respect to one of these points.

Note then that an exact Lipschitz extension may be derived in $\Theta(n^2)$ time in brute-force fashion, by enumerating all point pairs in S, calculating the optimal Lipschitz extension for x with respect to each pair alone, and then choosing the candidate value for y with the highest Lipschitz constant. However, we demonstrate that an approximate solution to the Lipschitz extension problem can be derived more efficiently.

Theorem 4.1. *An η-additive approximation to the Lipschitz extension problem can be computed in time $\eta^{-O(\mathrm{ddim}(\mathcal{X}))} \log n$.*

Proof. The algorithm is as follows: Round up all labels $f(x_i)$ to the nearest term $j\eta/2$ (for any integer $0 \le j \le 2/\eta$), and call the new label function $\tilde{f}$. We seek the value of $\tilde{f}(x)$, the optimal Lipschitz extension value for x for the new function $\tilde{f}$. Trivially, $f(x) \le \tilde{f}(x) \le f(x) + \eta/2$. Now, if we were given for each j the point with label $j\eta/2$ that is the nearest neighbor of x (among all points with this label), then we could run the brute-force algorithm described above on these $2/\eta$ points in time $O(\eta^{-2})$ and derive $\tilde{f}(x)$. However, exact metric nearest neighbor search is potentially expensive, and so we cannot find these points efficiently. We instead find for each j a point $x' \in S$ with label $\tilde{f}(x') = j\eta/2$ that is a

$(1+\frac{\eta}{2})$-approximate nearest neighbor of x among points with this label. (This can be done by presorting the points of S into $2/\eta$ buckets based on their $\tilde{f}$ label, and once x is received, running on each bucket a $(1+\frac{\eta}{2})$-approximate nearest neighbor search algorithm due to [CG06] that takes $\eta^{-O(\mathrm{ddim}(\mathcal{X}))}\log n$ time.) We then run the brute force algorithm on these $2/\eta$ points in time $O(\eta^{-2})$. The nearest neighbor search achieves approximation factor $1+\frac{\eta}{2}$, implying a similar multiplicative approximation to L, and thus also to $|y-f(x')|\leq 1$, which means at most $\eta/2$ additive error in the value y. We conclude that the algorithm's output solves the Lipschitz extension problem with additive η. □

References

[ABCH97] Alon, N., Ben-David, S., Cesa-Bianchi, N., Haussler, D.: Scale-sensitive dimensions, uniform convergence, and learnability. Journal of the ACM 44(4), 615–631 (1997)

[BBL05] Boucheron, S., Bousquet, O., Lugosi, G.: Theory of classification: A survey of recent advances. ESAIM Probab. Statist. 9, 323–375 (2005)

[BFOS84] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and regression trees. Wadsworth Statistics/Probability Series. Wadsworth Advanced Books and Software, Belmont (1984)

[BKL06] Beygelzimer, A., Kakade, S., Langford, J.: Cover trees for nearest neighbor. In: 23rd International Conference on Machine Learning, pp. 97–104. ACM (2006)

[CG06] Cole, R., Gottlieb, L.-A.: Searching dynamic point sets in spaces with bounded doubling dimension. In: 38th Annual ACM Symposium on Theory of Computing, pp. 574–583 (2006)

[Cla99] Clarkson, K.L.: Nearest neighbor queries in metric spaces. Discrete Comput. Geom. 22(1), 63–93 (1999)

[Cla06] Clarkson, K.: Nearest-neighbor searching and metric space dimensions. In: Shakhnarovich, G., Darrell, T., Indyk, P. (eds.) Nearest-Neighbor Methods for Learning and Vision: Theory and Practice, pp. 15–59. MIT Press (2006)

[DGL96] Devroye, L., Györfi, L., Lugosi, G.: A probabilistic theory of pattern recognition. Applications of Mathematics (New York), vol. 31. Springer, New York (1996)

[GK10] Gottlieb, L.-A., Krauthgamer, R.: Proximity algorithms for nearly-doubling spaces. In: Serna, M., Shaltiel, R., Jansen, K., Rolim, J. (eds.) APPROX 2010. LNCS, vol. 6302, pp. 192–204. Springer, Heidelberg (2010)

[GKK10] Gottlieb, L.-A., Kontorovich, L., Krauthgamer, R.: Efficient classification for metric data. In: COLT, pp. 433–440 (2010)

[GKK11] Gottlieb, L.-A., Kontorovich, A., Krauthgamer, R.: Efficient regression in metric spaces via approximate Lipschitz extension (2011), http://arxiv.org/abs/1111.4470

[GKK13] Gottlieb, L.-A., Kontorovich, A., Krauthgamer, R.: Adaptive metric dimensionality reduction (2013), http://arxiv.org/abs/1302.2752

[GKKW02] Györfi, L., Kohler, M., Krzyżak, A., Walk, H.: A distribution-free theory of nonparametric regression. Springer Series in Statistics. Springer, New York (2002)

[GKL03] Gupta, A., Krauthgamer, R., Lee, J.R.: Bounded geometries, fractals, and low-distortion embeddings. In: FOCS, pp. 534–543 (2003)

[HM06] Har-Peled, S., Mendel, M.: Fast construction of nets in low-dimensional metrics and their applications. SIAM Journal on Computing 35(5), 1148–1184 (2006)

[KD11] Kpotufe, S., Dasgupta, S.: A tree-based regressor that adapts to intrinsic dimension. Journal of Computer and System Sciences (2011) (to appear)

[KL04] Krauthgamer, R., Lee, J.R.: Navigating nets: Simple algorithms for proximity search. In: 15th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 791–801 (January 2004), http://dl.acm.org/citation.cfm?id=982792.982913

[Kpo09] Kpotufe, S.: Fast, smooth and adaptive regression in metric spaces. In: Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C.K.I., Culotta, A. (eds.) Advances in Neural Information Processing Systems 22, pp. 1024–1032 (2009)

[KSW09] Kleinberg, J., Slivkins, A., Wexler, T.: Triangulation and embedding using small sets of beacons. J. ACM 56, 32:1–32:37 (2009)

[LW08] Lafferty, J., Wasserman, L.: Rodeo: Sparse, greedy nonparametric regression. Ann. Stat. 36(1), 28–63 (2008)

[LZ95] Lugosi, G., Zeger, K.: Nonparametric estimation via empirical risk minimization. IEEE Transactions on Information Theory 41(3), 677–687 (1995)

[McS34] McShane, E.J.: Extension of range of functions. Bull. Amer. Math. Soc. 40(12), 837–842 (1934)

[MH04] Minh, H.Q., Hofmann, T.: Learning over compact metric spaces. In: Shawe-Taylor, J., Singer, Y. (eds.) COLT 2004. LNCS (LNAI), vol. 3120, pp. 239–254. Springer, Heidelberg (2004)

[Nad89] Nadaraya, È.A.: Nonparametric estimation of probability densities and regression curves. Mathematics and its Applications (Soviet Series), vol. 20. Kluwer Academic Publishers Group, Dordrecht (1989); Translated from the Russian by Samuel Kotz

[Ney06] Neylon, T.: Sparse solutions for linear prediction problems. PhD thesis, New York University (2006)

[Pol84] Pollard, D.: Convergence of Stochastic Processes. Springer (1984)

[SBWA98] Shawe-Taylor, J., Bartlett, P.L., Williamson, R.C., Anthony, M.: Structural risk minimization over data-dependent hierarchies. IEEE Transactions on Information Theory 44(5), 1926–1940 (1998)

[Tsy04] Tsybakov, A.B.: Introduction à l'estimation non-paramétrique. Mathématiques & Applications (Berlin), vol. 41. Springer, Berlin (2004)

[Vap95] Vapnik, V.N.: The nature of statistical learning theory. Springer, New York (1995)

[vLB04] von Luxburg, U., Bousquet, O.: Distance-based classification with Lipschitz functions. Journal of Machine Learning Research 5, 669–695 (2004)

[Was06] Wasserman, L.: All of nonparametric statistics. Springer Texts in Statistics. Springer, New York (2006)

[Whi34] Whitney, H.: Analytic extensions of differentiable functions defined in closed sets. Transactions of the American Mathematical Society 36(1), 63–89 (1934)

[You01] Young, N.E.: Sequential and parallel algorithms for mixed packing and covering. In: 42nd Annual IEEE Symposium on Foundations of Computer Science, pp. 538–546 (2001)

Data Analysis of (Non-)Metric Proximities at Linear Costs

Frank-Michael Schleif and Andrej Gisbrecht

CITEC Centre of Excellence, Bielefeld University, 33615 Bielefeld, Germany
{fschleif,agisbrec}@techfak.uni-bielefeld.de

Abstract. Domain specific (dis-)similarity or proximity measures, employed e.g. in alignment algorithms in bio-informatics, are often used to compare complex data objects and to cover domain specific data properties. Lacking an underlying vector space, data are given as pairwise (dis-)similarities. The few available methods for such data do not scale well to very large data sets. Kernel methods easily deal with *metric similarity* matrices, also at large scale, but costly transformations are necessary starting with non-metric (dis-) similarities. We propose an integrative combination of Nyström approximation, potential double centering and eigenvalue correction to obtain valid kernel matrices at linear costs. Accordingly effective kernel approaches, become accessible for these data. Evaluation at several larger (dis-)similarity data sets shows that the proposed method achieves much better runtime performance than the standard strategy while keeping competitive model accuracy. Our main contribution is an efficient linear technique, to convert (potentially non-metric) large scale *dissimilarity matrices* into approximated positive semi-definite kernel matrices.

1 Introduction

In many application areas such as bioinformatics, different technical systems, or the web, electronic data is getting larger and more complex in size and representation, using *domain specific* (dis-)similarity measures as a replacement or complement to Euclidean measures. Many classical machine learning techniques, have been proposed for Euclidean vectorial data. However, modern data are often associated to dedicated structures which make a representation in terms of Euclidean vectors difficult: biological sequence data, text files, XML data, trees, graphs, or time series [14,10,1] are of this type. These data are inherently compositional and a feature representation leads to information loss. As an alternative, a dedicated dissimilarity measure such as pairwise alignment, or kernels for structures can be used as the interface to the data. In such cases, machine learning techniques which can deal with pairwise similarities or dissimilarities have to be used [15]. Native methods for the analysis of dissimilarity data have been proposed in [15,8,7], but are widely based on non-convex optimization schemes and with quadratic to linear memory and runtime complexity, the later employing some of the approximation techniques discussed subsequently and additional heuristics.

Especially kernel methods, based on *metric similarity matrices*, revolutionized the possibility to deal with large electronic data, offering powerful tools to automatically extract regularities [19] in a convex optimization framework. But complex preprocessing steps are necessary, as discussed in the following, to apply them on non-metric

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 59–74, 2013.

(dis-) similarities. Large (dis-)similarity data are common in biology like the famous *UniProt/SwissProt*-DB with ≈ 500.000 entries or *GenBank* with ≈ 135.000 entries, but there are many more (dis-)similarity data as discussed in the work based on [15,16]. These growing data sets request effective modeling approaches. For protein and gene data recent work, proposed widely heuristically, strategies to improve the situation for large applications in unsupervised peptide retrieval [21].

Here we will show how potentially non-metric (dis-)similarities can be effectively processed by standard kernel methods with linear costs, also *in* the transformation step, which, to the authors best knowledge has not been reported before[1]. The proposed strategies permit the effective application of many kernel methods for these type of data under very mild conditions. Especially for metric dissimilarities the approach keeps the known guarantees like generalization bounds (see e.g. [3]) while for non-psd data corresponding proofs are still open, but our experiments are promising. The paper is organized as follows. First we give a short review about transformation techniques for dissimilarity data and discuss the influence of non-euclidean measures, by eigenvalue corrections. Subsequently, we discuss alternative methods for processing small dissimilarity data. We extend this discussion to approximation strategies, recalling the derivation of the low rank Nyström approximation for similarities and transfer this principle to dissimilarities. Then we link both strategies effectively to use kernel methods for the analysis of (non-)metric dissimilarity data and show the effectiveness by different exemplary supervised experiments. We also discuss differences and commons to some known approaches supported by experiments on simulated data.

2 Transformation Techniques for Dissimilarity Data

Let $\mathbf{v}_j \in \mathbb{V}$ be a set of objects defined in some data space, with $|\mathbb{V}| = N$. We assume, there exists a dissimilarity measure such that $\mathbf{D} \in \mathbb{R}^{N \times N}$ is a dissimilarity matrix measuring the pairwise dissimilarities $D_{ij} = d(\mathbf{v}_i, \mathbf{v}_j)$ between all pairs $(\mathbf{v}_i, \mathbf{v}_j) \in \mathbb{V}$. Any reasonable (possibly non-metric) distance measure is sufficient. We assume zero diagonal $d(\mathbf{v}_i, \mathbf{v}_i) = 0$ for all i and symmetry $d(\mathbf{v}_i, \mathbf{v}_j) = d(\mathbf{v}_j, \mathbf{v}_i)$ for all i, j.

2.1 Analyzing Dissimilarities by Means of Similarities for Small N

For every dissimilarity matrix $\mathbf{D}$, an associated similarity matrix $\mathbf{S}$ is induced by a process referred to as double centering with costs of $\mathcal{O}(N^2)$[15]:

$$\mathbf{S} = -\mathbf{J}\mathbf{D}\mathbf{J}/2$$
$$\mathbf{J} = (\mathbf{I} - \mathbf{1}\mathbf{1}^\top/N)$$

with identity matrix $\mathbf{I}$ and vector of ones $\mathbf{1}$. $\mathbf{D}$ is Euclidean if and only if $\mathbf{S}$ is positive semi-definite (psd). This means, we do not observe negative eigenvalues in the eigenspectrum of the matrix $\mathbf{S}$ associated to $\mathbf{D}$.

[1] Matlab code of the described transformations and test routines are available on request.

Many classification techniques have been proposed to deal with such psd kernel matrices $\mathbf{S}$ implicitly such as the support vector machine (SVM). In this case, preprocessing is *required to guarantee* psd. In [1] different strategies were analyzed to obtain valid kernel matrices for a given similarity matrix $\mathbf{S}$, most popular are: *clipping, flipping, shift correction, vector-representation*. The underlying idea is to remove negative eigenvalues in the eigenspectrum of the matrix $\mathbf{S}$.

Assuming we have a symmetric similarity matrix $\mathbf{S}$, it has an eigenvalue decomposition $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, with orthonormal matrix $\mathbf{U}$ and diagonal matrix $\mathbf{\Lambda}$ collecting the eigenvalues. In general, p eigenvectors of $\mathbf{S}$ have positive eigenvalues and q have negative eigenvalues, $(p, q, N - p - q)$ is referred to as the *signature*.

The *clip*-operation sets all negative eigenvalues to zero, the *flip*-operation takes the absolute values, the *shift*-operation increases all eigenvalues by the absolute value of the minimal eigenvalue.

The corrected matrix $\mathbf{S}^*$ is obtained as $\mathbf{S}^* = \mathbf{U}\mathbf{\Lambda}^*\mathbf{U}^\top$, with $\mathbf{\Lambda}^*$ as the modified eigenvalue matrix using one of the above operations. The obtained matrix $\mathbf{S}^*$ can now be considered as a valid kernel matrix $\mathbf{K}$.

As an alternative, data points can be treated as vectors which coefficients or variables are given by the pairwise (dis-)similarity. These vectors can be processed using standard kernels. However, this view is changing the original data representation and leads to a finite data space, limited by the number of samples.

Interestingly, some operations such as shift do not affect the location of global optima of important cost functions such as the quantization error [12], albeit the transformation can severely affect the performance of optimization algorithms [9]. The analysis in [17] indicates that for non-Euclidean dissimilarities some corrections like above may change the data representation such that information loss occurs.

A schematic view of the relations between $\mathbf{S}$ and $\mathbf{D}$ and its transformations[2] is shown in Figure 1. Here we also report the complexity of the transformations using current typical approaches. Some of the steps can be done more efficiently by known methods, but with additional constraints or in under atypical settings as discussed in the following.

2.2 Analyzing Dissimilarities by Dedicated Methods for Small N

Alternatively, techniques have been introduced which directly deal with possibly non-metric dissimilarities. Given a symmetric dissimilarity with zero diagonal, an embedding of the data in a pseudo-Euclidean vector space determined by the eigenvector decomposition of the associated matrix $\mathbf{S}$ is always possible. A symmetric bilinear form in this space is given by $\langle \mathbf{x}, \mathbf{y} \rangle_{p,q} = \mathbf{x}^\top \mathbf{I}_{p,q} \mathbf{y}$ where $\mathbf{I}_{p,q}$ is a diagonal matrix with p entries 1 and q entries -1. Taking the eigenvectors of $\mathbf{S}$ together with the square root of the absolute value of the eigenvalues, we obtain vectors $\mathbf{v}_i$ in a pseudo-Euclidean space such that $D_{ij} = \langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{v}_i - \mathbf{v}_j \rangle_{p,q}$ holds for every pair of data points. If the number of data is not limited, a generalization of this concept to Krein spaces with according decomposition is possible [15].

Vector operations can be directly transferred to the pseudo-Euclidean space, i.e. we can deal with center points (similar to k-means) as linear combinations of data in this

[2] Transformation equations are given also in the following sections.

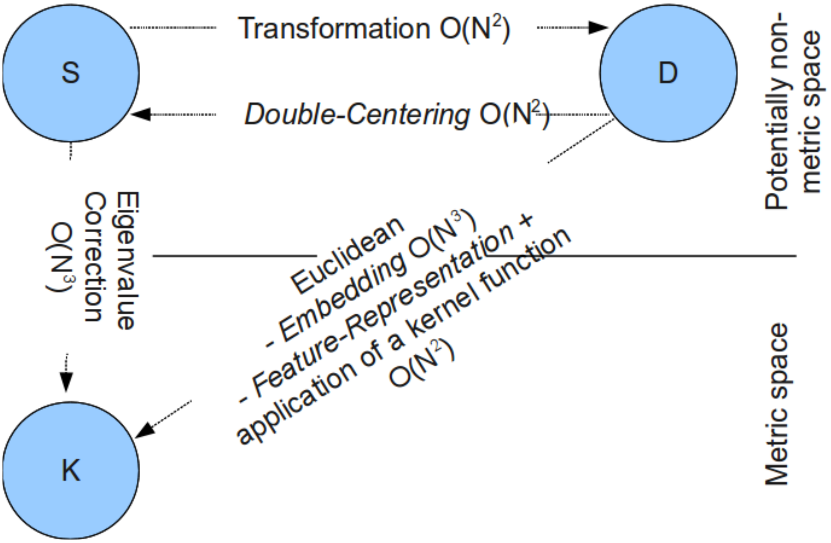

S
D
K
Transformation O(N²)
Double-Centering O(N²)
Eigenvalue Correction O(N³)
Euclidean
- Embedding O(N³)
- Feature-Representation +
application of a kernel function
O(N²)
Potentially non-metric space
Metric space

Fig. 1. Schema to illustrate the relation between similarities and dissimilarities

space. Hence we can use multiple machine learning algorithms explicitly in pseudo-Euclidean space, relying on vector operations only. One problem of this explicit transfer is given by the computational complexity of the embedding which is $\mathcal{O}(N^3)$, and, further, the fact that out-of-sample extensions to new data points characterized by pairwise dissimilarities are not immediate. An improved strategy for learning a valid relational kernel from a matrix S was recently proposed in [13], employing latent wishart processes, but this approach does not scale for larger datasets. A further strategy is to employ so called relational or proximity learning methods as discussed e.g. in [7] The underlying models consist of prototypes, which are implicitly defined as a weighted linear combination of training points: $\mathbf{w}_j = \sum_i \alpha_{ji}\mathbf{v}_i$ with $\sum_i \alpha_{ji} = 1$. But this explicit representation is not necessary because the algorithms are solely based on a specific form of distance calculations using only the matrix $\mathbf{D}$, the potentially unknown vector space V is not needed. The basic idea is an implicit computation of distances $d(\cdot,\cdot)$ during the model calculation based on the dissimilarity matrix $\mathbf{D}$ using weights α:

$$d(\mathbf{v}_i, \mathbf{w}_j) = [\mathbf{D} \cdot \alpha_j]_i - \frac{1}{2} \cdot \alpha_j^\top \mathbf{D}\alpha_j \tag{1}$$

details can be found in the aforementioned paper. As shown e.g. in [9] the mentioned methods do not rely on a metric dissimilarity matrix $\mathbf{D}$, but it is sufficient to have a symmetric $\mathbf{D}$ in a pseudo-euclidean space, with constant self-dissimilarities.

The methods discussed before are suitable for data analysis based on similarity or dissimilarity data where the number of samples N is rather small, e.g. scales by some thousand samples. For larger N only for *metric, similarity data* (valid kernels) efficient approaches have been proposed before, e.g. low-rank linearized SVM [25] or the Core-Vector Machine (CVM) [22].

In the following we discuss techniques to deal with larger sample sets for, potentially non-metric similarity and especially dissimilarity data. Especially we show how standard kernel methods can be used, assuming that for non-metric data, the necessary transformations have no severe negative influence on the data accuracy. Basically also core-set techniques become accessible for large potentially non-metric (dis-)similarity data in this way, but at the cost of multiple additional intermediate steps.

3 Nyström Approximation

The aforementioned methods depend on the similarity matrix $\mathbf{S}$ or dissimilarity matrix $\mathbf{D}$, respectively. For kernel methods and more recently for prototype based learning the usage of the Nystöm approximation is a well known technique to obtain effective learning algorithms [23,7].

3.1 Nyström Approximation for Similarities

The Nyström approximation technique has been proposed in the context of kernel methods in [23] with related proofs and bounds given in [3]. Here, we give a short review of this technique. One well known way to approximate a $N \times N$ Gram matrix, is to use a low-rank approximation. This can be done by computing the eigendecomposition of the kernel $\mathbf{K} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where $\mathbf{U}$ is a matrix, whose columns are orthonormal eigenvectors, and $\mathbf{\Lambda}$ is a diagonal matrix consisting of eigenvalues $\mathbf{\Lambda}_{11} \geq \mathbf{\Lambda}_{22} \geq ... \geq 0$, and keeping only the m eigenspaces which correspond to the m largest eigenvalues of the matrix. The approximation is $\mathbf{K} \approx \mathbf{U}_{N,m}\mathbf{\Lambda}_{m,m}\mathbf{U}_{m,N}$, where the indices refer to the size of the corresponding submatrix. The Nyström method approximates a kernel in a similar way, without computing the eigendecomposition of the whole matrix, which otherwise is an $O(N^3)$ operation.

By the Mercer theorem kernels $k(\mathbf{x}, \mathbf{y})$ can be expanded by orthonormal eigenfunctions ψ_i and non negative eigenvalues λ_i in the form

$$k(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} \lambda_i \psi_i(\mathbf{x}) \psi_i(\mathbf{y}).$$

The eigenfunctions and eigenvalues of a kernel are defined as the solution of the integral equation

$$\int k(\mathbf{y}, \mathbf{x}) \psi_i(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \lambda_i \psi_i(\mathbf{y}),$$

where $p(\mathbf{x})$ is the probability density of $\mathbf{x}$. This integral can be approximated based on the Nyström technique by sampling $\mathbf{x}^k$ i.i.d. according to $p(\mathbf{x})$:

$$\frac{1}{m} \sum_{k=1}^{m} k(\mathbf{y}, \mathbf{x}^k) \psi_i(\mathbf{x}^k) \approx \lambda_i \psi_i(\mathbf{y}).$$

Using this approximation and the matrix eigenproblem equation

$$\mathbf{K}^{(m)} \mathbf{U}^{(m)} = \mathbf{U}^{(m)} \mathbf{\Lambda}^{(m)}$$

of the corresponding $m \times m$ Gram sub-matrix $\mathbf{K}^{(m)}$ we can derive the approximations for the eigenfunctions and eigenvalues of the kernel k

$$\lambda_i \approx \frac{\lambda_i^{(m)}}{m}, \quad \psi_i(\mathbf{y}) \approx \frac{\sqrt{m}}{\lambda_i^{(m)}} \mathbf{k}_y \mathbf{u}_i^{(m)}, \tag{2}$$

where $\mathbf{u}_i^{(m)}$ is the ith column of $\mathbf{U}^{(m)}$. Thus, we can approximate ψ_i at an arbitrary point $\mathbf{y}$ as long as we know the vector $\mathbf{k}_y = (k(\mathbf{x}^1, \mathbf{y}), ..., k(\mathbf{x}^m, \mathbf{y}))^\top$.

For a given $N \times N$ Gram matrix $\mathbf{K}$ we randomly choose m rows and respective columns. The corresponding indices's are also called landmarks, and should be chosen such that the data distribution is sufficiently covered. A specific analysis about selection strategies was recently discussed in [24]. We denote these rows by $\mathbf{K}_{m,N}$. Using the formulas (2) we obtain $\tilde{\mathbf{K}} = \sum_{i=1}^m 1/\lambda_i^{(m)} \cdot \mathbf{K}_{m,N}^\top \mathbf{u}_i^{(m)} (\mathbf{u}_i^{(m)})^\top \mathbf{K}_{m,N}$, where $\lambda_i^{(m)}$ and $\mathbf{u}_i^{(m)}$ correspond to the $m \times m$ eigenproblem. Thus we get, $\mathbf{K}_{m,m}^{-1}$ denoting the Moore-Penrose pseudoinverse, an approximation of $\mathbf{K}$ as

$$\tilde{\mathbf{K}} = \mathbf{K}_{m,N}^\top \mathbf{K}_{m,m}^{-1} \mathbf{K}_{m,N}.$$

This approximation is exact, if $\mathbf{K}_{m,m}$ has the same rank as $\mathbf{K}$.

3.2 Nyström Approximation for Dissimilarity Data

For dissimilarity data, a direct transfer is possible, see [7] for earlier work on this topic. Earlier work in this line, but not equivalent, also appeared in the work around Landmark Multi-Dimensional-Scaling (LMDS) [20] which we address in the next section. According to the spectral theorem, a symmetric dissimilarity matrix $\mathbf{D}$ can be diagonalized $\mathbf{D} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ with $\mathbf{U}$ being a unitary matrix whose column vectors are the orthonormal eigenvectors of $\mathbf{D}$ and $\mathbf{\Lambda}$ a diagonal matrix with the corresponding eigenvalues of $\mathbf{D}$, Therefore the dissimilarity matrix can be seen as an operator

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} \lambda_i \psi_i(\mathbf{x}) \psi_i(\mathbf{y})$$

where $\lambda_i \in \mathbb{R}$ correspond to the diagonal elements of $\mathbf{\Lambda}$ and ψ_i denote the eigenfunctions. The only difference to an expansion of a kernel is that the eigenvalues can be negative. All further mathematical manipulations can be applied in the same way and we can write in an analogy to the equation 3.1

$$\hat{\mathbf{D}} = \mathbf{D}_{N,m} \mathbf{D}_{m,m}^{-1} \mathbf{D}_{N,m}^\top.$$

It allows to approximate dissimilarities between a point $\mathbf{w}^k$ represented by a coefficient vector α_k and a data point $\mathbf{x}^i$, as discussed within Eq (1), in the way

$$\begin{aligned} d(\mathbf{x}^i, \mathbf{w}^k) \approx & \left[\mathbf{D}_{m,N}^\top \left(\mathbf{D}_{m,m}^{-1} \left(\mathbf{D}_{m,N} \boldsymbol{\alpha}_k\right)\right)\right]_i \\ & -\frac{1}{2} \cdot \left(\boldsymbol{\alpha}_k^\top \mathbf{D}_{m,N}^\top\right) \cdot \\ & \left(\mathbf{D}_{m,m}^{-1} \left(\mathbf{D}_{m,N} \boldsymbol{\alpha}_k\right)\right) \end{aligned}$$

with a linear submatrix of m rows and a low rank matrix $\mathbf{D}_{m,m}$. Performing these matrix multiplications from right to left, this computation is $\mathcal{O}(m^2N)$ instead of $\mathcal{O}(N^2)$, i.e. it is linear in the number of data points N, assuming fixed approximation m.

A benefit of the Nyström technique is that it can be decided priorly which linear parts of the dissimilarity matrix will be used in training. Therefore, it is sufficient to *compute only a linear part of the full dissimilarity matrix* $\mathbf{D}$ to use these methods. A drawback of the Nyström approximation is that a good approximation can only be achieved if the rank of $\mathbf{D}$ is kept as much as possible, i.e. the chosen subset should be representative. The specific selection of the m landmark points has been recently analyzed in [24]. It was found that best results can be obtained by choosing the potential cluster centers of the data distribution as landmarks, rather a random subset, to be able to keep m smallest at lowest representation error. However the determination of these centers can become complicated for large data sets, since it can be obviously not be based on a Nyström approximated set. However the effect is not such severe as long as m is not too small.

4 Transformations of (Dis-)Similarities with Linear Costs

For *metric* similarity data, kernel methods can be applied directly, or in case of large N, the Nyström approximation can be used. We will discuss *non*-metric data later and focus now on metric or almost metric *dissimilarity* data $\mathbf{D}$.

4.1 Transformation of Dissimilarities to Similarities

As pointed out before current methods for large dissimilarity matrix $\mathbf{D}$ are non-convex approaches. On the other hand multiple effective convex kernel methods are available for metric similarity data using a matrix $\mathbf{S} = \mathbf{K}$ which we will now make accessible for matrices $\mathbf{D}$ in an effective manner. This requests for a transformation of the matrix $\mathbf{D}$ to $\mathbf{S}$ using double-centering as discussed above. This transformation contains a summation over the whole matrix and thus has quadratic complexity, which would be prohibitive for larger data sets.

One way to achieve this transformation in linear time, is to use landmark multidimensional scaling (LMDS) [20] which was shown to be a Nyström technique as well [18]. The idea is to sample a small amount m of points, called landmarks, compute the corresponding dissimilarity matrix, apply double centering on this matrix and finally project the data to a low dimensional space using eigenvalue decomposition. The remaining points can then be projected into the same space, taking into account the distances to the landmarks, and applying triangulation. Having vectorial representation of the data, it is then easy to retrieve the similarity matrix as a scalar product between the points.

Another possibility arises if we take into account our key observation, that we can combine both transformations, double centering and Nyström approximation, and make use of their linearity. Instead of applying double centering, followed by the Nyström approximation we first approximate the matrix $\mathbf{D}$ and then transform it by double centering, which yields the approximated similarity matrix $\hat{\mathbf{S}}$.

Both approaches have the costs of $\mathcal{O}(m^2N)$ and produce the same results, up to shift and rotation. This is because LMDS, in contrast to our approach, makes double centering only on a small part of $\mathbf{D}$, and thus is unable to detect the mean and the primary components of the whole data set. This can result in an unreliable impact, since similarities which are not centered might lead to an inferior performance of the algorithms and, thus, our approach should be used instead[3]. Additionally LMDS implicitly assumes that the dissimilarities are metric, respectively the negative eigenvalues of the corresponding similarity matrix are automatically clipped. This can have a negative impact on the data analysis as we show in a synthetic example in the following. Further LMDS is proposed as a projection technique leading to a low-dimensional, typically $2-3$ dimensional embedding of the data. Higher dimensional embeddings by LMDS are possible (limited by the number of positive eigenvalues), but to our best knowledge neither used nor discussed so far. A Nyström approximated kernel, avoiding the calculations of all dissimilarities, as shown in the following is not directly obtained but only after embedding of the corresponding dissimilarities and subsequent calculation of the inner products. But for this kernel the negative eigenvalues are always clipped which can have a negative impact on the analysis. Accordingly, the connection of LMDS to our approach is rather weak[4], which will get more obvious in the following derivations.

As mentioned before double centering of a matrix $\mathbf{D}$ is defined as:

$$\mathbf{S} = -\mathbf{JDJ}/2$$

where $\mathbf{J} = (\mathbf{I} - \mathbf{1}\mathbf{1}^\top/N)$ with identity matrix $\mathbf{I}$ and vector of ones $\mathbf{1}$. $\mathbf{S}$ is positive semi-definite (psd) if and only if D is Euclidean.

Lets start with a dissimilarity matrix D where we apply double centering, subsequently we approximate the obtained $\mathbf{S}$ by integrating the Nyström approximation to the matrix $\mathbf{D}$.

$$\begin{aligned}
\mathbf{S} &= -\frac{1}{2}\mathbf{JDJ}\\
&= -\frac{1}{2}\left(\left(\mathbf{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^\top\right)\mathbf{D}\left(\mathbf{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^\top\right)\right)\\
&= -\frac{1}{2}\left(\mathbf{IDI}-\frac{1}{N}\mathbf{1}\mathbf{1}^\top\mathbf{DI}-\mathbf{ID}\frac{1}{N}\mathbf{1}\mathbf{1}^\top+\frac{1}{N}\mathbf{1}\mathbf{1}^\top\mathbf{D}\frac{1}{N}\mathbf{1}\mathbf{1}^\top\right)\\
&= -\frac{1}{2}\left(\mathbf{D}-\frac{1}{N}\mathbf{D}\mathbf{1}\mathbf{1}^\top-\frac{1}{N}\mathbf{1}\mathbf{1}^\top\mathbf{D}+\frac{1}{N^2}\mathbf{1}\mathbf{1}^\top\mathbf{D}\mathbf{1}\mathbf{1}^\top\right)
\end{aligned}$$

[3] For domain specific dissimilarity measures and non-vectorial data as discussed here, it is, under practical conditions, hard to ensure that the underlying, implicit space is normalized to N(0,1), this is getting even more complicated if the measure is non-metric.

[4] Although LMDS can be adapted to provide similar results, with the exception that the small inner matrix is calculated differently with the pre-discussed influence on unnormalized data.

$$\mathbf{S} \overset{Ny}{\approx} \hat{\mathbf{S}} = -\frac{1}{2}\Bigg[\mathbf{D}_{N,m} \cdot \mathbf{D}_{m,m}^{-1} \cdot \mathbf{D}_{m,N} - \frac{1}{N}\mathbf{D}_{N,m} \\ \cdot(\mathbf{D}_{m,m}^{-1} \cdot (\mathbf{D}_{m,N}\mathbf{1}))\mathbf{1}^\top - \frac{1}{N}\mathbf{1}((\mathbf{1}^\top\mathbf{D}_{N,m}) \cdot \mathbf{D}_{m,m}^{-1}) \\ \cdot\mathbf{D}_{m,N} + \frac{1}{N^2}\mathbf{1}((\mathbf{1}^\top\mathbf{D}_{N,m}) \cdot \mathbf{D}_{m,m}^{-1} \cdot (\mathbf{D}_{m,N}\mathbf{1}))\mathbf{1}^\top\Bigg] \tag{3}$$

This equation can be rewritten for each entry of the matrix $\hat{\mathbf{S}}$

$$\hat{S}_{ij} = -\frac{1}{2}\Bigg[\mathbf{D}_{i,m} \cdot \mathbf{D}_{m,m}^{-1} \cdot \mathbf{D}_{m,j} - \frac{1}{N}\sum_k \mathbf{D}_{k,m} \cdot \mathbf{D}_{m,m}^{-1} \cdot \mathbf{D}_{m,j} \\ -\frac{1}{N}\sum_k \mathbf{D}_{i,m} \cdot \mathbf{D}_{m,m}^{-1} \cdot \mathbf{D}_{m,k} \\ +\frac{1}{N^2}\sum_{kl} \mathbf{D}_{k,m} \cdot \mathbf{D}_{m,m}^{-1} \cdot \mathbf{D}_{m,l}\Bigg],$$

as well as for the sub-matrices $\hat{\mathbf{S}}_{m,m}$ and $\hat{\mathbf{S}}_{N,m}$, in which we are interested for the Nyström approximation

$$\hat{\mathbf{S}}_{m,m} = -\frac{1}{2}\Bigg[\mathbf{D}_{m,m} - \frac{1}{N}\mathbf{1} \cdot \sum_k \mathbf{D}_{k,m} \\ -\frac{1}{N}\sum_k \mathbf{D}_{m,k} \cdot \mathbf{1}^\top \\ +\frac{1}{N^2}\mathbf{1} \cdot \sum_{kl} \mathbf{D}_{k,m} \cdot \mathbf{D}_{m,m}^{-1} \cdot \mathbf{D}_{m,l} \cdot \mathbf{1}^\top\Bigg]$$

$$\hat{\mathbf{S}}_{N,m} = -\frac{1}{2}\Bigg[\mathbf{D}_{N,m} - \frac{1}{N}\mathbf{1} \cdot \sum_k \mathbf{D}_{k,m} \\ -\frac{1}{N}\sum_k \mathbf{D}_{N,m} \cdot \mathbf{D}_{m,m}^{-1} \cdot \mathbf{D}_{m,k} \cdot \mathbf{1}^\top \\ +\frac{1}{N^2}\mathbf{1} \cdot \sum_{kl} \mathbf{D}_{k,m} \cdot \mathbf{D}_{m,m}^{-1} \cdot \mathbf{D}_{m,l} \cdot \mathbf{1}^\top\Bigg].$$

It should be noted that $\hat{\mathbf{S}}$ is only a valid kernel if $\hat{\mathbf{D}}$ is metric. The information loss obtained by the approximation is 0 if m corresponds to the rank of $\mathbf{S}$ and increases for smaller m.

4.2 Non-metric (Dis-)Similarities

In case of a non-metric $\mathbf{D}$ the transformation shown in equation 3 can still be used, but the obtained matrix $\hat{\mathbf{S}}$ is not a valid kernel. A strategy to obtain a valid kernel matrix $\hat{\mathbf{S}}$ is to apply an eigenvalue correction as discussed above. This however can be prohibitive for large matrices, since to correct the whole eigenvalue spectrum, the whole eigenvalue decomposition is needed, which has $\mathcal{O}(N^3)$ complexity. The Nyström approximation can again decrease computational costs dramatically. Since we now can apply the approximation on an arbitrary symmetric matrix, we can make the correction afterward. To correct an already approximated similarity matrix $\hat{\mathbf{S}}$ it is sufficient to correct the eigenvalues of $\mathbf{S}_{m,m}$. Altogether we get $\mathcal{O}(m^2N)$ complexity.

We can write for the approximated matrix $\hat{\mathbf{S}}$ its eigenvalue decomposition as

$$\hat{\mathbf{S}} = \mathbf{S}_{N,m}\mathbf{S}_{m,m}^{-1}\mathbf{S}_{N,m}^{\top} = \mathbf{S}_{N,m}\mathbf{U}\boldsymbol{\Lambda}^{-1}\mathbf{U}^{\top}\mathbf{S}_{N,m}^{\top},$$

where we can correct the eigenvalues $\boldsymbol{\Lambda}$ by some technique as discussed in section 2.1 to $\boldsymbol{\Lambda}^*$. The corrected approximated matrix $\hat{\mathbf{S}}^*$ is then simply

$$\hat{\mathbf{S}}^* = \mathbf{S}_{N,m}\mathbf{U}\left(\boldsymbol{\Lambda}^*\right)^{-1}\mathbf{U}^{\top}\mathbf{S}_{N,m}^{\top}. \quad (4)$$

This approach can also be used to correct dissimilarity matrices $\mathbf{D}$ by first approximating them, converting to similarities $\hat{\mathbf{S}}$ using equation 3 and then correcting the similarities. If it is desirable to work with the corrected dissimilarities, then we should note, that it is possible to transform the similarity matrix $\mathbf{S}$ to a dissimilarity matrix $\mathbf{D}$: $D_{ij}^2 = S_{ii} + S_{jj} - 2S_{ij}$. This obviously applies as well to the approximated and corrected matrices $\hat{\mathbf{S}}^*$ and $\hat{\mathbf{D}}^*$ and we get by substitution:

$$\hat{\mathbf{D}}^* = \mathbf{D}_{N,m}^*\left(\mathbf{D}_{m,m}^*\right)^{-1}\mathbf{D}_{N,m}^{*\top}. \quad (5)$$

Usually the algorithms are learned on a so called training set and we expect them to perform well on the new unseen data, or the test set. In such cases we need to provide an out of sample extension, i.e. a way to compute the algorithm on the new data. This might be a problem for the techniques dealing with (dis)similarities. If the matrices are corrected, we need to correct the new (dis)similarities as well to get consistent results. Fortunately, it is quite easy in the Nyström framework. By examining the equations 4 and 5 we see, that we simply need to extend the matrices $\mathbf{D}_{N,m}$ or $\mathbf{S}_{N,m}$, respectively, by uncorrected (dis)similarities between the new points and the landmarks to obtain the full approximated and *corrected* (dis)similarity matrices, which then can be used by the algorithms to compute the out of sample extension.

In [1] a similar approach is taken. First, the whole similarity matrix is corrected by means of a projection matrix. Then this projection matrix is applied to the new data, so that the corrected similarity between old and new data can be computed. This technique is in fact the Nyström approximation, where the whole similarity matrix $\mathbf{S}$ is treated as the approximation matrix $\mathbf{S}_{m,m}$ and the old data, together with the new data build the matrix $\mathbf{S}_{N,m}$. Rewriting this in the Nyström framework makes it clear and more obvious, without the need to compute the projection matrix and with an additional possibility to compute the similarities between the new points. In Figure 2 we depict

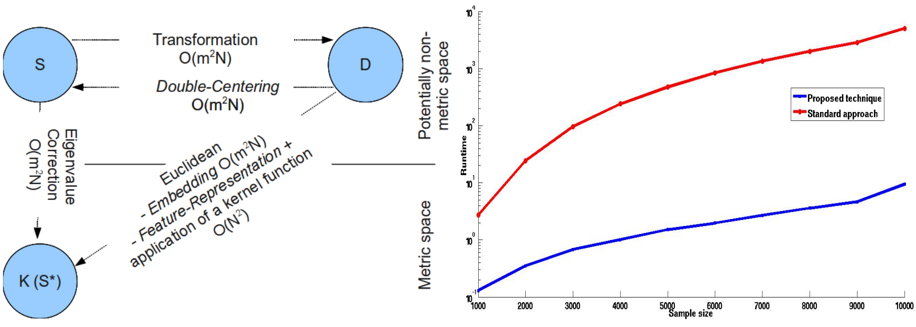


Fig. 2. Left: Updated schema from Figure 1 using the discussed approximation. The costs are now substantially smaller $m \ll N$. Right: Runtime in seconds at log-scale for the SwissProt-Runtime experiment. The standard approach is two magnitudes slower than the proposed technique.

schematically the new situation for similarity and dissimilarity data incorporating the proposed approach.

As a last point it should be mentioned that corrections like flipping, clipping or others are still under discussion and not always optimal [15]. Additionally the selection of landmark points can be complicated as discussed in [24]. Further for very large data sets (e.g. some 100 million points) the Nyström approximation may still be too costly and some other strategies have to be found.

We close this section by a small experiment on the ball dataset as proposed in [5]. It is an artificial dataset based on the surface distances of randomly positioned balls of two classes having a slightly different radius. The dataset is non-euclidean with substantial information encoded in the negative part of the eigenspectrum. We generated the data with 100 samples per class leading to a dissimilarity matrix $D = N \times N$, with $N = 200$. Now the data have been processed in four different ways to obtain a valid kernel matrix S. First we converted D into a valid kernel matrix by a full eigenvalue decomposition, followed by flipping of the negative eigenvalues and a reconstruction of the similarity matrix $K = S$, denoted as $SIM1$. This approach has a complexity of $\mathcal{O}(N^3)$. Further we generated an approximated similarity matrix $\hat{S}$ by using the proposed approach, flipping in the eigenvalue correction and 10 landmarks for the Nyström approximation. This dataset is denoted as $SIM2$ and was obtained with a complexity of $\mathcal{O}(m^2N)$. The third dataset $SIM3$ was obtained in the same way but the eigenvalues were clipped. The dataset $SIM4$ was obtained using landmark MDS with the same landmarks as for $SIM2$ and $SIM3$. The data are processed by a Support Vector Machine in a 10-fold crossvalidation results on the test sets are shown in Table 1. As mentioned the data

Table 1. Test set results of a 10-fold SVM run on the ball dataset using the different encodings

	$SIM1$	$SIM2$	$SIM3$	$SIM4$
Test-Accuracy	100 ± 0	87.00 ± 7.53	68.00 ± 6.32	52.00 ± 11.83

contain substantial information in the negative fraction of the eigenspectrum, accordingly one may expect that this eigenvalues should not be removed. This is also reflected in the results. LMDS removed the negative eigenvalues and the classification model based on this data shows random prediction accuracy. The SIM3 encoding is slightly better. Also in this case the negative eigenvalues are removed but the limited amount of class separation information, encoded in the positive fraction was better preserved, probably due to the different calculation of the matrix $\hat{S}_{mm}$. The SIM2 data used the flipping strategy and shows already quite good prediction accuracy, taking into account that the kernel matrix is only approximated by 10 landmarks and the relevant (original negative) eigenvalues are of small magnitude.

5 Experiments

We now apply the priorly derived approach to three non-metric dissimilarity and similarity data and show the effectiveness for a classification task. The considered data are (1) the SwissProt similarity data as described in [10] (DS1, 10988 samples, 30 classes, imbalanced, signature: $[8488, 2500, 0]$) (2) the chromosome dissimilarity data taken from [14] (DS2, 4200 samples, 21 classes, balanced, signature: $[2258, 1899, 43]$) and the proteom dissimilarity data set [4] (DS3, 2604 samples, 53 classes, imbalanced, signature: $[1502, 682, 420]$). All datasets are non-metric, multiclass and contain multiple thousand objects, such that a regular eigenvalue correction with a prior double-centering for dissimilarity data, as discussed before, is already very costly. The data are analyzed in two ways, employing either the flipping strategy as an eigenvalue correction, or by not-correcting the eigenvalues[5]. To be effective for the large number of object we also apply the Nyström approximation as discussed before using a sample rate of $1\%, 10\%, 30\%$[6], by selecting random landmarks from the data. Other sampling strategies have been discussed in [24,6], also the impact of the Nyström approximation with respect to kernel methods has been discussed recently in [2], but this is out of the focus of this paper.

To get comparable experiments, the same randomly drawn landmarks are used in each of the corresponding sub-experiments (along a column in the table). New landmarks are only drawn for different Nyström approximations and sample sizes like in Figure 3. Classification rates are calculated in a 10-fold crossvalidation using the Core-Vector-Machine (CVM) and the Support-Vector-Machine (SVM) (see [22,19]). The crossvalidation does not include a new draw of the landmarks, to cancel out the selection bias of the Nyström approximation, accordingly SVM and CVM use the same kernel matrices. However, our objective is not maximum classification performance (which is only one possible application) but to demonstrate the effectiveness of our approach for dissimilarity data of larger scale. The classification results are summarized

[5] Clipping and flipping were found similar effective, with a little advance for flipping. With flipping the information of the negative-eigenvalues is at least somewhat kept in the data representation so we focus on this representation. Shift correction was found to have a negative impact on the model as already discussed in [1].

[6] A larger sample size did not lead to further substantial improvements in the results.

Table 2. Average test set accuracy for SwissProt (DS1), Chromosome (DS2), Proteom (DS3) using a Nyström approximation of 1% and 10% and no or flip eigenvalue correction. Kernel matrices have been Nyström approximated either, as proposed during the eigenvalue correction, or later on, like in the standard approach. The signatures are based on the approximated kernel matrices.

	$DS1_{1\%}$	$DS2_{1\%}$	$DS3_{1\%}$	$DS1_{10\%}$	$DS2_{10\%}$	$DS3_{10\%}$
Signature	[109,1,10878]	[41,1,4158]	[25,1,2492]	[1078,19,9891]	[296,123,3781]	[235,10,2273]
CVM-No	92.81 ± 0.74	94.64 ± 0.88	64.42 ± 2.89	75.53 ± 0.90	40.43 ± 2.12	23.95 ± 2.4
SVM-No	92.82 ± 0.90	94.24 ± 1.00	45.59 ± 3.01	82.92 ± 2.00	47.21 ± 2.42	27.56 ± 2.93
Signature	[110,0,10878]	[42,0,4158]	[26,0,2492]	[1097,0,9891]	[419,0,3781]	[245,0,2273]
CVM-Flip	92.78 ± 0.74	94.62 ± 0.85	91.62 ± 1.57	97.01 ± 0.54	96.98 ± 0.77	96.98 ± 1.28
SVM-Flip	93.02 ± 0.70	94.31 ± 1.37	93.65 ± 1.52	97.56 ± 0.51	96.98 ± 0.88	97.34 ± 0.73

Table 3. Average test set accuracy for SwissProt (DS1), Chromosome (DS2), Proteom (DS3) using a Nyström approximation of 30% and no or flip eigenvalue correction. Kernel matrices have been Nyström approximated (with $L = 30\% \cdot N$) either, as proposed during the eigenvalue correction, or later on, like in the standard approach. The signatures are based on the approximated kernel matrices.

	DS1		DS2		DS3	
Signature	[2995,300,7693]		[759,493,2948]		[577,118,1823]	
CVM-No	72.14 ± 2.01		60.24 ± 3.12		56.75 ± 2.56	
SVM-No	77.01 ± 3.03		66.36 ± 2.94		49.21 ± 2.51	
Signature	[3295,0,7693]		[1252,0,2948]		[695,0,1823]	
CVM-Flip	96.85 ± 0.53		96.90 ± 0.66		99.17 ± 0.28	
SVM-Flip	97.49 ± 0.36		96.98 ± 0.45		98.85 ± 0.78	

in Table 2-3 for the different Nyström approximations 1%, 10% and 30%. First one observes that the eigenvalue correction has a strong, positive effect on the classification performance consistent with earlier findings [1]. However in case of a small number of landmarks the effect of the eigenvalue correction is less pronounced compared to the uncorrected experiment as shown in Table 2 for DS1 and DS2. In these cases the Nyström approximation has also reduced the number of non-negative eigenvalues, as shown by the corresponding signatures, such that an implicit eigenvalue correction is obtained. For DS3 the remaining eigenvector has a rather high magnitude and a strong impact accordingly, such that the classification performance is sub-optimal for the uncorrected experiment. Raising the number of landmarks Table 2-3 also the classification performance improves for the experiments with eigenvalue correction. The experiments without eigenvalue correction show however a degeneration in the performance, because more and more negative eigenvalues are still kept by the Nyström approximation as shown in the signatures[7].

[7] Comparing signatures at different Nyström approximations also shows that many eigenvalues are close to zero and are sometimes counted as positive,negative or zero.

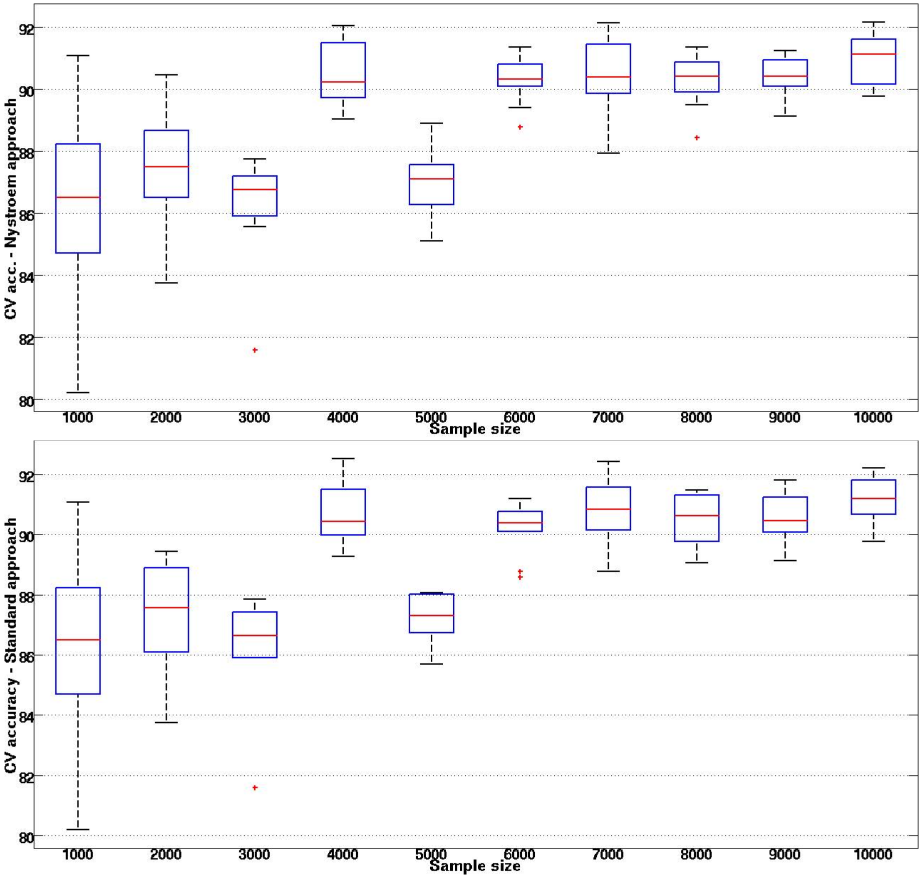


Fig. 3. Top: box-plots of the classification performance for different sample sizes of DS1 using the proposed approach with 100 landmarks. Bottom: The same experiment but with the standard approach. Obviously our approach does not sacrifice performance for computational speed.

As shown exemplary in Figure 3 the classification performance on eigenvalue-corrected data is approximately the same using our proposed strategy or the standard technique, but the runtime performance (right plot in Figure 2) is drastically better for an increase in the number of samples. To show this we selected subsets from the SwissProt data with different sizes from 1000 to 10000 points and calculated the runtime and classification performance using the CVM classifier in a 10-fold crossvalidation, with a fixed Nyström approximation of $L = 100$ and a flipping eigenvalue correction. The results of the proposed approach are shown in the left box-plots of Figure 3 and the results for the standard technique are shown in the right plot. The corresponding runtimes are shown in Figure 3, with the runtime of our approach as the curve on the bottom and the runtime of the standard method on the top, two magnitudes larger on log-scale.

6 Outlook and Conclusions

In this paper we discussed the relation between similarity and dissimilarity data and effective ways to move across the different representations in a systematic way. Using the presented approach, effective and *accurate* transformations are possible. Kernel approaches but also dissimilarity learners are now accessible for both types of data. While the parametrization of the Nyström approximation is already studied in [11,24] there are still different open issues. In future work we will analyze more deeply the handling of extremely large (dis-)similarity sets and transfer our approach to unsupervised problems. While the proposed strategy was found to be very effective e.g. to improve supervised learning of non-metric dissimilarities by kernel methods, it is however also limited again by the Nyström approximation, which may fail to provide sufficient approximation. Accordingly it is still very interesting to provide dedicated methods for such data as argued in [17]. For non-psd data the error introduced by the Nyström approximation is not yet fully understood and bounds similar as proposed in [3] are still an open issue. In our experiments we observed that flipping was an effective approach to keep the relevant structure of the data but this are only heuristic findings and not yet completely understood, we will address this in future work.

Acknowledgments. We would like to thank the *Max-Planck-Institute for Physics of Complex Systems in Dresden* and Michael Biehl, Thomas Villmann and Manfred Opper as the organizer of the *Statistical Inference: Models in Physics and Learning*-Workshop for providing a nice working atmosphere during the preparation of this manuscript. Frank-Michael Schleif was supported by the "German Science Foundation (DFG)" under grant number HA-2719/4-1. Financial support from the Cluster of Excellence 277 Cognitive Interaction Technology funded by the German Excellence Initiative is gratefully acknowledged. Further we would like to thank Fabrice Rossi for helpful discussions and suggestions.

References

[1] Chen, Y., Garcia, E.K., Gupta, M.R., Rahimi, A., Cazzanti, L.: Similarity-based classification: Concepts and algorithms. JMLR 10, 747–776 (2009)

[2] Cortes, C., Mohri, M., Talwalkar, A.: On the impact of kernel approximation on learning accuracy. JMLR - Proceedings Track 9, 113–120 (2010)

[3] Drineas, P., Mahoney, M.W.: On the nyström method for approximating a gram matrix for improved kernel-based learning. Journal of Machine Learning Research 6, 2153–2175 (2005)

[4] Duin, R.P.: PRTools (March 2012), `http://www.prtools.org`

[5] Duin, R.P.W., Pękalska, E.: Non-euclidean dissimilarities: Causes and informativeness. In: Hancock, E.R., Wilson, R.C., Windeatt, T., Ulusoy, I., Escolano, F. (eds.) SSPR&SPR 2010. LNCS, vol. 6218, pp. 324–333. Springer, Heidelberg (2010)

[6] Farahat, A.K., Ghodsi, A., Kamel, M.S.: A novel greedy algorithm for nyström approximation. JMLR - Proceedings Track 15, 269–277 (2011)

[7] Gisbrecht, A., Mokbel, B., Schleif, F.M., Zhu, X., Hammer, B.: Linear time relational prototype based learning. Journal of Neural Systems 22(5) (2012)

[8] Graepel, T., Obermayer, K.: A stochastic self-organizing map for proximity data. Neural Computation 11(1), 139–155 (1999)

[9] Hammer, B., Hasenfuss, A.: Topographic mapping of large dissimilarity data sets. Neural Computation 22(9), 2229–2284 (2010)

[10] Kohonen, T., Somervuo, P.: How to make large self-organizing maps for nonvectorial data. Neural Networks 15(8-9), 945–952 (2002)

[11] Kumar, S., Mohri, M., Talwalkar, A.: On sampling-based approximate spectral decomposition. In: ICML. ACM International Conference Proceeding Series, vol. 382, p. 70. ACM (2009)

[12] Laub, J., Roth, V., Buhmann, J.M., Müller, K.R.: On the information and representation of non-euclidean pairwise data. Pattern Recognition 39(10), 1815–1826 (2006)

[13] Li, W.J., Zhang, Z., Yeung, D.Y.: Latent wishart processes for relational kernel learning. JMLR - Proceedings Track 5, 336–343 (2009)

[14] Neuhaus, M., Bunke, H.: Edit distance based kernel functions for structural pattern classification. Pattern Recognition 39(10), 1852–1863 (2006)

[15] Pekalska, E., Duin, R.: The dissimilarity representation for pattern recognition. World Scientific (2005)

[16] Pekalska, E., Duin, R.P.W.: Beyond traditional kernels: Classification in two dissimilarity-based representation spaces. IEEE Transactions on Systems, Man, and Cybernetics Part C 38(6), 729–744 (2008)

[17] Pękalska, E.z., Duin, R.P.W., Günter, S., Bunke, H.: On not making dissimilarities euclidean. In: Fred, A., Caelli, T.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.) SSPR&SPR 2004. LNCS, vol. 3138, pp. 1145–1154. Springer, Heidelberg (2004)

[18] Platt, J.: Fastmap, metricmap, and landmark mds are all nyström algorithms (2005)

[19] Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis and Discovery. Cambridge University Press (2004)

[20] de Silva, V., Tenenbaum, J.B.: Global versus local methods in nonlinear dimensionality reduction. In: NIPS, pp. 705–712. MIT Press (2002)

[21] Tan, J., Kuchibhatla, D., Sirota, F.L.: Tachyon search speeds up retrieval of similar sequences by several orders of magnitude. Bio Informatics (April 23, 2012)

[22] Tsang, I.W., Kocsor, A., Kwok, J.T.: Simpler core vector machines with enclosing balls. In: ICML. ACM International Conference Proceeding Series, vol. 227, pp. 911–918. ACM (2007)

[23] Williams, C.K.I., Seeger, M.: Using the nyström method to speed up kernel machines. In: NIPS, pp. 682–688. MIT Press (2000)

[24] Zhang, K., Kwok, J.T.: Clustered nyström method for large scale manifold learning and dimension reduction. IEEE Transactions on Neural Networks 21(10), 1576–1587 (2010)

[25] Zhang, K., Lan, L., Wang, Z., Moerchen, F.: Scaling up kernel svm on limited resources: A low-rank linearization approach. JMLR - Proceedings Track 22, 1425–1434 (2012)

On the Informativeness of Asymmetric Dissimilarities

Yenisel Plasencia-Calaña[1,2], Veronika Cheplygina[2], Robert P.W. Duin[2],
Edel B. García-Reyes[1], Mauricio Orozco-Alzate[3], David M.J. Tax[2],
and Marco Loog[2]

[1] Advanced Technologies Application Center, La Habana, Cuba
{yplasencia,egarcia}@cenatav.co.cu
[2] Pattern Recognition Laboratory, Delft University of Technology, The Netherlands
{v.cheplygina,D.M.J.Tax,m.loog}@tudelft.nl, r.duin@ieee.org
[3] Departamento de Informática y Computación,
Universidad Nacional de Colombia - Sede Manizales, Colombia
morozcoa@unal.edu.co

Abstract. A widely used approach to cope with asymmetry in dissimilarities is by symmetrizing them. Usually, asymmetry is corrected by applying combiners such as average, minimum or maximum of the two directed dissimilarities. Whether or not these are the best approaches for combining the asymmetry remains an open issue. In this paper we study the performance of the extended asymmetric dissimilarity space (EADS) as an alternative to represent asymmetric dissimilarities for classification purposes. We show that EADS outperforms the representations found from the two directed dissimilarities as well as those created by the combiners under consideration in several cases. This holds specially for small numbers of prototypes; however, for large numbers of prototypes the EADS may suffer more from overfitting than the other approaches. Prototype selection is recommended to overcome overfitting in these cases.

1 Introduction

Statistical and structural representations of patterns are two complementary approaches in pattern recognition. Recently, dissimilarity representations [14,10] arose as a bridge between these representations. Dissimilarities can be computed from the original objects, but also on top of features or structures such as graphs or strings. This provides a way for bridging the gap between structural and statistical approaches. Dissimilarities are also a good alternative when the definition and selection of good features can be difficult or intractable (e.g. the search for the optimal subset of features has a computational complexity of $O(2^n)$, where n is the number of features) while a robust dissimilarity measure can be defined more easily for the problem at hand.

The classification of objects represented in a dissimilarity space (DS) has been an active research topic [16,15,17,20,4], but not much attention has been paid

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 75–89, 2013.

to the treatment of the asymmetry that can be present in the dissimilarities. Most traditional classification and clustering methods are devised for symmetric dissimilarity matrices, and therefore cannot deal with asymmetric input. In order to be suitable for these methods, asymmetric dissimilarities need to be symmetrized, for instance by averaging the matrix with its transpose. However, in the dissimilarity space, symmetry is not a required property and therefore a wider range of procedures for classification can be applied.

Asymmetric dissimilarity or similarity measures can arise in several situations; see [9] for a general analysis of the causes of non-Euclidean data. Asymmetry is common in human judgments. Including expert knowledge in defining a (dis)similarity measure, such as for fingerprint matching [4], may lead to asymmetry. In general, matching processes may often lead to asymmetric dissimilarities. Exact matches are often impossible and suboptimal procedures may lead to different matches from A to B than from B to A.

Symmetrization by averaging is widely used before embedding asymmetric dissimilarity data into (pseudo-)Euclidean spaces [14]. The use of a positive semi-definite matrix $K^T K$, where K denotes a nonsymmetric kernel [21] is also proposed in the context of kernel-based classification to make the kernel symmetric. A comparative study of methods for symmetrizing the kernel matrix for the application of the support vector machine (SVM) classifier can be found in [13]. While such methods that require symmetrized matrices show good results, it remains an open question whether asymmetry is an undesirable property, or that it, perhaps, contains useful information that is disregarded during symmetrization.

In this paper we explore using the information from asymmetric dissimilarities by concatenating them into an extended asymmetric dissimilarity space (EADS). Following up on [18], we investigate a broader range of circumstances where EADS may be a good choice for representation, and compare EADS to the directed dissimilarities, as well as to several symmetrization methods. The representation is studied for two shape matching and two multiple instance learning (MIL) problems. We show that EADS is able to outperform the directed and symmetrized dissimilarities, especially in cases where both directed dissimilarities are informative. It must be noted that EADS doubles the dimensionality of the problem, which may not be desirable. Therefore, we also include results using prototype selection in order to compare dissimilarity spaces with the same dimensionality, and show that EADS also leads to competitive results in the examples considered.

We begin with a number of examples that lead to asymmetric dissimilarities in Section 2. The dissimilarity space is explained in Section 3. Ways of dealing with asymmetry are then described: symmetrization (Section 4) and the proposed EADS (Section 5). Experimental results and discussion are provided in Section 6, followed by the conclusions in Section 7.

2 Asymmetric Dissimilarities

Although our notions of geometry may indicate otherwise, asymmetry is a natural characteristic when the concept of similarity or proximity is involved. Just think of a network of roads, where the roads can be one-way streets and one street is longer than the other. It is then clear that traveling from A to B may take longer than returning from B to A. Asymmetric dissimilarities also appear in human judgments [1]: it may be more natural to say that "Dutch is similar to German" than "German is similar to Dutch" because more people might be familiar with the German language and it is therefore a better point of reference for the comparison. Interestingly, this is also evidenced by the number of hits in Google: about ten times as many for the "Dutch is similar to German" sentence. When searching for these sentences in Dutch, the reverse is true.

Here we provide two examples of pattern recognition domains which may also naturally lead to asymmetric dissimilarities.

2.1 Shapes and Images

One possible cause of asymmetry is that the distances used directly on raw data such as images may be expensive to compute accurately. For example in [3], the edit distance used between shapes is originally symmetric. The distance has the problem that the returned values are different if the starting and ending points of the string representation of the shape are changed. In order to overcome this drawback, an improved rotation invariant distance was proposed. The computation of the new distance suffers from a higher computational complexity. Therefore, suboptimal procedures are applied in practice and, as a consequence, the distances returned are asymmetric.

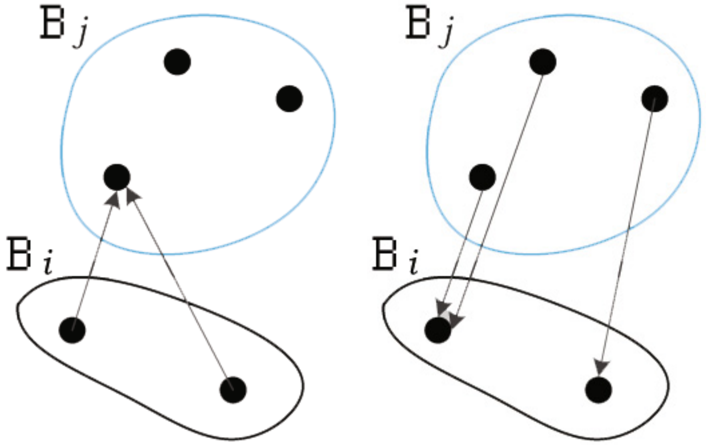
In template matching, the dissimilarity measure may be designed to compute the amount of deformation needed to transform one image into the other as in [12]. The amount of deformation required to transform image I_j into image I_k is generally different from the amount of deformation needed to transform image I_k into I_j. This makes the resulting dissimilarity matrix asymmetric.

2.2 Multiple Instance Learning

Multiple instance learning (MIL) [6] extends traditional supervised learning methods in order to learn from objects that are described by a set (*bag*) of feature vectors (*instances*), rather than a single feature vector only. The bag labels are available, but the labels of the individual instances are not. A bag with n_i instances is therefore represented as (B_i, y_i) where $B_i = \{x_{ik}; k = 1...n_i\}$. In this setting, traditional supervised learning techniques cannot be applied directly.

It is often assumed that the instances have (hidden) labels which influence the bag label. For instance, one assumption is that a bag is positive if and only if at least one of its instances is positive. Such positive instances are also called concept instances. One application for MIL is image classification. An image with several regions or segments can be represented by a bag of instances, where each instance corresponds to a segment in the image. For images that are positive for the "Tiger" class, concept instances are probably segments containing (parts of) a tiger, rather than segments containing plants, trees and other surroundings.

One of the approaches to MIL is to learn on bag level, by defining kernels [11] or (dis)similarities [22,5] between bags. Such dissimilarities are often defined by matching the instances of one bag to instances of another bag, and defining a statistic (such as average or maximum) over these matches. This creates asymmetric dissimilarities, as illustrated in Fig.1.



(a) Dissimilarity from B_i to B_j (b) Dissimilarity from B_j to B_i

Fig. 1. Asymmetry in bag dissimilarities. The minimum distances of one bag's instances are shown. In this paper, the bag dissimilarity is defined as the average of these minimum distances.

The direction in which the dissimilarity is measured defines which instances influence the dissimilarity. When using a positive prototype, it is important that the concept instances are involved, as these instances are responsible for the differences between the classes. Therefore, for positive prototypes it is expected that the dissimilarity from the prototype to the bag is more informative than the dissimilarity from the bag to the prototype. A more detailed explanation of this intuition is given in [5].

3 Dissimilarity Space

The DS was proposed in the context of dissimilarity-based classification [14]. It was postulated as a Euclidean vector space, implying that classifiers proposed for feature spaces can be used there as well. The motivation for this proposal is that the proximity information is more important for class membership than features [14]. Let $R = \{r_1, ..., r_k\}$ be the representation set, where k is its cardinality. This set is usually a subset of the training set T, though a semi-supervised approach with more prototypes than training objects may be preferable [7]. In order to create the DS, using a proper dissimilarity measure d, the dissimilarities of training objects to the prototypes in R are computed. The object representation is a vector of the object's dissimilarities to all the prototypes. Therefore, each dimension of the DS corresponds to the dissimilarities to some prototype. The representation $\mathbf{d}_x$ of an object x is:

$$\mathbf{d}_x = [d(x, r_1) \ ... \ d(x, r_k)] \tag{1}$$

3.1 Prototype Selection

Prototype selection has been proposed for the dimension reduction of DS [16]. Supervised (wrapper) and unsupervised (filter) methods can be considered for this purpose as well as different optimization strategies to guide the search. They select the 'best' prototypes according to their criterion. The selected prototypes are used for the generation of the DS. Prototype selection allows one to obtain low-dimensional spaces avoiding as much as possible a decrease in performance (e.g. classification accuracy). Therefore, they are very useful to achieve a trade-off between the desirable properties of compact representation and reasonable classification accuracy. The approach considered in this study for selecting prototypes is the forward selection optimizing the leave-one-out (LOO) nearest neighbour (1-NN) error (so supervised) in the dissimilarity space for the training set. It starts from the empty set, and sequentially adds the prototype that together with the selected ones ensures the best 1-NN classification accuracy.

4 Combining the Asymmetry Information

For two point sets, there are different ways to combine the two directed asymmetric dissimilarities. The maximum, minimum and average are used extensively and are very intuitive. Let $A = \{a_1, ..., a_k\}$ and $B = \{b_1, ..., b_l\}$ be two sets of points, and $D_1 = d(A, B)$ and $D_2 = d(B, A)$ the two directed dissimilarities. The maximum, minimum and average combiners are defined in (2) to (4) respectively:

$$max(A, B) = \max(D_1, D_2) \tag{2}$$

$$min(A, B) = \min(D_1, D_2) \tag{3}$$

$$avg(A, B) = \frac{1}{2}(D_1 + D_2) \tag{4}$$

All these rules for combining asymmetry information ensure a symmetric measure.

5 Extended Asymmetric Dissimilarity Space

For the purpose of combining the asymmetry information in both directions, we study the EADS. From the two directed dissimilarities D_1, D_2, we have that $D_i \rightarrow X_i \in \mathbb{R}^k, i = 1, 2$ represents the mapping of the dissimilarities to the dissimilarity space. The EADS is constructed by: $[D_1 \;\, D_2] \rightarrow X_1 \times X_2 \in \mathbb{R}^{k \times 2}$, which means that the extended space is the Cartesian product of the two directed spaces. Given the prototypes $R = \{r_1, ..., r_k\}$, the representation of an object in the EADS is defined by:

$$\mathbf{d}_x = [d(x, r_1) \; ... \; d(x, r_k) \;\; d(r_1, x) \; ... \; d(r_k, x)] \tag{5}$$

In the case that we have the full dissimilarity matrix using all training objects as prototypes, the EADS is constructed from the concatenation of the original matrix and its transpose. Rows of this new matrix correspond to the representation of objects in the EADS. As a result, the dimension of the EADS space is twice the dimension of the DS. Classifiers can be trained in the EADS in the same way they are trained in the DS. By doubling the dimension, the expressiveness of the representation is increased. This may be particularly useful when the number of prototypes is not very large. When the number of prototypes is large compared to the number of training objects, the EADS is expected to be more prone to overfitting than any of the symmetrized approaches.

Despite the fact that in the EADS symmetric distances or similarity measures can be used on top of the asymmetric representation, this does not mean that we are not exploiting the asymmetry information present in the original dissimilarities. The original asymmetric dissimilarities in the two directions are used in the object representation that is the input for classifiers in the EADS. These classifiers can use any symmetric distance or kernel computed on top of the representation.

Note that if the asymmetry does not exist in the measure, the representation of objects in the EADS contains the same information replicated. These redundancies in the best case lead to the same classification results as in the standard DS using only one direction [18]. However, it may even be counterproductive since it may lead to overfitting and small sample size problems for some classifiers. Therefore, doubling the dimension is not the cause for possible classification improvements when using the EADS. The fact that the two asymmetric dissimilarities are taken into account in the representation is what may help the classifiers to improve their outcomes.

6 Experiments

In this section we first describe the datasets and how the corresponding dissimilarity matrices are obtained. This is followed by the experimental setup and a discussion of the results.

6.1 Datasets

The dissimilarity dataset Chickenpieces-35-45 is computed from the Chickenpieces image dataset [3]. The images are in binary format representing silhouettes from five different parts of the chicken: wing (117 samples), back (76), drumstick (96), thigh and back (61), and breast (96). From these images the edges are extracted and approximated by segments of length 35 pixels, and a string representation of the angles between the segments is derived. The dissimilarity matrix is composed by edit distances between these strings. The cost function between the angles is defined as the difference in case of substitution, and as 45 in case of insertion or deletion.

The Zongker digit dissimilarity data is based on deformable template matching. The dissimilarity measure was computed between 2000 handwritten NIST digits in 10 classes. The measure is the result of an iterative optimization of the non-linear deformation of the grid [12].

AjaxOrange is a dataset from the SIVAL multiple instance datasets [19]. The original dataset has 25 distinct objects (such as bottle of dish soap called AjaxOrange) portrayed against 10 different backgrounds, and from 6 different orientations, resulting in 60 images for each object. This dataset has been converted into 25 binary MIL datasets by taking one class (AjaxOrange) in this case as the positive class (with 60 bags), and all others (with 1440 bags) as the negative one. Each image is represented by a bag of segments, and each segment is described by a feature vector with color and texture features.

The dissimilarity of two images is computed by what we call the "meanmin" dissimilarity, which is similar to modified versions of the Hausdorff distance:

$$d_{meanmin}(B_i, B_j) = \frac{1}{|B_i|} \sum_{x_{ik} \in B_i} \min_{x_{jl} \in B_j} d(x_{ik}, x_{jl}) \tag{6}$$

where $d(x_{ik}, x_{jl})$ is the squared Euclidean distance between two feature vectors.

Winter Wren is one of the binary MIL bird song datasets [2], created in a similar one-against-all way as SIVAL. Here, a bag is a spectrogram of an audio fragment with different birds singing. A bag is positive for a particular bird species (e.g. Winter Wren) if its song is present in the fragment. There are 109 fragments where the Winter Wren song is heard, and 439 fragments without it. Also here we use (6) to compute the dissimilarities.

The datasets and their properties are shown in Table 1. For each dissimilarity matrix we computed its asymmetry coefficient as follows:

$$AC = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{|d_{ij} - d_{ji}|}{\min(d_{ij}, d_{ji}) + \epsilon} \tag{7}$$

where n is the number of objects in the dataset. This coefficient measures the average normalized difference of the directed dissimilarities and is 0 for symmetric data.

Table 1. Properties of the datasets used in this study, AC refers to the asymmetry coefficient from (7); the larger the AC, the larger the asymmetry

Dataset	# Classes	# Obj. per class	AC
ChickenPieces-35-45	5	117, 76, 96, 61, 96	0.08
Zongker	10	10×200	0.18
AjaxOrange	2	60, 1440	0.31
Winter Wren	2	109, 439	0.23

The formulation in (7) assumes that $d_{ij} \neq 0$ for $i \neq j$, which may not necessarily be true for dissimilarity data. In the case that $d_{ij} = d_{ji}$, a term ϵ with a very small value such as 0.0001 must be added in the denominator to avoid divisions by zero.

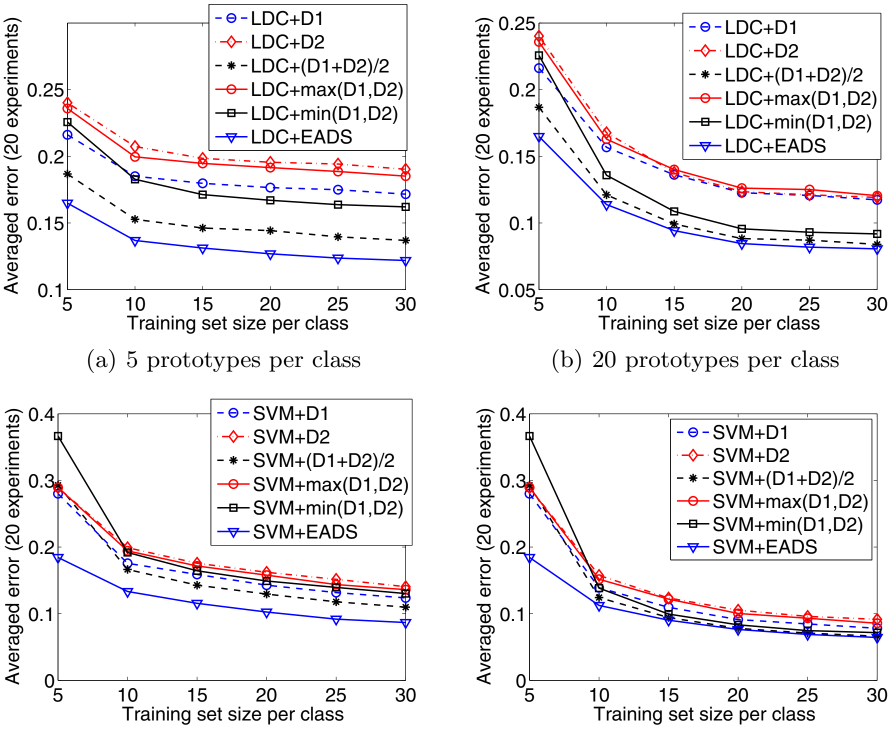
6.2 Experimental Setup

For each of the dissimilarity datasets, we evaluate the performances using asymmetric dissimilarity measures D_1 and D_2, the symmetrized measures (using minimum, average and maximum) and the EADS.

The classifiers compared are the linear discriminant classifier (LDA, but denoted LDC in our experiments) and the SVM, both in the dissimilarity space and implemented in PRTools [8]. For LDC we use regularization parameters $R = 0.01$ and $S = 0.9$, for SVM we use a linear kernel and a regularization parameter $C = 100$. These parameters show reasonable performances on all the datasets under investigation, and are, therefore, constant across all experiments and not optimized to fit a particular dataset.

We provide learning curves over 20 runs for each dissimilarity / classifier combination, for increasing training sizes from 5 to 30 objects per class. In each of the learning curves, the number of prototypes is fixed to either 5 or 20 per class in order to explore the behavior with a small and a large representation set size. This means that the dimensionality of the dissimilarity space is the same for D_1, D_2 and the symmetrized versions, but twice as much for the EADS. The approaches compared are:

- DS resulting from the computation of dissimilarities in the direction from the objects to the prototypes (D_1).
- DS resulting from the computation of dissimilarities from the prototypes to the objects (D_2).
- DS resulting from averaging the dissimilarities in the two directions ($(D_1 + D_2)/2$).
- DS resulting from the maximum of the two dissimilarities ($\max(D_1, D_2)$).
- DS resulting from the minimum of the two dissimilarities ($\min(D_1, D_2)$).
- The extended asymmetric dissimilarity space (EADS).



(a) 5 prototypes per class

(b) 20 prototypes per class

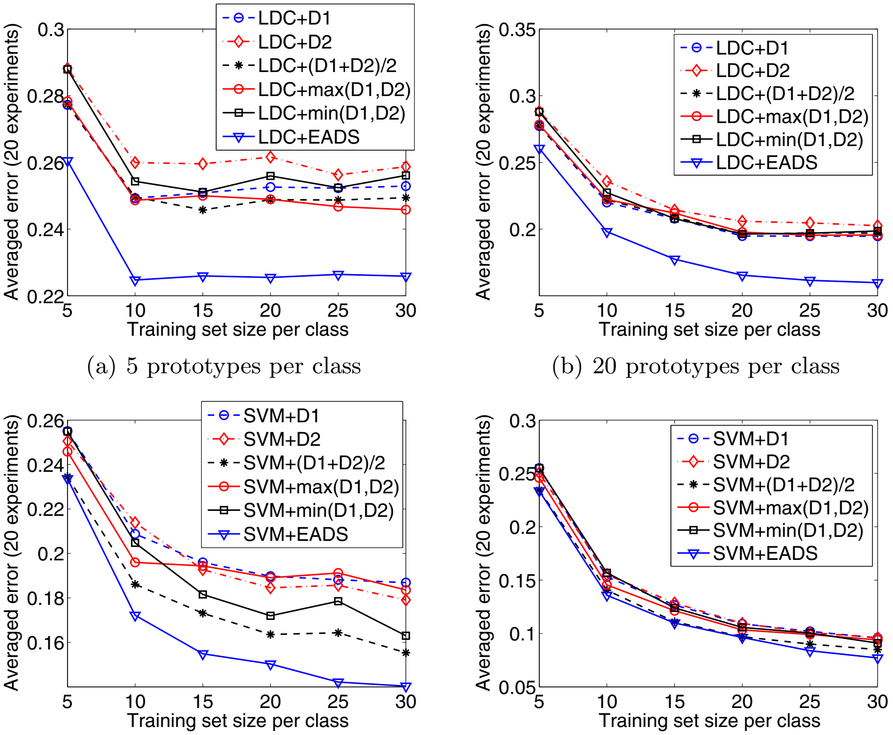
(c) 5 prototypes per class

(d) 20 prototypes per class

Fig. 2. LDC and SVM classification results in dissimilarity spaces for Zongker dataset

6.3 Results and Discussion

In Figs. 2 and 3 it can be seen from the results on the Zongker and Chicken Pieces datasets that the EADS outperforms the other approaches. This is especially true for a small number of prototypes (see Figs. 2 and 3 (a) and (c)). The results of the different approaches become more similar for the representation set of 20 prototypes per class, especially when SVM is used (see Figs. 2 and 3 (d)). The EADS is better than the individual spaces created from the directed dissimilarities, one explanation for this is that the directed dissimilarities provide complementary information so together they are more useful than individually. The EADS contains more information of the relations between the objects than an individual directed DS. The maximum operation is usually very sensitive to noise and outliers what explains its bad performance. The maximum dissimilarity makes objects belonging to the same class more different. These higher differences inside the class are likely to contain noise since objects of the same class should potentially be more similar. The average is more robust than maximum since it combines the information from both directed dissimilarities avoiding in some degree the influence of noise and outliers. Still, by averaging



(a) 5 prototypes per class

(b) 20 prototypes per class

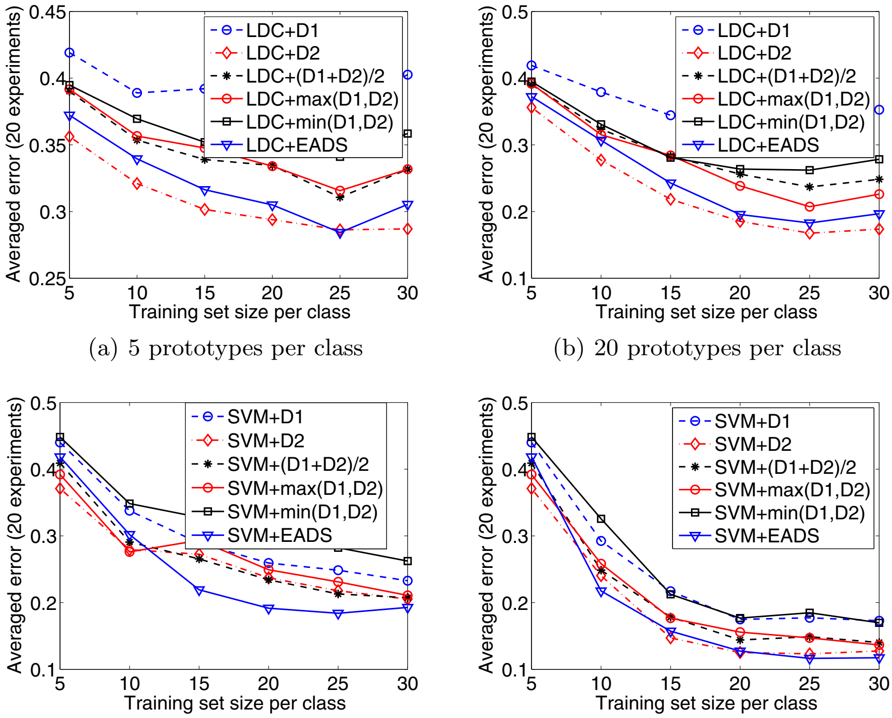
(c) 5 prototypes per class

(d) 20 prototypes per class

Fig. 3. LDC and SVM classification results in dissimilarity spaces for Chicken Pieces dataset

we may hamper the contribution of a very good directed dissimilarity if there is a noisy counterpart. The EADS may improve upon the average because the EADS does not obstructs the contribution of a good directed dissimilarity. The minimum operator is usually worse than EADS and averaging. One possible cause is that by using the minimum, the representation of all the objects is homogenized to some extent because for objects belonging to different classes the separability is decreased by selecting the minimum dissimilarity. Therefore, some discriminatory power is lost.

In AjaxOrange, it is an important observation that D_2 is more informative than D_1, especially for the LDC classifier (see Fig. 4 (a) and (b)). D_2 means that the dissimilarities are measured from the prototypes to the bags. The *meanmin* dissimilarity in (6) therefore ensures that, for a positive prototype, the positive instances (the AjaxOrange bottle) influence the dissimilarity value by definition, as all instances of the prototype have to be matched to instances in the training bag. Measuring the dissimilarity to positive prototypes, on the other hand, may result in very similar values for positive and negative bags because of identical backgrounds, therefore creating class overlap.



(a) 5 prototypes per class

(b) 20 prototypes per class

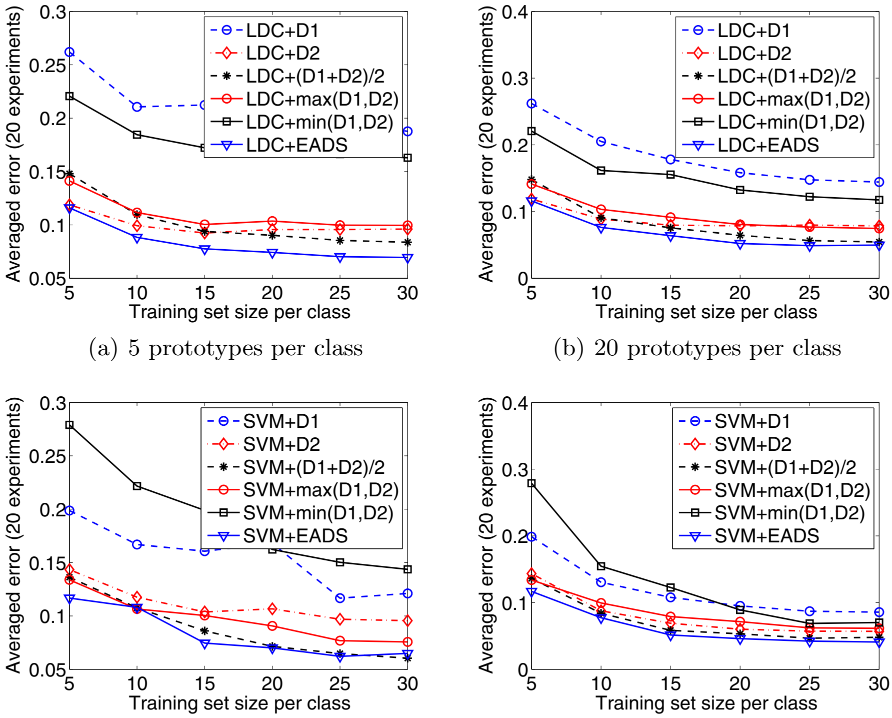
(c) 5 prototypes per class

(d) 20 prototypes per class

Fig. 4. LDC and SVM classification results in dissimilarity spaces for AjaxOrange dataset

Because D_1 contains potentially harmful information, the combining methods do not succeed in combining this information from D_1 and D_2 in a way that is beneficial to the classifier. This is particularly evident for the LDC classifier (see Fig. 4 (a) and (b)), where only EADS has similar (but still worse) performance than D_2. For the SVM classifier, EADS performs well only when a few prototypes are used, but as more prototypes (and more harmful information from D_1) are involved, there is almost no advantage over D_2 alone.

From the results reported in Fig. 5 for Winter Wren, we again see that D_2 is more informative than D_1. However, what is different in this situation is that both directions contain useful information for classification, this is evident due to the success of the average, maximum and EADS combiners. The difference lies in the negative instances (fragments of other birds species, or background objects in the images) of positive bags. While in AjaxOrange, background objects are non-informative, the background in the audio fragments may be informative for the class of the sound. In particular, it is possible that some bird species are heard together more often: e.g. there is a correlation of 0.63 between the labels of Winter Wren and Pacific-slope Flycatcher. Therefore, also measuring



(a) 5 prototypes per class

(b) 20 prototypes per class

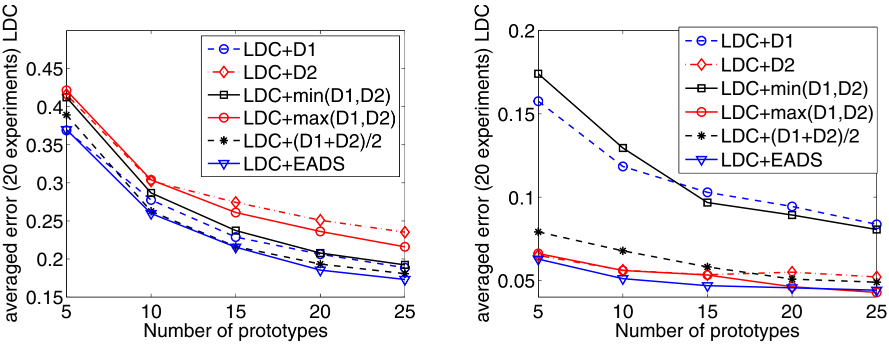
(c) 5 prototypes per class

(d) 20 prototypes per class

Fig. 5. LDC and SVM classification results in dissimilarity spaces for the Winter Wren dataset

dissimilarities to the prototypes produces dissimilarity values that are different for positive and negative bags.

The increased dimensionality of the EADS is one of the main problems of this approach, as in small sample size cases the increased dimensionality may lead to overfitting. In order to overcome this, prototype selection can be considered. We developed other experiments using prototype selection for all the spaces compared. A fixed training set size of 200 objects was used, leading to spaces of dimensionality 5, 10, 15, 20 and 25. The choice to perform the selection of the prototypes was the forward selection optimizing the LOO 1-NN classification error in the training set. One example of standard and MIL dissimilarity datasets were considered: the Zongker and Winter Wren. Prototypes are selected for EADS as it is usually done for a standard DS. Prototypes using the two directed dissimilarities are available as candidates but the prototype selection method may discard one of the two or maybe both if they are not discriminative according to the selection criterion. The EADS is compared now with the other spaces on the basis of the same dimensionality.



(a) Zongker

(b) Winter Wren

Fig. 6. Classification results after prototype selection for the Zongker and Winter Wren datasets

From the results in Fig. 6 (a) it can be seen that, for the Zongker dataset, the best approaches are the EADS and the average. An interesting observation is that this dataset is intrinsically high-dimensional because the number of principal components (PCs) that retain 95% of the data variance is equal to 529. The average approach adds more information in each dimension since every dissimilarity encodes a combination of two. This implies that, for the dimensions considered that are small compared to 529, it performs as good as the EADS. On the contrary, the Winter Wren dataset is intrinsically low-dimensional, since the number of PCs retaining 95% of the data variance is equal to 3. This is a possible explanation of why the EADS is the best in this case (see Fig. 6 (b)), because the average approach is likely to introduce some noise.

One interesting issue of using prototype selection in EADS is that not only the dimensions are decreased, but also the accuracy of the EADS itself may be improved especially in the datasets where one of the directed dissimilarities is the best and the other is very bad (e.g. MIL datasets). The EADS without prototype selection in these cases may be worse than the best directed dissimilarity (see Fig. 4 (a) and (b)). However, by using a suitable prototype selection method in the EADS, only the prototypes from the best directed dissimilarity should be kept, and noisy prototypes from the bad directed dissimilarity should be discarded. This should make the results of the EADS similar to those of the best directed dissimilarity. This can be achieved if a proper prototype selection method is used. In the prototype selection executed for the Winter Wren, where one directed dissimilarity is remarkably better than the other, this can partially be seen. For example, in one run, the method selected 18 prototypes from the best directed dissimilarity in the set of 25 prototypes selected. Future work will include the study of suitable prototype selectors for EADS.

7 Conclusions

In this paper we study the EADS as an alternative to different approaches for dealing with asymmetric dissimilarities. The EADS outperforms the other approaches for a small number of prototypes in standard dissimilarity datasets, when both dissimilarities are about equally informative.

In MIL datasets, conclusions are slightly different because of the way the dissimilarities are created. It may be the case that the best option is one of the directed dissimilarities. However, if there is no knowledge on which directed dissimilarity is the best, the EADS may be the best choice. This especially holds when only a low number of prototypes is available.

It should be noted that the EADS increases the dimensionality as opposed to other combining approaches, therefore increasing the risk of overfitting. Prototype selection should be considered to keep the dimensionality low. After prototype selection, the EADS also shows good results in examples of intrinsically low- and high-dimensional datasets. However, for intrinsically high-dimensional datasets, averaging is also worth considering as combining rule.

Our main conclusion is that asymmetry is not an artefact that has to be removed in order to apply embedding or kernel methods to the classification problem. On the contrary, asymmetric dissimilarities may contain very useful information, and it is advisable to consider the dissimilarity representation as a means to fully use this information.

References

1. Bowdle, B., Gentner, D.: Informativity and asymmetry in comparisons. Cogn. Psychol. 34(3), 244–286 (1997)
2. Briggs, F., Lakshminarayanan, B., Neal, L., Fern, X., Raich, R., Hadley, S., Hadley, A., Betts, M.: Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. J. Acoust. Soc. Am. 131, 4640 (2012)
3. Bunke, H., Bühler, U.: Applications of approximate string matching to 2D shape recognition. Pattern Recogn. 26(12), 1797–1812 (1993)
4. 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)
5. Cheplygina, V., Tax, D.M.J., Loog, M.: Class-dependent dissimilarity measures for multiple instance learning. 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. 602–610. Springer, Heidelberg (2012)
6. Dietterich, T., Lathrop, R., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artif. Intell. 89(1-2), 31–71 (1997)
7. Dinh, C., Duin, R.P.W., Loog, M.: A study on semi-supervised dissimilarity representation. In: International Conference on Pattern Recognition (2012)
8. Duin, R.P.W., Juszczak, P., Paclik, P., Pękalska, E., De Ridder, D., Tax, D.M.J., Verzakov, S.: A Matlab toolbox for pattern recognition. PRTools version 3 (2000)

9. Duin, R.P.W., Pękalska, E.z.: Non-Euclidean dissimilarities: Causes and informativeness. In: Hancock, E.R., Wilson, R.C., Windeatt, T., Ulusoy, I., Escolano, F. (eds.) SSPR&SPR 2010. LNCS, vol. 6218, pp. 324–333. Springer, Heidelberg (2010)
10. Duin, R.P.W., Pękalska, E.: The dissimilarity space: bridging structural and statistical pattern recognition. Pattern Recogn. Lett. 33(7), 826–832 (2012)
11. Gärtner, T., Flach, P., Kowalczyk, A., Smola, A.: Multi-instance kernels. In: Proc. of the 19th Int. Conf. on Machine Learning, pp. 179–186 (2002)
12. Jain, A.K., Zongker, D.: Representation and recognition of handwritten digits using deformable templates. IEEE Trans. Pattern Anal. Mach. Intell. 19, 1386–1391 (1997)
13. Muñoz, A., de Diego, I.M., Moguerza, J.M.: Support vector machine classifiers for asymmetric proximities. In: Kaynak, O., Alpaydın, E., Oja, E., Xu, L. (eds.) ICANN/ICONIP 2003. LNCS, vol. 2714, pp. 217–224. Springer, Heidelberg (2003)
14. Pękalska, 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)
15. Pękalska, E., Duin, R.P.W.: Beyond traditional kernels: Classification in two dissimilarity-based representation spaces. IEEE Trans. Syst. Man Cybern. C, Appl. Rev. 38(6), 729–744 (2008)
16. Pękalska, E., Duin, R.P.W., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recogn. 39(2), 189–208 (2006)
17. Pękalska, E., Paclik, P., Duin, R.P.W.: A generalized kernel approach to dissimilarity-based classification. J. Mach. Learn. Res. 2, 175–211 (2002)
18. Plasencia-Calaña, Y., García-Reyes, E.B., Duin, R.P.W., Orozco-Alzate, M.: On using asymmetry information for classification in extended dissimilarity spaces. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (eds.) CIARP 2012. LNCS, vol. 7441, pp. 503–510. Springer, Heidelberg (2012)
19. Rahmani, R., Goldman, S., Zhang, H., Krettek, J., Fritts, J.: Localized content based image retrieval. In: Proc. of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval, pp. 227–236. ACM (2005)
20. 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)
21. Schölkopf, B., Mika, S., Burges, C.J.C., Knirsch, P., Müller, K.R., Rätsch, G., Smola, A.J.: Input space versus feature space in kernel-based methods. IEEE Trans. Neural Netw. 10(5), 1000–1017 (1999)
22. Tax, D.M.J., Loog, M., Duin, R.P.W., Cheplygina, V., Lee, W.-J.: Bag dissimilarities for multiple instance learning. In: Pelillo, M., Hancock, E.R. (eds.) SIMBAD 2011. LNCS, vol. 7005, pp. 222–234. Springer, Heidelberg (2011)

Information-Theoretic Dissimilarities for Graphs

Francisco Escolano[1], Edwin R. Hancock[2], Meizhu Liu[3],
and Miguel Angel Lozano[1]

[1] University of Alicante, Spain
{sco,malozano}@dccia.ua.es
[2] University of York, UK
erh@cs.york.ac.uk
[3] Siemens Corporate Research, Princeton, USA
liufkmc@gmail.com

Abstract. This is a survey paper in which we explore the connection between graph representations and dissimilarity measures from an information-theoretic perspective. Firstly, we pose graph comparison (or indexing) in terms of *entropic manifold alignment*. In this regard, graphs are encoded by multi-dimensional point clouds resulting from their embedding. Once these point clouds are aligned, we explore several dissimilarity measures: multi-dimensional statistical tests (such as the Henze-Penrose Divergence and the Total Variation k-dP Divergence), the Symmetrized Normalized Entropy Square variation (SNESV) and Mutual Information. Most of the latter divergences rely on multi-dimensional entropy estimators. Secondly, we address the representation of graphs in terms of populations of tensors resulting from characterizing topological multi-scale subgraphs in terms of covariances of informative spectral features. Such covariances are mapped to a proper tangent space and then considered zero-mean Gaussian distributions. Therefore each graph can be encoded by a linear combination of Gaussians where the coefficients of the combination rely on unbiased geodesics. Distributional graph representations allows us to exploit a large family of dissimilarities used in information theory. We will focus on Bregman divergences (particularly Total Bregman Divergences) based on the Jensen-Shannon and Jensen-Rényi divergences. This latter approach is referred to as *tensor-based distributional comparison* for distributions can be also estimated from embeddings through Gaussian mixtures.

1 Introduction

One of the key elements for building a pattern theory is the definition of a set of principled dissimilarity measures between the mathematical objects motivating that theory. For instance, in vectorial pattern recognition, one of the fundamental axes of an information theoretic algorithm is the definition of a divergence: mutual information, Kullback-Leibler, Bregman divergence, and so on [1]. However, when the object at hand is an structural pattern, the extension of the latter concepts, as the first step for formulating an information theory for graphs, is a challenging task. Following the path of entropy bypass estimators which do

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 90–105, 2013.

not rely on probability density functions, we address the point of bypassing the rigid discrete representation of graphs. This implies defining transformations either between vertices and multi-dimensional spaces (embeddings) or between subgraphs and other spaces like tensorial ones. In both cases, we pursue probabilistic representations which encode the rich topological information of the original graphs. With such representations at hand it is possible to build principled information-theoretic divergences whose estimation is highly influenced by the development of bypass methods. The rest of the paper is a survey of such representations and divergences. Firstly we will investigate divergences between embeddings (including kernels) and later we will address the transformation of a graph into a set of *node coverages* (redundancy is needed to some extent) so that we can propose divergences between tensors projected into a proper tangent space. The computational cost of computing both types of representation is a serious drawback that recommends a trade-off. At the end of the paper we will address such trade-off by proposing the computation of mutual information between graphs.

2 Divergences between Embeddings

Let $G = (V, E)$ be an undirected and unweighted graph with: node set V, edge set E, adjacency matrix $\mathbf{A}$ of dimension $n \times n$ (where $n = |V|$) and Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$ (where $\mathbf{D}$ is the diagonal degree matrix whose trace is the volume of the graph $vol = trace(\mathbf{D})$). Then, a *graph embedding* is typically a function of the eigenvalues and/or eigenvectors of $\mathbf{L}$. For instance, Heat Kernel (HK) and Commute Times (CT) embeddings result from a function $\mathcal{F}(.)$ of the Laplacian eigen-decomposition $\mathcal{F}(\mathbf{L}) = \Phi\mathcal{F}(\Lambda)\Phi^T = \Theta^T\Theta$. For CT, $\mathcal{F}(\mathbf{L}) = \sqrt{vol}\Lambda^{-1/2}\Phi^T$; for HK we have $\mathcal{F}(\mathbf{L}) = \exp\left(-\frac{1}{2}t\Lambda\right)$ where t is time; and for Diffusion Maps (DM), we have $\mathcal{F}(\mathbf{L}) = \Lambda^t$ where Λ results from a generalized eigenvalue/eigenvector problem as in the case of Laplacian Eigenmap (LEM) where $\mathcal{F}(\mathbf{L}) = \Phi$. Finally, ISOMAP considers the top eigenvectors of the geodesic distance matrix. Different embeddings yield different point distributions for the same dimensionality. In general $\mathcal{F}(\mathbf{L}) = \Theta^T\Theta$, where Θ results from the Young-Householder decomposition. In general, Θ is an $n \times n$ matrix where column $i-$th represent the n coordinates of the $i-$th node in the space defined by the embedding, therefore $\Theta : V \rightarrow \mathbb{R}^n$ and different embeddings produce also different multi-dimensional point clouds. For instance, CT produces denser clouds than LEM (see [2]). Such point clouds encode spectral properties of the graph. In this regard, CT embedding is an interesting choice because the squared Euclidean distance between two columns (mappings of the corresponding nodes) is equal to the commute time between the corresponding nodes, that is $||\Theta^{(i)} - \Theta^{(j)}||^2 = CT(u, v)$. In [3] we show that CT embedding outperforms the other ones in terms of retrieval/recall for the best dissimilarity measure (see below). In addition, the fact that the latter embedding induces a metric allows us to work in the multi-dimensional space of the embedding, where problems such as finding prototypes, are more tractable and then return to the original embedding space via inverse embedding [4]. Therefore, in this section we exploit commute time embeddings from normalized Laplacian matrices $\mathcal{L} = D^{-1/2}\mathbf{L}D^{-1/2}$ which is given by $\mathcal{F}(\mathbf{L}) = \sqrt{vol}\Lambda^{-1/2}\Phi^T\mathbf{D}^{-1/2}$. Therefore, Θ has the following form for the $i-$th node:

$$\Theta^{(i)} = \sqrt{\frac{vol}{d_i}}\left(\frac{1}{\sqrt{\lambda^{(2)}}}\phi^{(2)}(i)\ldots\frac{1}{\sqrt{\lambda^{(n)}}}\phi^{(n)}(i)\right)^T \ , \tag{1}$$

where $d_i = \mathbf{D}(i,i)$, and $\{\phi^{(z)}, \lambda^{(z)}\}_{z=2\ldots n}$ are the non-trivial eigenvectors and eigenvalues of $\mathbf{L}$. The *commute time* $CT(i,j)$ is the expected time for the random walk to travel from node i to reach node j and then return. As a result $CT(i,j) = O(i,j) + O(j,i)$. In terms of the Green's function the commute time is given by

$$CT(i,j) = vol\left(G(i,i) + G(j,j) - 2G(i,j)\right) \ , \tag{2}$$

where

$$G(i,j) = \sum_{z=2}^{n}\frac{1}{\lambda^{(z)}}\frac{\phi^{(z)}(i)}{\sqrt{d_i}}\frac{\phi^{(z)}(j)}{\sqrt{d_j}} \ . \tag{3}$$

Therefore, the spectral definition of CT is given by

$$CT(i,j) = vol\sum_{z=2}^{n}\frac{1}{\lambda^{(z)}}\left(\frac{\phi^{(z)}(i)}{\sqrt{d_i}} - \frac{\phi^{(z)}(j)}{\sqrt{d_j}} \ .\right)^2 \tag{4}$$

Let $X = (V_X, E_X)$ and $Y = (V_Y, E_Y)$ be two undirected and unweighted graphs with respective node-sets V_X and V_Y, edge-sets E_X and E_Y and number of nodes $n = |V_X|$ and $m = |V_Y|$. Given a dimension $d << min(m,n)$, their *approximate CT* are given by

$$\widehat{CT}(i,j) = vol_X\sum_{z=2}^{d+1}\frac{1}{\lambda_X^{(z)}}\left(\frac{\phi_X^{(z)}(i)}{\sqrt{d_i}} - \frac{\phi_X^{(z)}(j)}{\sqrt{d_j}}\right)^2 \leq CT(i,j) \ , \tag{5}$$

for $i,j \in V_X$, and similarly for nodes $u,v \in V_Y$. Let $i \in V_X$ and $u \in V_Y$ be nodes of graphs X and Y and let $\mathcal{T}$ be *a non rigid transformation which aligns the approximated manifold* $\hat{\Theta}_Y$ with $\hat{\Theta}_X$. Then, we can define $\widetilde{CT}^*(i,u) = ||\hat{\Theta}_X^{(i)} - \mathcal{T}^*(\hat{\Theta}_Y^{(u)})||^2$ where $\mathcal{T}^*(.)$ is the optimal non rigid transformation aligning $\hat{\Theta}_Y$ with $\hat{\Theta}_X$. Finding $\widetilde{CT}^*$ is then posed in terms of *non-rigid manifold alignment.* In this regard, the CPD (Coherent Point Drift) formulation [5] is particularly useful in the context of manifold alignment because it generalizes non-rigid alignment to an arbitrary number of dimensions, say d, of the input data (manifolds in this case). The key point to note here is that the ability of CPD for managing an arbitrary number of dimensions allows us to increase the impact of the structural information contained in the graphs in pattern recognition and shape recognition tasks as we increase d. At low d we cancel high frequencies in the manifold which contain the local structure of the graphs being compared. In practice we are performing non-linear (kernel) PCA. Later, we will show the impact, in terms of pattern discrimination, of setting d with respect to the estimated *intrinsic dimension* [6].

2.1 Symmetrized Normalized Entropy Square Variation

After obtaining the optimal transformation $\mathcal{T}^*(\hat{\Theta}_Y)$ a principled similarity measure between the manifolds requires incorporating a criterion that compares the spatial distributions of both the deformed/aligned $\mathcal{T}^*(\hat{\Theta}_Y)$ and the static $\hat{\Theta}_X$ manifolds. If well designed, such a *distributional measure* could quantify implicitly both the matching costs and the transformation cost. Distributional measures are not new in point registration. In [7] the estimation of the cumulative distribution functions (CDFs) of the point sets and then the estimation and minimization of their Havrda-Charvát (HC) divergence drives point-set registration. In this latter case, the quality of the registration is evaluated through a Kolmogorov-Smirnov test for 2D/3D. Therefore, Information Theory (IT) is a valuable source of inspiration for cost functions for registration. However, their role in point-set similarity (manifolds in this case) has been poorly evaluated in the past. In this regard, here we introduce a new IT measure referred as the *normalized-entropy-square variation* (NESV) [3]:

$$\begin{aligned}\mathcal{V}(\hat{\Theta}_X, \mathcal{T}^*(\hat{\Theta}_Y)) &= \frac{(H(\mathcal{T}^*(\hat{\Theta}_Y)) - H(\hat{\Theta}_X))^2}{H(\mathcal{T}^*(\hat{\Theta}_Y)) + H(\hat{\Theta}_X)} \\ &= \frac{(H(\mathcal{T}^*(\hat{\Theta}_Y)) - H(\hat{\Theta}_X))^2}{I(\mathcal{T}^*(\hat{\Theta}_Y); \hat{\Theta}_X) + H(\mathcal{T}^*(\hat{\Theta}_Y), \hat{\Theta}_X)} \ ,\end{aligned} \tag{6}$$

where $H(.)$ and $H(.,.)$ are respectively the Shannon entropy and joint entropy, and $I(.;.)$ denotes the mutual information. The above measure quantifies the degree of entropy similarity after alignment, normalized by the sum of entropies. Normalization is key when comparing graphs (manifolds) with a significantly different number of nodes (points) and is also consistent with mutual information maximization. Despite its discrimination capability (we will be more precise in the experimental section) one of the benefits of the NESV is that we can *infer a kernel between the probability functions for the manifolds* and, thus, implicitly between the graphs. Inferring such kernels is of pivotal importance for principled comparisons of the probability distributions associated with the manifolds [8], and when these manifolds result from graph embedding we are implicitly learning kernels between graphs. It is straightforward to prove that the induced p.d. kernel is

$$K_{\mathcal{V}}(p_X, p_Y^*) = \frac{e^{-\beta(H(\mathcal{T}^*(\hat{\Theta}_Y)) - H(\hat{\Theta}_X))^2}}{H(\mathcal{T}^*(\hat{\Theta}_Y)) + H(\hat{\Theta}_X) + a} \tag{7}$$

where $a > 0, \beta > 0$, p_X and p_Y^* are the pdfs induced by $\hat{\Theta}_X$ and $\mathcal{T}^*(\hat{\Theta}_Y)$ respectively. As a result $K_{\mathcal{V}}$ is p.d. However, *it is not a kernel* because, in general $\mathcal{V}(\hat{\Theta}_X, \mathcal{T}^*(\hat{\Theta}_Y)) \neq \mathcal{V}(\mathcal{T}^*(\hat{\Theta}_X), \hat{\Theta}_Y)$, that is, it is not symmetric with respect to transforming $\hat{\Theta}_X$, or equivalently locating $\mathcal{T}^*(\hat{\Theta}_X)$ in order to match $\hat{\Theta}_Y$. Consequently, the *symmetrized normalized-entropy-square variation* SNESV is defined by

$$\mathcal{SV}(\hat{\Theta}_X, \hat{\Theta}_Y) = \frac{(H(\mathcal{T}^*(\hat{\Theta}_Y)) - H(\hat{\Theta}_X))^2}{H(\mathcal{T}^*(\hat{\Theta}_Y)) + H(\hat{\Theta}_X)} + \frac{(H(\mathcal{T}^*(\hat{\Theta}_Y)) - H(\hat{\Theta}_X))^2}{H(\mathcal{T}^*(\hat{\Theta}_X)) + H(\hat{\Theta}_Y)} \ . \tag{8}$$

Consequently, its associated p.d. kernel is

$$K_{\mathcal{SV}}(p_X^*, p_Y^*) = \frac{e^{-\beta_y(H(\mathcal{T}^*(\hat{\Theta}_Y))-H(\hat{\Theta}_X))^2}}{H(\mathcal{T}^*(\hat{\Theta}_Y)) + H(\hat{\Theta}_X) + a_y} + \frac{e^{-\beta_x(H(\mathcal{T}^*(\hat{\Theta}_X))-H(\hat{\Theta}_Y))^2}}{H(\mathcal{T}^*(\hat{\Theta}_X)) + H(\hat{\Theta}_Y) + a_x} \ , \tag{9}$$

where β_y, β_x, a_y, $a_x > 0$. If we have a training set, the latter parameters must be learned in order to optimize the kernel machine used for manifold/graph classification.

2.2 Leonenko et al. Entropy Estimator

One of the problems of using IT measures in high dimensional domains is the estimation of the measures themselves. Given that the dimensionality of the manifolds may be too high for a plug-in entropy estimator, in this work we exploit the kNN-based bypass estimator proposed by Leonenko et al. [9]:

$$\hat{H}_{N,k,1} = \frac{1}{N}\sum_{i=1}^{N} \log\{(N-1)e^{-\Psi(k)}V_d(\rho_{k,N-1}^{(i)})^d\}, \tag{10}$$

where N is the number of i.i.d. samples (points) $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in R^d, k the maximum number of nearest neighbors, $\Psi(k) = \Gamma'(k)/\Gamma(k) = -\gamma + A_{k-1}$ the digamma function with $\gamma \approx 0.5772$ (Euler constant) and $A_0 = 0$,$A_j = \sum_{i=1}^{j} 1/i$, $V_d = \pi^{d/2}/\Gamma(d/2+1)$ is the volume of the unit ball $\mathcal{B}(0,1)$ in $\mathbb{R}^d$, and $\rho_{k,N-1}^{(i)}$ is the $k-$th nearest neighbor distance from $\mathbf{x}_i$ to some other $\mathbf{x}_j$. This estimator is both consistent and fast to compute.

2.3 Henze-Penrose Divergence

The Henze and Penrose divergence [10] between two distributions f and g is

$$D_{HP}(f||g) = \int \frac{p^2 f^2(z) + q^2 g^2(z)}{pf(z) + qg(z)} dz \ \ , \tag{11}$$

where $p \in [0,1]$ and $q = 1-p$. This divergence is the limit of the Friedman-Rafsky run length statistic [11], that in turn is a multi-dimensional generalization based on MST[1]s of the Wald-Wolfowitz test. The Wald-Wolfowitz statistic computes the divergence between two distributions f_X and g_O in $\mathbb{R}^d$, when $d = 1$, from two sets of n_x and n_o samples, respectively. First, the $n = n_x + n_o$ samples are ordered in ascending order and labeled as X and O according to their corresponding distribution. The test is based on the number of runs R, being a run a sequence of consecutive and equally labeled samples. The test is calculated as:

$$W = \frac{R - \frac{2n_o n_x}{n} - 1}{\left(\frac{2n_x n_o(2n_x n_o - n)}{n^2(n-1)}\right)^{\frac{1}{2}}} \ \ . \tag{12}$$

[1] Minimum-Spanning Tree.

The two distributions are considered similar if R is low and therefore W is also low. This test is consistent in the case that n_x/n_o is not close to 0 or ∞, and when $n_x, n_o \to \infty$. The Friedman-Rafsky test generalizes Eq. 12 to $d > 1$, due to the fact that the MST relates samples that are close in $\mathbb{R}^d$. Let $X = \{\mathbf{x}_i\}$ and $O = \{\mathbf{o}_i\}$ be two sets of samples drawn from f_X and g_O, respectively. The steps of the Friedman-Rafsky test are:

1. Build the MST over the samples from both X and O.
2. Remove the edges that do not connect a sample from X with a sample from O.
3. The proportion of non-removed edges converges to 1 minus the Henze Penrose divergence (Eq. 11) between f_X and g_O.

See an example in Fig. 1.

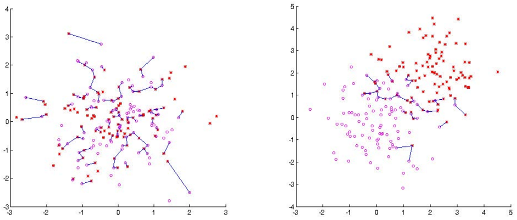


Fig. 1. Two examples of Friedman-Rafsky estimation of the Henze and Penrose divergence applied to samples drawn from two Gaussian densities. Left: the two densities have the same mean and covariance matrix ($D_{HP}(f||g) = 0.5427$). Right: the two densities have different means ($D_{HP}(f||g) = 0.8191$).

2.4 Total Variation k-dP Divergence

The main drawback of both the Henze-Penrose and the Leonenko's-based divergences is the high temporal cost of building the underlying data structures (e.g. MSTs). This computational burden is due to the calculation of distances. A new entropy estimator recently developed by Stowell and Plumbley overcomes this problem [12]. They proposed an entropy estimation algorithm that relies on data spacing without computing any distance. This method is inspired by the data partition step in the k-d tree algorithm. Let X be a d-dimensional random variable, and $f(\mathbf{x})$ its pdf. Let $A = \{A_j | j = 1, \ldots, m\}$ be a partition of X for which $A_i \cap A_j = \emptyset$ if $i \neq j$ and $\bigcup_j A_j = X$. Then, we can approximate $f(\mathbf{x})$ in each cell as $f_{A_j} = \int_{A_j} f(\mathbf{x})/\mu(A_j)$, where $\mu(A_j)$ is the d-dimensional volume of A_j. If $f(\mathbf{x})$ is unknown and we are given a set of samples $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ from it, being $\mathbf{x}_i \in \mathbb{R}^d$, we can approximate the probability of $f(x)$ in each cell as $p_j = n_j/n$, where n_j is the number of samples in cell A_j. Thus, $\hat{f}_{A_j}(\mathbf{x}) = n_j/n\mu(A_j)$ being $\hat{f}_{A_j}(x)$ a consistent estimator of $f(\mathbf{x})$ as $n \to \infty$. Then, to obtain the entropy estimation for A we have

$$\hat{H} = \sum_{j=1}^{m} \frac{n_j}{n} \log\left(\frac{n}{n_j}\mu(A_j)\right) \ . \tag{13}$$

The partition is created recursively following the data splitting method of the k-d tree algorithm. At each level, data is split at the median along one axis. Then, data splitting is recursively applied to each subspace until an uniformity stop criterion is satisfied. The aim of this stop criterion is to ensure that there is an uniform density in each cell in order to best approximate $f(\mathbf{x})$. The chosen uniformity test is fast and depends on the median. The distribution of the median of the samples in A_j tends to a normal distribution that can be standardized as:

$$Z_j = \sqrt{n_j}\frac{2med_d(A_j) - min_d(A_j) - max_d(A_j)}{max_d(A_j) - min_d(A_j)} \ , \tag{14}$$

where $med_d(A_j)$, $min_d(A_j)$ and $max_d(A_j)$ are the median, minimum and maximum, respectively, of the samples in cell A_j along dimension d. An improbable value of Z_j, that is, $|Z_j| > 1.96$ (the 95% confidence threshold of a standard normal distribution) indicates significant deviation from uniformity. Non-uniform cells should be divided further. An additional heuristic is included in the algorithm in order to let the tree reach a minimum depth level: the uniformity test is not applied until there are less than $\sqrt{n}$ data points in each partition, that is, until the level $L_n = \lceil \frac{1}{2}\log_2(n) \rceil$ is reached. Then, our k-d partition based divergence (k-dP divergence) follows the spirit of the *total variation* distance, but may also be interpreted as a L1-norm distance. The total variation distance between two probability measures P and Q on a σ-algebra F[2] is given by $sup\{|P(X) - Q(X)| : X \in F\}$. In the case of a finite alphabet, the total variation distance is $\delta(P,Q) = \frac{1}{2}\sum_{\mathbf{x}} |P(\mathbf{x}) - Q(\mathbf{x})|$. Let $f(\mathbf{x})$ and $g(\mathbf{x})$ be two distributions, from which we draw a set X of n_x samples and a set O of n_o samples, respectively. If we apply the partition scheme of the k-d partition algorithm to the set of samples $X\bigcup O$, the result is a partition A of $X\bigcup O$, being $A = \{A_j | j = 1, \ldots, p\}$. For $f(x)$ and $g(x)$ the probability of any cell A_j is respectively given by

$$f(A_j) = \frac{n_{x,j}}{n_x} = f_j, \ \ g(A_j) = \frac{n_{o,j}}{n_o} = g_j \tag{15}$$

where $n_{x,j}$ is the number of samples of X in cell A_j and $n_{o,j}$ is the number of samples of O in the cell A_j. Since the same partition A is applied to both sample sets, and considering the set of cells A_j a finite alphabet, we can compute the *k-dP total variation divergence* between $f(\mathbf{x})$ and $g(\mathbf{x})$ as:

$$D_{kdP}(f||g) = \frac{1}{2}\sum_{j=1}^{p} |f_j - g_j| \ . \tag{16}$$

[2] A σ-algebra over a set X is a non-empty collection of subsets of X (including X itself) that is closed under complementation and countable unions of its members.

The latter divergence satisfies $0 \leq D(f||g) \leq 1$. The minimum value $D(O||X) = 0$ is obtained when all the cells A_j contain the same proportion of samples from X and O. By the other hand, the maximum value $D(O||X) = 1$ is obtained when all the samples in any cell A_j belong to the same distribution. We show in Fig. 2 two examples of divergence estimation using Eq. 16.

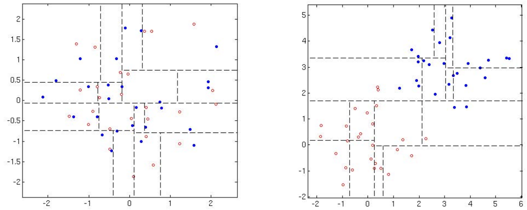


Fig. 2. Two examples of divergence estimation applied to samples drawn from two Gaussian densities. Left: both densities have the same mean and covariance matrix ($D(f||g) = 0.24$). Right: the two densities have different means. Almost all the cells contain samples obtained from only one distribution ($D(f||g) = 0.92$).

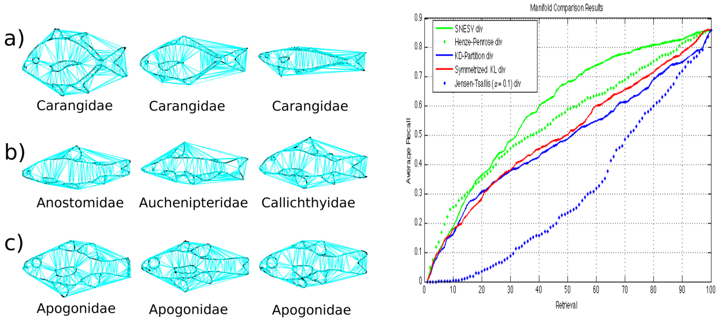


Fig. 3. Examples of the Gator database (left) and average recall-retrieval curves (right)

2.5 Retrieval from GatorBait

In order to test SNESV, Henze Penrose and kdP we have chosen a challening database, the *GatorBait_100*[3] ichthyology database. GatorBait has 100 shapes representing fishes from 30 different classes [3] . We have extracted Delaunay graphs from their shape quantization (Canny algorithm followed by contour decimation). Since the classes are associated to fish genus and not to species,

[3] http://www.cise.ufl.edu/~anand/publications.html

we find high intraclass variability in many cases – see a) in Fig. 3-left where the corresponding class has 8 species. There are also very similar species from different classes (row b)) and few homogeneous clases (row c)). There are 10 classes with one species (not included in the analysis and performance curves), 11 with $1-3$ individuals, 5 with $4-6$ individuals and only 4 classes with more than 6 species. Hence, it is hard to devise a measure which produces an average retrieval-recall curve (Fig. 3-right) far above the diagonal. This is the case for SNESV. We have focused all our analysis in the curves for $d=5$, where the 5D setting is selected experimentally since the estimations of the intrinsic dimensions are in the interval (11.6307 ± 2.8846). Overestimation is due to the curse of dimensionality. For instance, for 10D, SNESV is near diagonal. The more competitive IT measure with respect to SNESV is the Henze-Penrose divergence. Two alternative measures which originate a p.d. kernel are studied: a) the symmetrized Kullback-Leibler divergence which is also close to the diagonal and b) the Jensen-Tsallis divergence for $q=0.1$ (both estimated through Leonenko's method). We also studied the behavior of a total variation (L_1) divergence (kdP) where the entropy is estimated through k-d tree partitions. In all cases $k=4$. In all these experiments the CT embedding is considered.

3 Tensor-Based Divergences

3.1 Tensor-Based Graph Representations

Let $G=(V,E)$ with $|V|=n$. Then the *history of a node* $i \in V$ is $h_i(G) = \{e(i), e^2(i)), \ldots, e^p(i)\}$ where: $e(i) \subseteq G$ is the *first-order expansion subgraph* given by i and all $j \sim i$, $e^2(i) = e(e(i)) \subseteq G$ is the *second-order expansion* consisting on $z \sim j : j \in V_{e(i)}, z \notin V_{e(i)}$, and so on until p cannot be increased. If G is connected $e^p(i) = G$, otherwise $e^p(i)$ is the connected component to which i belongs [16]. Every $h_i(G)$ defines a set of subgraphs $h_i(G) = \{e(i), e^2(i)), \ldots, e^p(i)\}$ where $e^l(i) \subseteq e^k(i)$ when $k > l$. If we select $k < p$ we obtain a $k-$order *partial node coverage* given by the subgraph $e^k(i)$. If we overlap the $k-$order partial coverages associated to all $i \in V$ we obtain a $k-$order *graph coverage.* For instance, in Fig. 4 we show that two subgraphs with the same order but around different nodes (B and C) with the same order have, in general, a different structure (but in a complete graph). As happens in images, where scale invariance is key for the persistence of local descriptors, given the graph structure it should be desirable to select a different value of k for different nodes. However, translating the concept of scale analysis to the domain of graphs is quite computational demanding; we suggest a sort of optimal k selection using a Harris detector, but in graphs: define a set of features for each subgraph and track their variability until a peak in the node history is detected. Consequently, in this paper we will set experimentally a constant order k for all subgraphs. In this regard, it is convenient to choose a small constant order in comparison to $|V|$ in order to maximize the entropy of the subgraph distribution, that is, for providing more informative coverages.

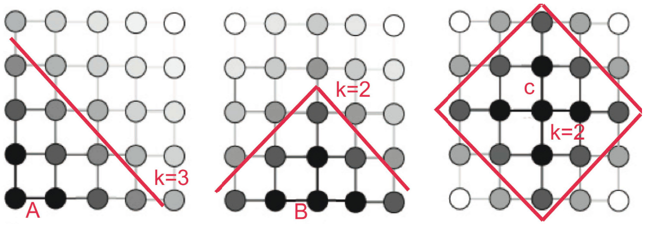


Fig. 4. Examples of partial node coverages in a 5 × 5 regular and 4-neighboring grid. Nodes with the same gray intensity have been added in the same order expansion. The lighter the intensity the higher the order. We consider nodes A ($k = 3$) (left), B ($k = 2$) and C ($k = 2$).

In the information-geometry approach followed here (see [17][18]), the features are covariance matrices relying on spectral descriptors. More precisely, the features are vectorized covariances projected on a given tangent space (exponential chart). Consider $\Phi(i) = (f_1(i), \ldots, f_d(i))^T$ a vector of spectral descriptors of the partial node coverage $H = e^k(i) \subseteq G$ (commute times, Fiedler vector, Perron-Frobenius vector and node centrality. Such descriptors have been determined to be very informative for graph discrimination [19]. For commute times (CT) we consider both the Laplacian and the normalized Laplacian of H: the elements of the upper off-diagonal elements of the CT kernel are downsampled to select $m = |V_H|$ elements and they are normalized by m^2. Fiedler and Perron-Frobenius vectors have m elements by definition. Node centrality is more selective than degree and it is related to the number of closed walks starting and ending at a each node. This measure is also normalized by m^2. *For each partial coverage* H we can compute the statistics of d spectral descriptors taking m samples (one sample per element of H). Such statistics can be easily encoded in a covariance matrix. In this paper we assume non-attributed graphs, but if the application domain imposes weights in the edges it is also possible to compute the spectral features described or referenced above.

The set of $d \times d$ covariance matrices $\boldsymbol{X}_i = \frac{1}{n-1}\sum_{i=1}^{m}(\Phi(i) - \boldsymbol{\mu})(\Phi(i) - \boldsymbol{\mu})^T$, being $m = |V_H|$, lie in a Riemannian manifold $\mathcal{M}$ (see Fig. 5). For each $\boldsymbol{X} \in \mathcal{M}$ there exists a neighborhood which can be mapped to a given neighborhood in $\mathbb{R}^{d\times d}$. Such mapping is continuous bidirectional and one-to-one. As a Riemann manifold is differentiable, the derivatives at each $\boldsymbol{X}$ always exist, and such derivatives lie in the so called tangent space $T_{\boldsymbol{X}}$, which is a vector space in $\mathbb{R}^{d\times d}$. The tangent space at $\boldsymbol{T_X}$ is endowed with an inner product $< ., . >_{\boldsymbol{X}}$ being $< \boldsymbol{u}, \boldsymbol{v} >_{\boldsymbol{X}} = trace(\boldsymbol{X}^{-\frac{1}{2}}\boldsymbol{u}\boldsymbol{X}^{-1}\boldsymbol{v}\boldsymbol{X}^{-\frac{1}{2}})$. The tangent space is also endowed with an exponential map $\exp_{\boldsymbol{X}} : T_{\boldsymbol{X}} \to \mathcal{M}$ which maps a tangent vector $\boldsymbol{u}$ to a point $\boldsymbol{U} = \exp_{\boldsymbol{X}}(\boldsymbol{u}) \in \mathcal{M}$. Such mapping is one-to-one, bidirectional and continuously differentiable and maps u to the point reached by the unique geodesic (minimum-length curve connecting two points in the manifold) from $\boldsymbol{X}$ to $\boldsymbol{U}$: $g(\boldsymbol{X}, \boldsymbol{U})$. The exponential map is only one-to-one in the neighborhood of $\boldsymbol{X}$ and this implies that the inverse mapping $\log_{\boldsymbol{X}} : \mathcal{M} \rightarrow T_{\boldsymbol{X}}$ is uniquely defined in a small neighborhood of $\boldsymbol{X}$. Therefore, we have the following mappings for going to the manifold and back (to the tangent space) respectively:

$$\exp_{\boldsymbol{X}}(\boldsymbol{u}) = \boldsymbol{X}^{\frac{1}{2}} \exp(\boldsymbol{X}^{-\frac{1}{2}} \boldsymbol{u} \boldsymbol{X}^{-\frac{1}{2}}) \boldsymbol{X}^{\frac{1}{2}}, \;\; \log_{\boldsymbol{X}}(\boldsymbol{U}) = \boldsymbol{X}^{\frac{1}{2}} \log(\boldsymbol{X}^{-\frac{1}{2}} \boldsymbol{U} \boldsymbol{X}^{-\frac{1}{2}}) \boldsymbol{X}^{\frac{1}{2}} \;, \tag{17}$$

and the corresponding geodesic between two tensors $\boldsymbol{X}$ and $\boldsymbol{U}$ in the manifold:

$$g^2(\boldsymbol{X}, \boldsymbol{U}) =< \log_{\boldsymbol{X}}(\boldsymbol{U}), \log_{\boldsymbol{X}}(\boldsymbol{U}) >_{\boldsymbol{X}} = trace\left(\log^2(\boldsymbol{X}^{-\frac{1}{2}} \boldsymbol{U} \boldsymbol{X}^{-\frac{1}{2}})\right) \;. \tag{18}$$

In all the definitions above we take the matrix exponentiation and logarithm.

Each graph X has $n_X = |V_X|$ partial coverages, one for each node. Therefore, we have n overlapped subgraphs H_{X_i} each one characterized by a covariance matrix $\boldsymbol{X}_i$ based on $m_{H_{X_i}} = |V_{H_{X_i}}|$ samples. Then, ech graph can be encoded by a population of n_X points in a manifold $\mathcal{M}$. For instance, another graph Y will be encoded by n_Y covariance matrices $\boldsymbol{Y}_j$ in the same manifold $\mathcal{M}$. In order to compare both populations we can map then back to a given tangent space. However we must determine what is the origin of such space. Let us denote by $\boldsymbol{Z_k}$ with $k = 1, \ldots, N$ (being $N = n_X + n_Y$) each covariance matrix coming from X or from Y. A fair selection of the tangent space origin is the Karcher mean defined as $\mu = \arg\min_{\boldsymbol{Z} \in \mathcal{M}} d^2(\boldsymbol{Z}_k, \boldsymbol{Z})$. The Karcher mean can be obtained after few iterations of $\mu^{t+1} = \exp_{\mu^t}(\bar{\boldsymbol{X}}^t)$ where $\bar{\boldsymbol{X}}^t = \frac{1}{N} \sum_{k=1}^{N} \log_{\mu^t}(\boldsymbol{Z}_k)$. Once we have μ, we have an origin for the tangent space, and then we can project all matrices $\boldsymbol{Z}_k$ in such space (see Fig. 5-left) through $\mathbf{Z}_k = \log_{\mu}(\boldsymbol{Z}_k)$.

Therefore, in the tangent space, whose origin is $\log_{\mu}(\mu) = \mathbf{0}$, we will have two distributions of tensors: $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_{n_x}\}$ and $\mathcal{Y} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_{n_y}\}$. In [20]

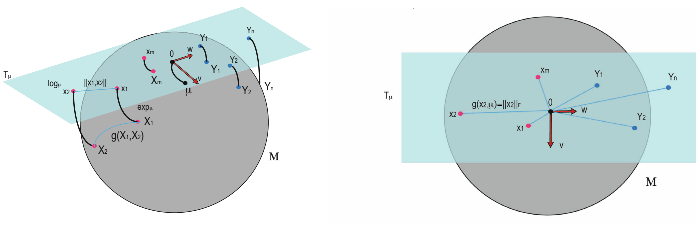


Fig. 5. Left: Riemannian manifold (the sphere) and tangent space T_μ at point μ. Points in the tangent space are the de-projections (log) of their corresponding projections (exp) which lie in the Manifold. We show in different colors points corresponding to subgraphs of two different graphs: X_i and Y_j. We also show some examples of distances in the manifold (geodesics) $g(.,.)$ and in the tangent space $||.,.||$ (vectorization), and tangents u and w. Right: Tangent space (from a zenithal view) with the geodesics $||.||_F$ (Frobenius norms) used for building the linear combination of Gaussians (the barycenter).

computing graph similarity is posed in terms of vectorizing each distribution and then compute a multi-dimensional divergence, the Henze-Penrose one [10], which is estimated by the Friedman-Rafsky test. Although this divergence outperforms our results obtained through entropic manifold alignment followed by the application of the SNESV divergence, such success is due to two facts: (i) in the node coverage each sugraph is characterized by highly discriminative spectral features, and (ii) the multi-dimensional Friedman-Rafsky test in this context is more structural than comparing entropies of two different manifolds. However, we argue herein that there is additional room for improvement: the Friedman-Rafsky test relies on computing minimum spanning trees, that is, on computing Euclidean distances between the vectorized tensors in the tangent space. Such distances are different from geodesics and consequently their use may imply a significant loss of metric information. Metric errors are attenuated by placing the tangent space at the Karcher mean, but they may exist. As each tensor in the tangent space defines a geodesic from the origin (pole) of the manifold, its Frobenius norm $||\mathbf{X}||_F = trace(\mathbf{X}\mathbf{X}^T)$ is coincident with its geodesic distance to the pole $\mu = \log_\mu(\mu) = \mathbf{0}$:

$$g(\mu, \boldsymbol{X}) = \left\|\log(\mu^{-\frac{1}{2}}\boldsymbol{X}\mu^{-\frac{1}{2}})\right\|_F = \left\|\log_\mu(\boldsymbol{X})\right\|_F = \|\mathbf{X}\|_F \ . \tag{19}$$

Therefore we can use the latter norms safely to build an error-free distributional representative (prototype) for each set of tensors (graph).

Given that the tensors in the tangent space are covariance matrices, they define zero mean d-dimensional Gaussian variables $\boldsymbol{x}_i$ (respectively $\boldsymbol{y}_j$) with pdf

$$p(\boldsymbol{x}_i; \mathbf{0}, \mathbf{X}_i) = \frac{1}{\sqrt{(2\pi)^d|\mathbf{X}_i|}} \exp\left(-\frac{1}{2}\boldsymbol{x}_i^T\mathbf{X}_i^{-1}\boldsymbol{x}_i\right) \ , \tag{20}$$

and similarly for $\boldsymbol{y}_j$. A simple way of *combining* or fusing several variables to define a prototype is to perform a linear combination:

$$\boldsymbol{c}_x = \sum_{i=1}^{n_x} a_i\boldsymbol{x}_i, \ \ \boldsymbol{c}_x \sim \mathcal{N}\left(\mathbf{0}, \sum_{i=1}^{n_x} a_i^2\mathbf{X}_i\right) \ , \boldsymbol{c}_y = \sum_{i=1}^{n_y} b_i\boldsymbol{y}_i, \ \ \boldsymbol{c}_y \sim \mathcal{N}\left(\mathbf{0}, \sum_{j=1}^{n_y} b_j^2\mathbf{Y}_j\right) \ , \tag{21}$$

where $a_i = \frac{1}{||\mathbf{X}_i||_F}$ and $b_j = \frac{1}{||\mathbf{Y}_j||_F}$ are the inverses of the Frobenius norms (distances to the origin) as we show in Fig. 5-right. The choice of the barycenter is, by far, more discriminative than the uniform weighting: $a_i = \frac{1}{n_x}$, $b_i = \frac{1}{n_y}$. In addition, using the latter inverse Frobenius coefficients we tend to non-trivially minimize the entropy of the prototype: large (distant) covariances contribute less to the linear combination than smaller (close) ones. In any case, neither the variables of the linear combination nor the resulting prototypes lie in the tangent space; we exploit them to focus on the resulting covariances because in the Gaussian case, the entropy relies only on the covariance of the distribution [21].

3.2 Bregman and Total Bregman Divergences

The *Bregman divergence* [13] d_f associated with a real valued strictly convex and differentiable function f defined on a convex set X between points $x, y \in X$ is given by,

$$d_f(x,y) = f(x) - f(y) - \langle x - y, \nabla f(y) \rangle, \tag{22}$$

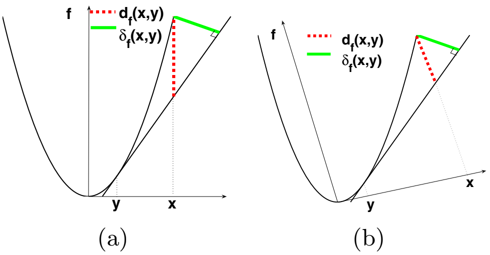
where $\nabla f(y)$ is the gradient of f at y and $\langle \cdot, \cdot \rangle$ is the inner product determined by the space on which the inner product is being taken.

For example, if $f : \mathbb{R}^n \to \mathbb{R}$, then $\langle \cdot, \cdot \rangle$ is just the inner product of vectors in $\mathbb{R}^n$, and $d_f(\cdot, y)$ can be seen as the distance between the first order Taylor approximation to f at y and the function evaluated at x. Bregman divergence d_f is non-negative definite and does not satisfy the triangular inequality thus making it a divergence. As shown in Fig. 6, Bregman divergence measures the ordinate distance, the length of the dotted red line which is parallel to the y-axis. It is dependent on the coordinate system, for example, if we rotate the coordinate system, the ordinate distance will change (see the dotted lines in Fig. 6(a) and (b)). This coordinate dependent distance has great limitations because it requires a fixed coordinate system, which is unrealistic in the cases where a fixed coordinate system is difficult to build. With the motivation to overcome this shorting and release the freedom of choosing coordinate systems, we proposed total Bregman divergence.

The *total Bregman divergence* [14] δ_f associated with a real valued strictly convex and differentiable function f defined on a convex set X between points $x, y \in X$ is defined as,

$$\delta_f(x,y) = \frac{f(x) - f(y) - \langle x - y, \nabla f(y) \rangle}{\sqrt{1 + \|\nabla f(y)\|^2}}, \tag{23}$$

$\langle \cdot, \cdot \rangle$ is inner product as in the definition of Bregman divergence, and $\|\nabla f(y)\|^2 = \langle \nabla f(y), \nabla f(y) \rangle$ generally.



(a) (b)

Fig. 6. In each figure, $d_f(x,y)$ (dotted red line) is BD, $\delta_f(x,y)$ (bold green line) is TBD, and the two arrows indicate the coordinate system. (a) shows $d_f(x,y)$ and $\delta_f(x,y)$ before rotating the coordinate system. (b) shows $d_f(x,y)$ and $\delta_f(x,y)$ after rotating the coordinate system. Note that $d_f(x,y)$ changes with rotation unlike $\delta_f(x,y)$ which is invariant to rotation.

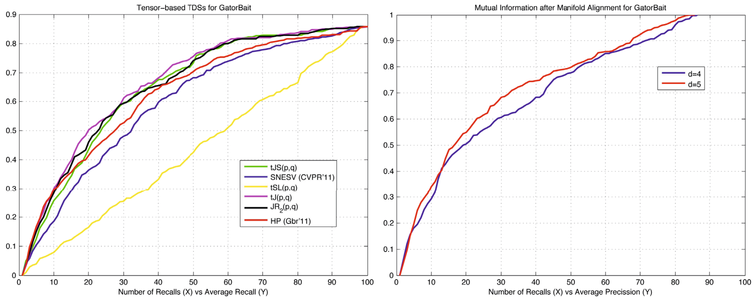


Fig. 7. Average recall (Y axis) vs number of retrievals (X axis). Left: Curves for tensor-based divergences. Right: Curves for (estimated) Mutual Information between aligned manifold.

As shown in Fig. 6, TBD measures the orthogonal distance, and if we translate or rotate the coordinate system, $\delta(\cdot,\cdot)$ will not change. Also $\delta_f(\cdot, y)$ can be seen as a higher order "Taylor" approximation to f at y and the function evaluated at x. Then

$$\delta_f(x,y) = d_f(x,y) - \frac{\|\nabla f(y)\|^2}{2} d_f(x,y) + O(\|\nabla f(y)\|^4) \tag{24}$$

where $O(\cdot)$ is the Big O notation, which is usually small compared to the first term and thus one can ignore it without worrying about the accuracy of the result. Also, we can choose the higher order "Taylor" expansion if necessary.

Compared to the BD, TBD contains a weight factor (the denominator) which complicates the computations. However, this structure brings up many new and interesting properties and makes TBD an "adaptive" divergence measure in many applications. Note that, in practice, X can be an interval, the Euclidean space, a d-simplex, the space of non-singular matrices or the space of functions[15]. For instance, in the application to shape representation, we let p and q be two pdfs, and $f(p) := \int p \log p$, then $\delta_f(p,q)$ becomes what we will call the total Kullback-Leibler divergence (tKL).

Jeffreys divergence is a symmetrized version of the Kullback-Leibler Divergence. Let p and q two pdfs defined respectively by prototypes $\boldsymbol{c}_x$ and $\boldsymbol{c}_y$, where we set $\Sigma_x = \sum_{i=1}^{n_x} a_i^2 \mathbf{X}_i$ and $\Sigma_y = \sum_{j=1}^{n_y} b_j^2 \mathbf{Y}_i$, then the Jeffreys divergence between them is given by

$$J(p,q) = \frac{1}{2}\left(trace(\Sigma_y^{-1}\Sigma_x) + trace(\Sigma_x^{-1}\Sigma_y\right) - d\ . \tag{25}$$

Therefore, the definition of the $tJ(p,q)$ (Jeffreys TBD) depends on the definition of $tLK(.,.)$ (the KL TBD):

$$\begin{aligned} tJ(p,q) = tKL(p,q) + tKL(q,p) = \frac{\log\left|\Sigma_x^{-1}\Sigma_y\right| + trace(\Sigma_y^{-1}\Sigma_x) - d}{2\sqrt{2(1-H(q))}} + \\ + \frac{\log\left|\Sigma_y^{-1}\Sigma_x\right| + trace(\Sigma_x^{-1}\Sigma_y) - d}{2\sqrt{2(1-H(p))}}\ , \end{aligned} \tag{26}$$

being $H(.)$ the entropy: $H(p) = \log \sqrt{(2\pi e)^d |\Sigma_x|}$ and similarly for $H(q)$. This divergence (tJ) is the most discriminative for GatorBait, outperforming SNESV and many other TBD divergences (see Fig. 7-left). It is highly competitive with total Jensen-Shannon (tJS) and the quadratic Jensen-Rényi (JR2). These latter divergences outperform the Henze-Penrose divergence applied to vectorization of covariance matrices (HP). These good results are both due to a change of representation (which contributes to break iso-spectrality) and the use of novel divergences (total Bregman ones). However, the computational cost of building a node coverage given a fixed order ($k = 5$ in this paper) is $O(n^3 \times k)$ where $n = |V|, k << n$. Then we must compute the spectra for the features ($O(n^3)$), the covariance matrices, the projection in the tangent space (product of several matrices including the matrix logarithm of the product of 3 matrices which also implies a spectral decomposition; then we have $O(n^8)$ (for each tensor) and the Karcher mean (several iterations, each one involving $N = n_x + n_y$ matrix logarithms taking $O(n^3)$ each). After that we compute the divergence. Therefore the global complexity of tensor representation is dominated by $O(n^8)$. On the other hand, when we use the entropic manifold alignment, the cost of making the alignment can be linear per iteration. Then, the embedding takes $O(n^3)$ for computing the spectral decomposition and then a product of matrices, and this is done once (before alignment). Estimating the entropy is $O(n^2 + n \log n)$. In any case, entropic alignment is more efficient than tensor-based methods but these latter ones are more discriminative. Is it possible to improve discriminability without increasing also the computational cost? Fortunately there is an intermediate method relying on manifold alignment but changing the divergence. This new divergence is an approximation of multi-dimensional *mutual information* using the estimator described in [22]. This estimator relies on computing copulas which require a sorting for each dimension and then a kNN estimator of the Rényi entropy (quadratic). If we have $d-$dimensional samples, we must estimate a $2d$ joint entropy and 2 $d-$dimensional ones (unfortunately we do not have space in this paper to detail this method and we have only preliminary results and insights). As we show in Fig. 7-right, for $d = 5$ in the embedding mutual information outperforms tJ, the best tensor-based divergence.

4 Conclusions

The main contribution of this paper is to explore the link between graph representations and divergences. Future work includes the development of mutual information estimators jointly with optimal alignment.

Acknowledgements. F. Escolano is supported by the project TIN2012-32839 of the Spanish Government. M.A. Lozano is supported by the University of Alicante research grant GRE10-21.E.R. Hancock is supported by a Royal Society Wolfson Research Merit Award.

References

1. Escolano, F., Suau, P., Bonev, B.: Information Theory in Computer Vision and Pattern Recognition. Springer, New York (2009)
2. Qiu, H., Hancock, E.R.: Clustering and Embedding using Commute Times. IEEE Trans. on PAMI 29(11), 1873–1890 (2007)
3. Escolano, F., Hancock, E.R., Lozano, M.A.: Graph matching through entropic manifold alignment. In: Proc. of CVPR 2011, pp. 2417–2424 (2011)
4. Escolano, F.: Hancock: From Points to Nodes: Inverse Graph Embedding through a Lagrangian Formulation. In: Proc. of CAIP (1) 2011, pp. 194–201 (2011)
5. Myronenko, A., Song, X.B.: Point-Set Registration: Coherent Point Drift. IEEE Trans. on PAMI 32(12), 2262–2275 (2010)
6. Costa, J.A., Hero, A.O.: Geodesic Entropic Graphs for Dimension and Entropy Estimation in Manifold Learning. IEEE Transactions on Signal Processing 52(8), 2210–2221 (2004)
7. Chen, T., Vemuri, B.C., Rangarajan, A., Eisenschenk, S.J.: Group-Wise Point-Set Registration Using a Novel CDF-Based Havrda-Charvát Divergence. International Journal of Computer Vision 86(1), 111–124 (2010)
8. Martins, A., Smith, N., Xing, E., Aguiar, P., Figueiredo, M.: Nonextensive Information Theoretic Kernels on Measures. Journal of Machine Learning Research 10, 935–975 (2009)
9. Leonenko, N., Pronzato, L., Savani, V.: A class of Rényi Information Estimators for Multidimensional Densities. Annals of Statistics 36(5), 2153–2182 (2008)
10. Henze, N., Penrose, M.: On the multi-variate runs test. Annals of statistics 27, 290–298 (1999)
11. Friedman, J.H., Rafsky, L.C.: Mutivariate Generalization of the Wald-Wolfowitz and Smirnov Two-Sample Tests. Annals of Statistics 7(4), 697–717 (1979)
12. Stowell, D., Plumbley, M.D.: Fast Multidimensional Entropy Estimation by K-d Partitioning. IEEE Signal Processing Letters 16(6), 537–540 (2009)
13. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with Bregman Divergences. J. Mach. Learn. Res. 6, 1705–1749 (2005)
14. Liu, M., Vemuri, B., Amari, S.-I., Nielsen, F.: Total Bregman Divergence and its Applications to Shape Retrieval. In: Proc. of CVPR 2010, pp. 3463–3468 (2010)
15. Frigyik, B.A., Srivastava, S., Gupta, M.R.: Functional Bregman Divergence. Int. Symp. Inf. Theory 54, 5130–5139 (2008)
16. Escolano, F., Hancock, E.R., Lozano, M.A.: Heat Diffusion: Thermodynamic Depth Complexity of Networks. Physical Review E 85, 036206 (2012)
17. Tuzel, O., Porikli, F., Meer, P.: Pedestrian Detection via Classification on Riemannian Manifolds. IEEE Trans. on PAMI 30(10), 1–15 (2008)
18. Pennec, X., Fillard, P., Ayache, N.: A Riemannian Framework for Tensor Computing. IJCV 38(1), 41–66 (2006)
19. Bonev, B., Escolano, F., Giorgi, D., Biasotti, S.: Information-theoretic Selection of High-dimensional Spectral Features for Structural Recognition. Computer Vision and Image Understanding 117(3), 214–228 (2013)
20. Escolano, F., Bonev, B., Lozano, M.A.: Information- geometric Graph Indexing from Bags of Partial Node Coverages. In: Jiang, X., Ferrer, M., Torsello, A. (eds.) GbRPR 2011. LNCS, vol. 6658, pp. 52–61. Springer, Heidelberg (2011)
21. Escolano, F., Liu, M., Hancock, E.R.: Tensor-based Total Bregman Divergences between Graphs. In: Proc. ICCV Workshops 2011, pp. 1440–1447 (2011)
22. Pál, D., Póczos, B., Szepesvári, C.: Estimation of Rényi Entropy and Mutual Information Based Generalized Nearest-Neighbor Graphs. In: Proc. NIPS (2010)

Information Theoretic Pairwise Clustering

Avishay Friedman and Jacob Goldberger

Engineering Faculty, Bar-Ilan University, Ramat-Gan 52299, Israel

Abstract. In this paper we develop an information-theoretic approach for pairwise clustering. The Laplacian of the pairwise similarity matrix can be used to define a Markov random walk on the data points. This view forms a probabilistic interpretation of spectral clustering methods. We utilize this probabilistic model to define a novel clustering cost function that is based on maximizing the mutual information between consecutively visited clusters of states of the Markov chain defined by the graph Laplacian matrix. The algorithm complexity is linear on sparse graphs. The improved performance and the reduced computational complexity of the proposed algorithm are demonstrated on several standard datasets.

1 Introduction

Effective automatic grouping of objects into clusters is one of the fundamental problems in machine learning and in other fields of study. In many approaches, the first step toward clustering a dataset is extracting a feature vector from each object. This reduces the problem to the aggregation of groups of vectors in a feature space. A commonly used algorithm in this case is the k-means. However, in many cases we are only given pairwise similarity information between data points. For example, in social networks, only binary neighborhood relations are given. In these cases k-means cannot be applied in a straightforward way. Instead, we seek for a partition of the data based on the similarity measure between the points. Out of the numerous clustering algorithms, spectral clustering [14,16] has gained considerable attention in recent years due to its strong performance on arbitrary shaped clusters, and its well-defined mathematical framework [20].

Another family of clustering algorithms, that are derived from information-theory concepts, corresponds to the case of distributional clustering. Here each data point is described as a distribution. This situation is illustrated by the generic example of document clustering based on word histograms [18],[17]. In this case, the mutual information between word occurrences and clusters of documents is a natural clustering criterion [19] [4] that has been proven to be powerful in many cases. The information-theoretical principle described above is only applicable when a feature distribution, associated with each data point, is provided as part of the problem setup. In this paper we extend the mutual information clustering criterion to the domain of pairwise clustering. The probabilistic interpretation of spectral clustering, based on a Markov random walk, is used to associate a distribution with each data point via the corresponding

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 106–119, 2013.

conditional distribution row in the Markov transition matrix. In particular, we define a random walk on the data points and maximize the mutual information between cluster labels of data-points that are visited during the random walk. We show that this results in an efficient clustering method with state-of-the-art performance on real-world datasets. The remainder of this paper is organized as follows. Section 2 defines the notation of similarity graphs and the associated Laplacian matrix. Section 3 describes the minimum information-loss criterion for clustering the Markovian random-walk states. Section 4 introduces the Information-Theoretic Pairwise Clustering (ITPC) algorithm. Section 5 reviews related work and Section 6 describes numerical experiments on several standard datasets.

2 Similarity Graphs and Random Walks

Given a set of data points $x_1, ..., x_n$ and some symmetric notion of similarity $w_{ij} \geq 0$ between all pairs of data points x_i and x_j, the goal of clustering is to divide the data points into several groups such that points in the same group are similar and points in different groups are dissimilar to each other. In the common case where the data points live in the Euclidean space R^d, a reasonable candidate for a similarity measure is the Gaussian function $w_{ij} = \exp(-\|x_i - x_j\|^2/(2\sigma^2))$ (where the parameter σ controls the width of the local neighborhoods). Ultimately, the choice of the similarity function depends on the domain the data come from and the specific clustering task.

In the case where we have information in the form of similarities between data points, we can represent the data as a similarity graph $G = (V, E)$. Each vertex i in this graph represents a data point x_i. Two vertices are connected if the similarity w_{ij} between the corresponding data points x_i and x_j is positive and the edge is weighted by w_{ij}. The problem of clustering can now be reformulated using the similarity graph: we want to find a partition of the graph in which existing edges between different groups have low weights and edges within a group have high weights.

Denote the similarity matrix by $W = (w_{ij})$. The degree of a vertex $i \in V$ is defined as $d_i = \sum_{j=1}^{n} w_{ij}$. The degree matrix D is defined as the diagonal matrix with the degrees $d_1, ..., d_n$ on the diagonal. The normalized Laplacian matrix L is defined as $L = I - D^{-1}W$ [1]. (Note that in the literature there is no unique convention as to which matrix exactly is called “Graph Laplacian” [20].) All variants of the spectral clustering algorithm are based on using eigenvectors of the Laplacian matrix to represent the abstract data points as points in the Euclidean space. The clusters can be then obtained by applying simple clustering algorithms such as k-means in the embedded space [14,16,22]. The matrix $P = D^{-1}W = I - L$ is a stochastic matrix (non-negative entries, row sums are all 1). Using $n \times n$ transition matrix P we can define a stationary Markov chain that corresponds to a random walk on the graph nodes. Let $X = \{X_t\}$

be the n-valued stationary Markov chain defined by:

$$P_{ij} = (D^{-1}W)_{ij} = p(X_2 = j|X_1 = i) = \frac{w_{ij}}{\sum_k w_{ik}} \tag{1}$$

The transition probability P_{ij} of jumping in one step from i to j is proportional to the edge weight w_{ij}. Let $\pi = (\pi_1, ..., \pi_n)$, where $\pi_i = d_i/(\sum_j d_j)$. It can be easily verified that $P^\top \pi = \pi$. Hence, if the graph is connected and non-bipartite, then π is the unique stationary distribution of the Markov chain defined by P [20]. Therefore, the joint stationary probability of X_1 and X_2 is:

$$p(X_1 = i, X_2 = j) = \frac{w_{ij}}{\sum_{kl} w_{kl}}. \tag{2}$$

Given the random walk model (1) we can translate the pairwise clustering problem, into the problem of clustering the states of a Markov chain.

3 Clustering the States of a Markov Chain

Let $\{A_1, ..., A_m\}$ be a partition of the states $\{1, ..., n\}$ into m clusters and let C denote the subset membership function, i.e. $C(i) = j$ if $i \in A_j$. For each t we define a random variable $Y_t = C(X_t)$ indicating the cluster membership of the state visited by the random walk at time t. The joint distribution of the random variables (Y_1, Y_2) defined on the clusters is:

$$\begin{aligned} p(Y_1 = i, Y_2 = j) &= p(X_1 \in A_i, X_2 \in A_j) \qquad (3) \\ &= \frac{1}{\mathrm{vol}(V)} \sum_{k \in A_i, l \in A_j} w_{kl} \end{aligned}$$

such that $\mathrm{vol}(V) = \sum_{ij} w_{ij}$. The model is illustrated by the following diagram:

$$\begin{tikzcd}
X_1 \arrow[r] \arrow[d, "C"'] & X_2 \arrow[r] \arrow[d, "C"'] & X_3 \arrow[r] \arrow[d, "C"'] & ... \arrow[d] \\
Y_1 & Y_2 & Y_3 & ...
\end{tikzcd}$$

Each clustering $\{A_1, ..., A_m\}$ induces a joint distribution $p(Y_1, Y_2)$ on the clusters visited on consecutive time units. To find the best clustering based on the joint distribution of Y_1 and Y_2, we need to extract from the $m \times m$ matrix $(p(Y_1 = i, Y_2 = j))$ a single number that measures the clustering quality. Once decided on a suitable clustering score, we can find the clustering that optimizes this score.

An intuitive clustering score, that we would like to minimize, is:

$$p(Y_2 \neq Y_1) = \sum_{i=1}^{m} p(Y_2 \neq i|Y_1 = i)p(Y_1 = i) \tag{4}$$

which is the probability that consecutive visited points would be in different clusters. However, the clustering that minimizes criterion (4) is the one formed by a single cluster that contains all the data points. Even if we enforce that all the m clusters should be non-empty, the score (4) still favors clusterings that are very unbalanced. To overcome this degeneracy we can modify the clustering score we minimize as follows:

$$\text{Ncut}(A_1, ..., A_m) = \sum_{i=1}^{m} p(Y_2 \neq i | Y_1 = i) \tag{5}$$

where Ncut is the Normalized-Cut score of the partition $C = \{A_1, ..., A_m\}$. Meila and Shi [12] showed that the Ncut spectral clustering algorithm [16] [21] is an algorithm that finds an optimal solution for a relaxation of the Ncut criterion (5) for clustering the states of the random walk defined by the Laplacian of affinity graph. Dhillon et el. [3] applied kernel k-means (with kernel $K = D^{-1}WD^{-1}$) to directly optimize the Ncut score.

In this study we suggest to apply the information-theoretical principle of minimal information loss to cluster the states of the random walk. The mutual information induced by the clustering $C = \{A_1, ..., A_m\}$ is:

$$MI(A_1, ..., A_m) = I(Y_1; Y_2) = \tag{6}$$

$$\sum_{ij} p(Y_1 = i, Y_2 = j) \log \frac{p(Y_1 = i, Y_2 = j)}{p(Y_1 = i)p(Y_2 = j)}.$$

The original walk over the points also determines a walk over the clusters. The goal of clustering is to choose the clustering such that the loss in mutual information due to clustering is minimized. A good Markov-state clustering should preserve maximum information on the visited points. Using the mutual information criterion, the best clustering of the given n points into m clusters is the one that minimizes the information loss of the mutual information $I(X_1; X_2) - I(Y_1; Y_2)$ over all the partitions of the state-space into m subsets. The definition of mutual information implies that:

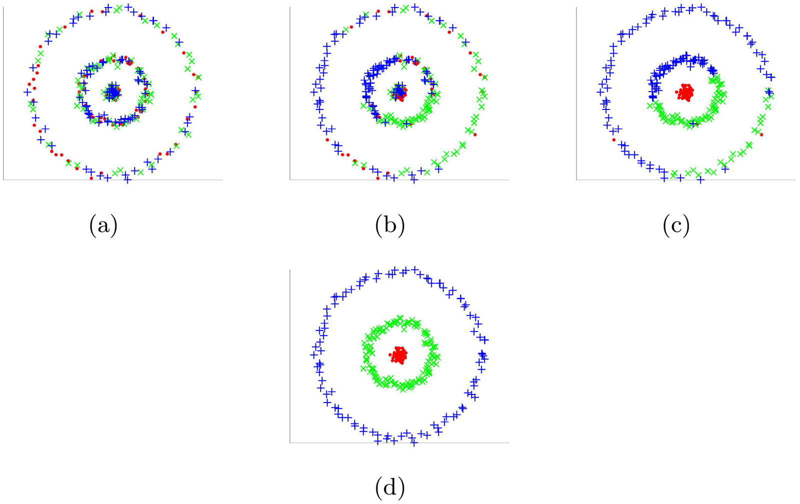
$$I(Y_1; Y_2) = H(Y_2) - \sum_{i=1}^{m} H(Y_2 | Y_1 = i) p(Y_1 = i) \tag{7}$$

When maximizing $I(Y_1; Y_2)$ the first term of (7) encourages clusters to have similar sizes and the second term discourages the random walk from jumping from cluster to cluster.

Utilizing standard information-theory manipulations we can derive several equivalent forms for the information loss function we want to minimize.

$$\begin{aligned} \text{score}(C) &= I(X_1; X_2) - I(Y_1; Y_2) \qquad (8) \\ &= D(p(X_1, X_2) \| p(Y_1, Y_2) p(X_1 | Y_1) p(X_2 | Y_2)) \\ &= H(Y_1, Y_2) + H(X_1 | Y_1) + H(X_2 | Y_2) - H(X_1, X_2) \\ &= D(p(X_2 | X_1) \| p(X_2 | Y_1)) + D(p(Y_1 | X_2) \| p(Y_1 | Y_2)) \end{aligned}$$

where $Y_1 = C(X_1)$, $Y_2 = C(X_2)$, D is the Kullback-Leibler divergence and H is the entropy function [2]. The optimal state-clustering is the one that minimizes the information-loss function score(C). Note that the information-theoretic equality (8) is correct for clustering the states of a general Markov chain. In our case, because the similarity matrix is symmetric, the Markov chain is also reversible.


(a)

(b)

(c)

(d)

Fig. 1. The steps of the ITPC algorithm on a three-circles data set. (a) random initializing, (b),(c) intermediate results, (d) final results (obtained after two passes over the data points).

Following [14], to understand the cost function we optimize, it is instructive to consider its behavior in the "ideal" case in which all points in different clusters are infinitely far apart and the m clusters are equal in shape. In this case the joint cluster distribution $(C(X_1), C(X_2))$ of the correct clustering is the $m \times m$ scalar matrix $\frac{1}{m}I$. Hence, for the correct clustering $H(C(X_1)) = \log(m)$ and $H(C(X_2)|C(X_1)) = 0$ and therefore, $I(C(X_1); C(X_2)) = \log(m)$. However, for any joint distribution (U, V) on $m \times m$ elements we have: $I(U; V) = H(U) - H(U|V) \leq H(U) \leq \log(m)$. Hence, the mutual information score $I(C(X_1); C(X_2))$ of the correct clustering is maximal.

4 The Clustering Algorithm

There is no closed-form solution for the minimal information-loss criterion stated in the previous section. Several standard optimization algorithms can be utilized to find the best clustering. In this study we apply a greedy sequential algorithm (see e.g. [17]). The sequential greedy algorithm has been found to perform well in terms of both clustering quality and computational complexity. The sequential clustering algorithm starts with a random clustering of the n graph nodes into m clusters. We then go over the data points in a circular manner and check for each point whether its removal from one cluster to another can reduce the information loss. This loop is iterated until no single-point transition offers an improvement. Since there is no guarantee that the algorithm will find the global optimum, we can run the algorithm on several initial random partitions and choose the best local optimum. Alternatively we can use a multi-level clustering approach [9].

The basic step of this algorithm is computing the information loss caused by merging a singleton cluster into an existing cluster. More generally we can define a distance measure between two clusters as the information-loss caused by merging the two clusters into a single one; i.e. the difference between the mutual information before and after the two clusters are merged. Direct computation of $I(Y_1;Y_2)$ requires $O(m^2)$ operations where m is the number of clusters. We next show that we can efficiently compute the information loss caused by merging two clusters in a time that is linear in the number of clusters.

Assume we are given a data partition $\{A_1,...,A_m\}$ and we want to compute the information loss caused by merging the clusters A_1 and A_2 to obtain a new partition $\{A_1 \cup A_2, A_3,...,A_m\}$ composed of $m-1$ clusters. Let Y_1 and Y_2 be the cluster membership random variable associated with the original clustering $\{A_1,...,A_m\}$ and $\hat{Y}_1$ and $\hat{Y}_2$ are the cluster membership random variables associated with the clustering after merging A_1 and A_2 into a single cluster. The following formula provides an efficiently computed expression for the information loss caused by the merging:

$$d(A_1,A_2) = I(Y_1;Y_2) - I(\hat{Y}_1;\hat{Y}_2) \tag{9}$$

$$= 2\sum_{i=1}^{2}\sum_{j=1}^{m} p(Y_1 = i, Y_2 = j) \log \frac{p(Y_2 = j|Y_1 = i)}{p(Y_2 = j|Y_1 \in \{1,2\})}$$

$$- \sum_{i=1}^{2}\sum_{j=1}^{2} p(Y_1 = i, Y_2 = j) \log \frac{p(Y_2 = j|Y_1 = i)}{p(Y_2 \in \{1,2\}|Y_1 \in \{1,2\})}$$

$$= 2p(Y_1 \in 12) JS(p(Y_2|Y_1 = 1)||p(Y_2|Y_1 = 2))$$

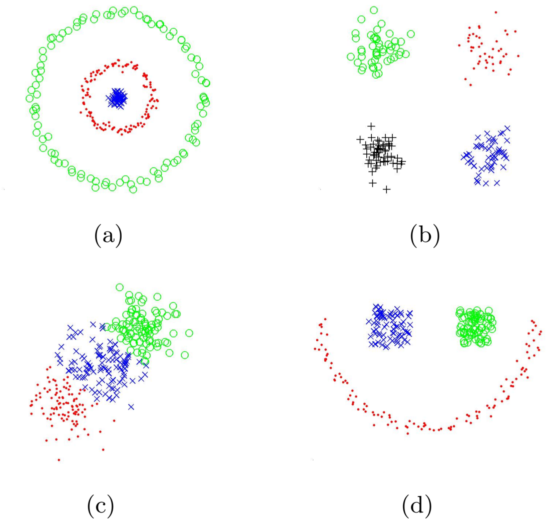
$$-p(Y_1 \in 12, Y_2 \in 12) I(Y_1;Y_2|Y_1 \in 12, Y_2 \in 12)$$

such that JS is the Jensen-Shannon divergence [2] and '12' is an abbreviation for $\{1,2\}$. The equality follows from the fact that the joint distributions of (Y_1,Y_2) and $(\hat{Y}_1;\hat{Y}_2)$ are very similar. For every i,j that are both larger than 2 we have $p(Y_1 = i, Y_2 = j) = p(\hat{Y}_1 = i, \hat{Y}_2 = j)$. Hence, most terms in the difference $I(Y_1;Y_2) - I(\hat{Y}_1;\hat{Y}_2)$ are canceled and the distance measure $d(A_i,A_j)$ (9) can be computed in $O(m)$ operations where m is the number of clusters. The sequential clustering algorithm requires the computation of the change in the cost function when moving a point from one cluster to another. This can be efficiently done using expression (9).

Table 1. The Information-Theoretic Pairwise Clustering (ITPC) algorithm

```
Input: A similarity graph defined by the $n \times n$ weight matrix W.
Output: A partition of the graph vertices into $m$ clusters.

Algorithm:

  1. Convert the graph into a Markov chain:
         $\widetilde{w}_{ij} \triangleq p(X_1 = i, X_2 = j) = \frac{w_{ij}}{\sum_{kl} w_{kl}}$
  2. Choose a random partition $A_1, ..., A_m$ of the Markov states and compute the
     cluster distribution $m \times m$ matrix:
         $q_{ij} = p(Y_1 = i, Y_2 = j) = p(X_1 \in A_i, X_2 \in A_j)$.
  3. Loop until there is no change
       – for $i = 1, ..., n$ move state $i$ into the cluster that minimizes the information
         loss.
           • Remove state $i$ from its current cluster.
           • for $j = 1, ..., m$
               ∗ Add state $i$ to cluster $A_j$ and compute $d(\{i\}, A_j)$ (see Eq. (9)).
           • Choose the cluster which minimize the information-loss.

Removing/Adding state $i$ from/to cluster $A_j$ in a constant time (assuming
each node has at most $k$ neighbors):

  – Go over all $s \in$neighbors of node $i$
      • Assume $s$ is in cluster $A_l$.
      • $q_{jl} \leftarrow q_{jl} - \widetilde{w}_{is}$   /   $q_{jl} \leftarrow q_{jl} + \widetilde{w}_{is}$
      • $q_{lj} \leftarrow q_{lj} - \widetilde{w}_{is}$   /   $q_{lj} \leftarrow q_{lj} + \widetilde{w}_{is}$
```

The computational complexity of the proposed clustering algorithm is as follows. To recompute the joint distribution of (Y_1, Y_2) after moving a point i from one cluster to another we need to go over all weights on edges connected to i. Hence, it takes $O(n)$ for the basic step of searching all possible cluster memberships of a given data point. Assuming a fixed number of iterations over the dataset, the complexity is $O(n^2)$. In the (usual) case where the graph is sparse and each point is connected to at most k neighbors, the number of operations needed to recompute the clustering joint distribution, after moving a point from one cluster to another, is bounded by k. Hence, the computational complexity for sparse graphs is linear in the size of the dataset n. Note that when using spectral clustering methods, finding the eigenvectors of a large matrix is computationally costly. It takes $O(n^3)$ in general, and even with fast approximating techniques vast amount of space and time are required for larger datasets. We dub the proposed algorithm "Information-Theoretic Pairwise Clustering" (ITPC).



(a) (b)

(c) (d)

Fig. 2. Clustering of several synthetic datasets by ITPC (using Euclidean knn graph)

The linear time implementation of the ITPC algorithm is summarized in Table 1. An example of applying the sequential procedure on a synthetic dataset is shown in Figure 1.

One drawback of the sequential algorithm (in contrast to agglomerative approaches) is that the number of clusters must be given as input to the algorithm. In case we do not know the exact number of clusters we can slightly modify the algorithm in such a way that we can simply provide a rough estimation (upper bound) on the number of desired clusters. Consider the case of a cluster that contains a single object i. The iterative-sequential algorithm will not merge i into any other cluster because obviously this cannot increase the cost function $I(Y_1; Y_2)$. The algorithm will always prefer to leave i as a single member of a cluster. In the modified version we enforce a singleton cluster to be merged into another cluster. More generally if a cluster size is less than a predefined number, we enforce the cluster's members to be merged into other clusters. This step reduces the number of clusters by one. Utilizing this scheme, the number of output clusters can be adapted to the data.

5 Related Work

Information-theoretic approaches have been intensively used for data clustering algorithms. The standard problem setup is based on a given joint distribution of objects and features denoted by the random variables X_1 and X_2 respectively.

A one-sided clustering of the object set X_1, denoted by $C(X_1)$, aims to maximize the mutual information $I(C(X_1);X_2)$ between the object clusters and the features [19,17]. A co-clustering (aka two-sided clustering) applies a clustering procedure on both the objects set and the feature set. Denote the object clustering by $C_1(X_1)$ and the feature clustering by $C_2(X_2)$. The best co-clustering is the one that maximizes the mutual information between the object clusters and the feature clusters $I(C_1(X_1);C_2(X_2))$ [5]. Note that in the co-clustering setup the object set and the feature set are different and therefore the object clustering and the feature clustering are different. In our setup of pairwise clustering the objects set and the feature set are the same and therefore by clustering the objects we automatically also cluster the features. The two random variables X_1 and X_2 correspond to two instances of the same set and the *same* clustering function is *simultaneously* applied to the two random variables X_1 and X_2. The target is to find a clustering C such that the mutual information $I(C(X_1);C(X_2))$ is maximized. The three clustering cases are illustrated bellow:

$$Y_1 \xleftarrow{C} X_1 \longleftrightarrow X_2 \qquad \text{one-sided} \tag{10}$$

$$Y_1 \xleftarrow{C_1} X_1 \longleftrightarrow X_2 \xrightarrow{C_2} Y_2 \qquad \text{two-sided} \tag{11}$$

$$Y_1 \xleftarrow{C} X_1 \longleftrightarrow X_2 \xrightarrow{C} Y_2 \qquad \text{simultaneous} \tag{12}$$

Sequential optimization algorithm has been applied for one-sided clustering [17]. In that case if the number of features is kept fixed, the algorithm is linear in the number of data points. The basic step of the sequential algorithms is finding the best cluster assignment for a given point. This step requires computing the Jensen-Shannon (JS) divergence between the cluster and the point. Computing the JS divergence is linear in the number of features. Hence, in our case of pairwise clustering, where the number of features is equal to the number of data points, the complexity of the one-sided algorithm is quadratic in the data size. Note that even if the graph is sparse and therefore the distribution corresponds to each object is sparse, the cluster distribution is not necessarily sparse. Hence, the complexity of the one-sided clustering algorithm [17], applied to pairwise clustering problem, is quadratic. When applying a sequential optimization to the case of co-clustering (11), we need to iterate between feature clustering given the object clusters and object clustering given the feature clusters [5]. In contrast to previous methods, the complexity of the proposed ITPC algorithm when applied to sparse pairwise clustering is linear and there is no need to iterate between feature clustering and object clustering. An information theoretic clustering approach of the states of a general Markov chain has been suggest in [7]. Unlike our algorithm whose complexity is linear (on sparse graphs), the complexity of their algorithm is quadratic in the dataset size. Another iterative bipartition algorithm which uses JS divergence as the statistical dissimilarity measure has been suggest in [6].

Spectral clustering algorithms [14] [16], are based on finding a low dimensional embedding using eigenvector computation which can be slow. The Power Iteration Clustering (PIC) [10] is a variant of spectral clustering that directly finds the low-dimensional embedding. Graclus [3] is another efficient graph clustering algorithm that is based on directly optimize the Ncut score using multilevel kernel k-means and avoids the eigenvector computations. The main difference between our algorithm and the Graclus algorithm [3] is the cost function that is being optimized. We optimize the mutual information score (6) while Graclus optimizes the Ncut score (5). Another minor difference is that Graclus uses a batch version of the kernel k-means which is not guaranteed to converge if the kernel is not positive definite. We use a sequential greedy algorithm which monotonically improves the cost function and therefore always converges to a local optimum.

6 Experimental Results

In this section, we demonstrate our proposed ITPC method on the following commonly used real-world datasets: **Iris** contains flower petal and sepal measurements from three species of irises, 150 instances. **Glass** has 214 instances separated into six classes of glass. **Wine** are the results of a chemical analysis of wines. The analysis determined the quantities of 13 constituents found in each of three types of wines. 178 instances. **Wisconsin Diagnostic Breast Cancer (WDPC)** has 359 instances separated into two classes. Each instance has 30 continuous features. Features are computed from a digitized image of a fine needle aspiration (FNA) of a breast mass. **Olivetti Faces (OlFace5)** 10 images of 5 different people, 64 × 64 size [15]. **USPS-01:** 1100 instances of handwritten digits 0 and 1 from the USPS dataset. **USPS-17:** 1100 instances of handwritten digits 1 and 7 from the USPS dataset. **USPS-245:** 1650 instances of handwritten digits 2,4 and 5 from the USPS dataset [8]. **20ng*** are subsets of the 20 newsgroups text dataset [13]. The dataset 20ngA contains 100 documents from 2 newsgroups: misc.forsale and soc.religion.christian, 20ngB adds 100 documents to each group of 20ngA, 20ngC adds 200 from talk.politics.guns to 20ngB and 20ngD adds 200 from rec.sport.baseball to 20ngC.

To construct the pairwise similarity matrix we first need to choose a kernel and tune its parameters. Automatic parameter and kernel selection for unsupervised learning is still a difficult problem. Furthermore, different parameter values may be found to be optimal for different clustering algorithms. To avoid this problem we chose parameter-free affinity matrices. For the text datasets **20ng***, the affinity matrix we used is the cosine similarity between feature vectors. Note that no parameter needs to be tuned in the cosine kernel. In all other datasets, we used the k-nearest neighbor graph, based on the Euclidean distance, to construct the pairwise relations. We set $w_{ij} = 1$ if node i is a k-nearest neighbor of node j or j is a k-nearest neighbor of i. Otherwise, we set $w_{ij} = 0$.

Table 2. Clustering performance comparison on several real datasets. For all measures a higher number means better clustering. Bold numbers mark the best results for each dataset.

		PIC [10]			**NJW** [14]			**Graclus** [3]			**ITPC**		
Dataset	**k**	**Pur**	**NMI**	**RI**	**Pur**	**NMI**	**RI**	**Pur**	**NMI**	**RI**	**Pur**	**NMI**	**RI**
Iris	3	.687	.510	.727	.900	.778	.887	.840	.722	.837	**.973**	**.901**	**.966**
Glass	6	.579	.235	.702	.594	.299	.718	.584	.280	.713	**.626**	**.326**	**.727**
Wine	3	.961	.837	.947	**.966**	**.878**	**.955**	**.966**	**.878**	**.955**	.955	.847	.940
WDBC	2	.747	.307	.622	.932	.628	.872	**.947**	**.719**	**.900**	.893	.494	.809
OlFace5	5	.500	.365	.754	.560	.439	.793	.600	**.462**	**.806**	**.620**	.460	.803
USPS-01	2	.692	.243	.574	.988	.915	.982	**.991**	**.934**	**.982**	**.991**	**.934**	**.982**
USPS-17	2	.548	.010	.505	.979	.856	.959	**.982**	**.869**	**.964**	**.982**	**.869**	**.964**
USPS-245	3	.700	.510	.765	.664	.492	.707	.864	.660	.847	**.958**	**.844**	**.947**
20ngA	2	**.960**	**.759**	**.923**	**.960**	**.759**	**.923**	.945	.701	.896	.955	.736	.914
20ngB	2	.885	.568	.796	.508	.030	.500	.927	.626	.865	**.958**	**.747**	**.919**
20ngC	3	.642	.489	.692	.625	.339	.679	.603	.387	.678	**.713**	**.401**	**.736**
20ngD	4	.539	.295	.650	.504	.281	.669	.599	**.402**	.687	**.616**	.345	**.748**
Average		.703	.427	.721	.765	.558	.803	.821	.637	.844	**.853**	**.659**	**.871**

To evaluate the performance of the clustering methods we measured the clustering results against the true labels using three external validation indices: cluster purity (Pur), normalized mutual information (NMI), and the Rand index (RI). We used all these measures to ensure a more thorough evaluation of clustering results due to the different characteristics of each measure. We refer the reader to [11] for details regarding these measures.

Table 2 presents the results of comparing ITPC to three other clustering algorithms: Spectral clustering (NJW) [14], Power Iteration Clustering (PIC) [10] and the Graclus algorithm [3]. We also tried the Ncut [16] version of spectral clustering and the results were slightly worse than those obtained by the NJW algorithm. We also ran the k-means algorithm (using the i-th row of the weight matrix W as the feature vector for the point i) and its results were the worst. It can be seen that on most datasets ITPC outperformed the other methods or at least produced quite similar results which indicates that the MI clustering score is more suitable for pairwise clustering than the Ncut score. Note that the Graclus algorithm outperforms spectral methods which validates our optimization strategy and indicates that direct optimization of a pairwise clustering score is better (and faster) than applying eigen-vector based methods. Note also that in one case the NJW algorithm failed badly (20ngB) and in another case (USPS-17) the PIC algorithm failed badly. The most likely cause being that the top eigen-vectors of the graph Laplacian failed to provide a good low-dimensional embedding for the k-means. Such problem does not exist in sequential optimization.

The ITPC algorithm utilizes a greedy approach to maximize the mutual information score $I(Y_1; Y_2)$ (6). In principle, this optimization approach can get stuck in local maxima points. Next we demonstrated that in the datasets we used there was no problem of getting stuck in local optima. Using the ground-truth labels we can compute the mutual-information score of the true clustering and compare it to the score of the clustering obtained by the ITPC algorithm. Table 3 shows the mutual-information score for all the datasets we used. In all cases the score of the clustering obtained by ITPC algorithm was higher than the score of the true clustering. Therefore, although there is no guarantee that we obtained the global maximum, it indicates that our optimization process works well.

Table 3. Comparison of the cluster-membership mutual-information score (6) of the ITPC clustering vs. the ground-truth clustering.

Dataset	ITPC Score	True Score
Iris	.949	.903
Glass	1.127	.349
Wine	.806	.761
WDBC	.474	.413
OlFace5	.554	.382
USPS-01	.580	.564
USPS-17	.539	.502
USPS-245	.916	.871
20ngA	.599	.595
20ngB	.535	.530
20ngC	.775	.756
20ngD	.886	.849

Although in pathological cases a sequential algorithm can take many iterations until convergence, in practice the number of needed iterations is much less than the number of points. In our experiments we limited the number of iterations on the data points to be 30. Note that in spectral methods, even if we use efficient algorithms to find eigenvectors, in the second step we apply k-means on the embedding results and we face a complexity issue that is also solved by limiting the number of k-means iterations.

7 Conclusion

To conclude, we introduced a simple pairwise clustering method based on applying a random-walk associated with the affinity matrix of the data points and computing the mutual information between visited clusters. The main point of our paper is defining an information theoretical criterion for pairwise clustering and showing that it yields better results than Ncut criterion and its variants. Dhillon et al. [3] showed that direct optimization of Ncut, using variants of K-means, outperforms spectral methods (that optimize an approximated cost function) in terms of both accuracy and complexity. Hence, even if we try hard to develop efficient spectral clustering variants we will not gain much. We validated this observation in Table 2.

The proposed ITPC method has linear computational complexity which makes it easily scalable for large datasets. Therefore, our algorithm is applicable to large-scale clustering tasks. Experimental results show that our algorithm outperforms state-of-the-art pairwise clustering algorithms in terms of speed, memory usage, and clustering quality. A possible weakness of the greedy method we used for optimization is getting stuck in local optima points. We showed, however, that this problem does not occur in the real datasets we analyzed. The main advantage of spectral clustering is that there is an analytic solution (for a relaxation of the Ncut cost function) and hence there is no problem of getting stuck on local optimum. We can combine ITPC and spectral clustering by first applying spectral clustering on a small subset of our data and using the result as a starting point for our approach by merging the other points to one of the obtained clusters. In this study we concentrated on the problem of pairwise clustering. The proposed method can be applied also to the more general problem of aggregating the states of a large scale Markov chain.

References

1. Chung, F.: Spectral graph theory. CBMS Regional Conference Series in Mathematics, vol. 92 (1997)
2. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley Interscience (1991)
3. Dhillon, I., Guan, Y., Kulis, B.: Weighted graph cuts without eigenvectors: A multilevel approach. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), pp. 1944–1957 (2007)
4. Dhillon, I., Mallela, S., Kumar, R.: A divisive information-theoretic feature clustering algorithm for text classification. Journal of Machine Learning Research 3, 1265–1287 (2003)
5. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: ACM SIGKDD (2003)
6. Dubnov, S., El-Yaniv, R., Gdalyahu, Y., Schneidman, E., Tishby, N., Yona, G.: A new nonparametric pairwise clustering algorithm based on iterative estimation of distance profiles. Macine Learning 47(1), 35–61 (2002)
7. Goldberger, J., Erez, K., Abeles, M.: A Markov clustering method for analyzing movement trajectories. In: IEEE Machine Learning for Signal Processing Workshop, MLSP (2007)
8. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer (2001)
9. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Scientific Computing, 359–392 (1999)
10. Lin, F., Cohen, W.: Power iteration clustering. In: Int. Conf. on Machine Learning (2010)
11. Manning, C., Raghavan, P., Schutze, H.: Introduction to information retrieval. Cambridge University Press (2008)
12. Meila, M., Shi, J.: A random walks view of spectral segmentation. In: AISTATS (2001)
13. Mitchell, T.: Machine learning. McGraw Hill (1997)

14. Ng, A.Y., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems 14 (2002)
15. Samaria, F., Harter, A.: Parameterisation of a stochastic model for human face identification. In: IEEE Workshop on Applications of Computer Vision (1994)
16. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. pattern Anal. Machine Intell. 22(8), 888–905 (2000)
17. Slonim, N., Friedman, N., Tishby, N.: Unsupervised document classification using sequential information maximization. In: Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (2002)
18. Slonim, N., Tishby, N.: Document clustering using word clusters via the information bottleneck method. In: Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (2000)
19. Tishby, N., Pereira, F., Bialek, W.: The information bottleneck method. In: Allerton Conf. on Communication, Control and Computing (1999)
20. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing, 395–416 (2007)
21. Yu, S.X., Shi, J.: Multiclass spectral clustering. In: Int'l Conf. Computer Vision (2003)
22. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: Advances in Neural Information Processing Systems 17 (2005)

Correlation Clustering with Stochastic Labellings

Nicola Rebagliati[1], Samuel Rota Bulò[2], and Marcello Pelillo[2]

[1] VTT Technical Research Centre of Finland, 02044, Finland
nicola.rebagliati@gmail.com
[2] Department of Enviromental Science, Computer Science and Statistics,
Universitá Ca' Foscari Venezia, 30121 Italy
{srotabul,pelillo}@dsi.unive.it

Abstract. Correlation clustering is the problem of finding a crisp partition of the vertices of a correlation graph in such a way as to minimize the disagreements in the cluster assignments. In this paper, we discuss a relaxation to the original problem setting which allows probabilistic assignments of vertices to labels. By so doing, overlapping clusters can be captured. We also show that a known optimization heuristic can be applied to the problem formulation, but with the automatic selection of the number of classes. Additionally, we propose a simple way of building an ensemble of agreement functions sampled from a reproducing kernel Hilbert space, which allows to apply correlation clustering without the empirical estimation of pairwise correlation values.

Keywords: Correlation clustering, stochastic labelling, ensemble clustering, Baum-Eagon inequality.

1 Introduction

Correlation Clustering is a recent clustering formulation, introduced in [4], which consists in partitioning vertices of a graph, whose edges are labelled as positive (similar) or negative (dissimilar). The goal is to find a partition in such a way as to minimize the number of negative intra-cluster edges and positive inter-cluster edges. Such a setting can be found, *e.g.*, in document clustering, where the number of clusters (topics) is not known in advance and a classifier is given which outputs whether two documents are similar or not. Unlike traditional partitional clustering approaches, this formulation does not need the number of clusters as a user parameter, but it is able to automatically perform a model selection.

Due to the difficulty of the problem, which is NP-complete [4], much work has been done in the direction of finding bounds and approximate solutions. In [4], the authors provide a constant time approximation for minimizing the disagreement and a polynomial time approximation scheme for maximizing the agreements. Later theoretical and practical improvements were made by [1][13][26] with insightful approximation algorithms that exploit linear programming or

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 120–133, 2013.

semidefinite programming. A spectral approach to solve correlation clustering with 2 clusters has been proposed in [12]. A learning theoretical analysis of correlation clustering is presented in [19]. Practical considerations, comparison and experimentation with different algorithms, also heuristical ones, can be found in [20][14].

An important application of correlation clustering is *consensus* clustering [25,23,16], *i.e.* a methodology for summarizing an ensemble of different partitions of the same dataset into a single partition. The partitions are typically obtained by applying different clustering algorithms with possibly different parametrizations on the dataset. Correlation clustering can be used for the consensus clustering algorithm, by noting that each partition in the ensemble provides observations of graph vertices to co-occur in a cluster. Indeed, these observations can be combined to estimate the similarity or dissimilarity among vertices in the graph.

Motivation and Contribution. The classic correlation clustering formulation leads to a hard partition of the graph vertices. This inhibits the possibility of capturing overlapping clusters, which is useful in many applications. To overcome this limitation, we discuss in this paper two alternative formulations of correlation clustering, where the requirement of having a crisp partition of the graph vertices is relaxed by allowing probabilistic assignments of vertices to clusters, which are regarded to as stochastic labellings. By so doing, vertices can be potentially assigned to multiple clusters. However, we show that the first formulation is essentially equivalent to classic correlation clustering, whereas the second one is different as it is able to capture overlapping clusters, preserving nevertheless the important property of automatic selection of the number of clusters. For each formulation an iterative scheme, based on the work of [10,9], allows to find a locally minimizing solution. In addition, we introduce a simple way of building an ensemble of agreement functions sampled from a reproducing kernel Hilbert space, without resorting on empirical estimations of the probability that two vertices will co-occur in the same class.

Previous Work. Our reference scheme is an adaptation of [24] to correlation clustering. In [24] they use stochastic assignments for finding overlapping communities in a social network. See also [3] for a rather different approach to the problem of finding groups from similarity matrices. However both [3,24] fix the number of classes K. By modifying the approach of [10] we have a different algorithm which automatically selects the number of classes K. In [8] they attack the problem of finding overlapping groups in correlation clustering by extending the Correlation Clustering functional with multi-labelling functions, instead of relaxing the ownership assignments.

Outline. The paper is organized as follows. Section 2 formally introduces the problem of correlation clustering within a more general setting, where we might have missing edges in the graph and noisy labels on the edges. Section 3 introduces two relaxed formulations of correlation clustering, which allow for stochastic assignments of vertices to clusters, and show some theoretical properties among which the ability of capturing overlapping clusters. We address the optimization problems related to the two proposed formulations in Section 4, where we make use of a result due to Baum and Eagon. In section 5 we introduce our ensemble of agreement functions sampled from kernel space and in Section 6 we show experiments on real and synthetic datasets. In section 7 we draw the conclusions.

2 Correlation Clustering

A *correlation graph* $G = (V, E, w)$ is an edge-weighted graph without self-loops, where $V = \{1, \dots, n\}$ is a set of vertices, $E \subseteq V \times V$ is a set of edges and $w : E \to \{0, 1\}$ is a function mapping edges $(i, j) \in E$ to 1 or 0 according to whether i and j are *correlated* or not. Hereafter, we write w_{ij} for $w(i, j)$.

Let $L_k = \{1, \dots, k\}$ be a set of k labels. A (stochastic) *k-labelling*, or simply labelling if k is understood by the context, for a correlation graph $G = (V, E, w)$ is a matrix $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_n) \in \Delta_k^n$, where $\mathbf{y}_i \in \Delta_k$ is a probabilistic assignment of a label in L_k to a vertex $i \in V$, where

$$\Delta_k = \left\{ \mathbf{z} \in \mathbb{R}^k \, : \, \sum_{\ell \in L_k} z_\ell = 1 \text{ and } z_\ell \geq 0 \text{ for all } \ell \in L_k \right\}$$

is the $(k - 1)$-dimensional *simplex*. We denote by $\Lambda_k = \Delta_k \cap \{0, 1\}^k$ the set of deterministic assignments of labels to vertices, *i.e.* the set of distributions with full mass on a specific label in L_k. A labelling $\mathbf{X} \in \Lambda_k^n$ is regarded as a *deterministic labelling.* Note that for stochastic as well as deterministic labellings, parameter k should be intended as the maximum number of labels assignable to vertices. This implies that some labels in L_k may not be used. Moreover, for all $k' > k$, Λ_k and Δ_k can be naturally embedded in $\Lambda_{k'}$ and $\Delta_{k'}$, respectively.

Given a deterministic labelling $\mathbf{X} \in \Lambda_k^n$ for a correlation graph $G = (V, E, w)$, we say that two vertices connected by an edge $(i, j) \in E$ *agree* if $\mathbf{x}_i^\top \mathbf{x}_j = w_{ij}$. We say that they *disagree*, in all other cases. The *total disagreement* $\phi_G(\mathbf{X})$ of a labelling $\mathbf{X} \in \Lambda_k^n$ for a correlation graph $G = (V, E, w)$ is the number of edges in G consisting of disagreeing vertices, *i.e.*

$$\phi_G(\mathbf{X}) = \sum_{(i,j) \in E} w_{ij} (1 - \mathbf{x}_i^\top \mathbf{x}_j) + (1 - w_{ij})\, \mathbf{x}_i^\top \mathbf{x}_j \, . \tag{1}$$

Similarly, the *total agreement* of a labelling $\mathbf{X} \in \Lambda_k^n$ for G is the number of edges in E consisting of agreeing vertices.

A *correlation k-clustering* of a correlation graph $G = (V, E, w)$ is a k-labelling $\mathbf{X}^* \in \Lambda_k^n$ minimizing the total disagreement, *i.e.*

$$\phi^*_{G,\Lambda_k} = \phi_G(\mathbf{X}^*) = \min \{ \phi_G(\mathbf{X}) \, : \, \mathbf{X} \in \Lambda_k^n \} \, . \tag{2}$$

A correlation n-clustering for a correlation graph G with n vertices is called simply a *correlation clustering* for G. As argued by [4], we can state the following remark.

Remark 1 (Model Selection Property). There is an optimal parameter value k^* such that $\phi^*_{G,\Lambda_k} \geq \phi^*_{G,\Lambda_{k^*}}$ holds for all k. Furthermore, if $k' > k$ it holds that $\phi^*_{G,\Lambda_{k'}} \leq \phi^*_{G,\Lambda_k}$. Hence, by selecting $k = n$, where n is the number of vertices of G, we are guaranteed that $\mathbf{X}^*$ is a k-labelling achieving minimum disagreement over all possible choices of k.

2.1 Clustering with Noisy Correlation Graphs

We depart from the standard correlation clustering problem, by assuming input graphs to be noisy with respect to the edge correlation values. Specifically, we are not given w_{ij} explicitly, but probabilities p_{ij} are provided of observing i and j correlated. Let $\mathcal{G} = (V, E, p)$ be a random variable generating correlation graphs (*random correlation graph variable*) with vertex set V and edge set E, where for each edge $(i, j) \in E$ the value of w_{ij} is independently drawn according to a Bernoulli distribution with parameter p_{ij}. The *expected total disagreement* of a labelling $\mathbf{X} \in \Lambda^n_k$ with respect to $\mathcal{G}$ is given by:

$$\phi_{\mathcal{G}}(\mathbf{X}) = \mathbb{E}_{\mathcal{G}}\left[\phi_{\mathcal{G}}(\mathbf{X})\right] = \sum_{(i,j)\in E} p_{ij} + \mathbf{x}_i^\top \mathbf{x}_j(1 - 2p_{ij})\,. \tag{3}$$

For notational convenience, we express total disagreement in equation (1) and expected total disagreement in equation (3) with the same symbol ϕ, but they differ in the subscript being a correlation graph in the former case and a random correlation graph variable in the latter.

In order to cope with random correlation graphs, we consider a correlation clustering formulation, where we aim at finding a labelling in such a way as to minimize the *expected* total disagreement with respect to a random correlation graph variable $\mathcal{G}$. This yields the following minimization problem

$$\phi^*_{\mathcal{G},\Lambda_k} = \phi_{\mathcal{G}}(\mathbf{X}^*) = \min\left\{\phi_{\mathcal{G}}(\mathbf{X}) \;:\; \mathbf{X} \in \Lambda^n_k\right\}\,, \tag{P}$$

where $\mathbf{X}^* \in \Lambda^n_k$ denotes a labelling achieving minimum expected disagreement. The model selection property stated in Remark 1 holds straightforwardly also for this formulation. Note that weighted versions of correlation clustering has been addressed also in [19].

3 Relaxed Formulations with Stochastic Labellings

In this section we will relax the assumption on the labelling by allowing for stochastic assignments of vertices to labels. There is a two-fold reason why we introduce stochastic labellings. In first place it allows us to move from a discrete optimization problem to a continuous one and make use of a result known as

Baum-Eagon inequality in probability domain for finding a local solution (see Section 4). Secondly, having stochastic label assignments allows to capture overlapping clusters, by letting graph vertices to be assigned to more labels with non-zero probability.

We move from deterministic labellings to stochastic ones by replacing the variables $\mathbf{X} \in \Lambda_k^n$ with variables $\mathbf{Y} \in \Delta_k^n$ in (3):

$$\phi_{\mathcal{G}}(\mathbf{Y}) = \sum_{(i,j)\in E} p_{ij} + \mathbf{y}_i^\top \mathbf{y}_j (1 - 2p_{ij}) . \tag{4}$$

Here, $\mathbf{y}_i^\top \mathbf{y}_j$ represents the probability of vertices i and j to occur in the same class, under independence assumption. The relaxed version of correlation k-clustering can thus be formulated as

$$\phi^*_{\mathcal{G},\Delta_k} = \phi_{\mathcal{G}}(\mathbf{Y}^*) = \min\left\{\phi_{\mathcal{G}}(\mathbf{Y}) \,:\, \mathbf{Y} \in \Delta_k^n\right\} , \tag{Q1}$$

where $\mathbf{Y}^* \in \Delta_k^n$ denotes an optimal stochastic k-labelling achieving minimum expected disagreement.

The relaxed formulation of correlation clustering in (Q1) is a continuous optimization problem, which turns out to be substantially equivalent to (P). Consequently, despite the stochastic label assignments, overlapping clusters are not captured. The following proposition shows that, for all choices of k, (P) and (Q1) yield the same value.

Proposition 1. *Let $\mathcal{G} = (V, E, p)$ be a random correlation graph variable. Then $\phi^*_{\mathcal{G},\Lambda_k} = \phi^*_{\mathcal{G},\Delta_k}$ for all choices of $k > 0$.*

Proof. Note that any variable $X \in \Lambda_k^n \subset \Delta_k^n$. Hence, the domain of program (P) is a strict subset of the one of (Q1), which implies $\phi^*_{\mathcal{G},\Lambda_k} \geq \phi^*_{\mathcal{G},\Delta_k}$. On the other hand, let $Y^* = (\mathbf{y}_1^*, \dots, \mathbf{y}_n^*)$ be a solution of (Q1), let $\mathcal{X}_i \in \Lambda_k$, $1 \leq i \leq n$, be multinomial random vectors with parameters $n = 1$ and probabilities $\mathbf{y}_i^*$, and let $\mathcal{X} = (\mathcal{X}_1, \dots \mathcal{X}_n) \in \Lambda_k^n$ be a random (deterministic) labelling generator. Then $\mathbb{E}_{\mathcal{X}}[\phi_{\mathcal{G}}(\mathcal{X})] \geq \phi^*_{\mathcal{G},\Lambda_k}$, but since $\mathbb{E}_{\mathcal{X}}[\phi_{\mathcal{G}}(\mathcal{X})] = \phi^*_{\mathcal{G},\Delta_k}$ we have that $\phi^*_{\mathcal{G},\Delta_k} \geq \phi^*_{\mathcal{G},\Lambda_k}$.

We show in Figure 1 an example of correlation clustering, where we have 3 clear overlapping clusters. In 1(a) we show the values of p_{ij} and in 1(b) we can clearly see that the solution obtained by (Q1) is a deterministic labelling $\mathbf{Y}^* \in \Lambda_k^n$ as the matrix of probabilities of co-occurrence $(\mathbf{Y}^*)^\top \mathbf{Y}^*$ contains 0s and 1s. This confirms the intuition coming from Proposition 1 and shows a clear inability of this formulation to capture overlapping clusters.

In order to overcome the limitations of (Q1) we consider a different way of computing the total disagreement of a labelling $\mathbf{X} \in \Lambda_k$ for a correlation graph $G = (V, E, w)$, which is given by

$$\varphi_G(\mathbf{X}) = \sum_{(i,j)\in E} \left(\mathbf{x}_i^\top \mathbf{x}_j - w_{ij}\right)^2 . \tag{5}$$

In the presence of random correlation graphs generated according to $\mathcal{G} = (V, E, p)$, the corresponding expected total disagreement of a labelling $\mathbf{X}$ for $\mathcal{G}$ gives

$$\varphi_{\mathcal{G}}(\mathbf{X}) = \mathbb{E}_{\mathcal{G}}\left[\varphi_{\mathcal{G}}(\mathbf{X})\right] = \sum_{(i,j)\in E} p_{ij} + \mathbf{x}_i^\top \mathbf{x}_j(\mathbf{x}_i^\top \mathbf{x}_j - 2p_{ij})\,. \tag{6}$$

Note that $\varphi_G(\mathbf{X}) = \phi_G(\mathbf{X})$ and $\varphi_{\mathcal{G}}(\mathbf{X}) = \phi_{\mathcal{G}}(\mathbf{X})$. The relaxed version of (6), which uses a stochastic labelling $\mathbf{Y}$, is

$$\varphi_{\mathcal{G}}(\mathbf{Y}) = \sum_{(i,j)\in E} p_{ij} + \mathbf{y}_i^\top \mathbf{y}_j(\mathbf{y}_i^\top \mathbf{y}_j - 2p_{ij})\,. \tag{7}$$

Finally, the relaxed correlation k-clustering formulation related to (7) is given by

$$\varphi^*_{\mathcal{G},\Delta_k} = \varphi_{\mathcal{G}}(\mathbf{Y}^*) = \min\left\{\varphi_{\mathcal{G}}(\mathbf{Y}) \,:\, \mathbf{Y} \in \Delta_k^n\right\}\,, \tag{Q2}$$

where $\mathbf{Y}^* \in \Delta_k^n$ denotes an optimal stochastic k-labelling for the minimization.

Let $d_{\mathcal{G}}(\mathbf{Y})$ be the following function

$$d_{\mathcal{G}}(\mathbf{Y}) = \sum_{(i,j)\in E} \mathbf{y}_i^\top \mathbf{y}_j(1 - \mathbf{y}_i^\top \mathbf{y}_j)$$

which measures the uncertainty of the stochastic labelling $\mathbf{Y}$. Indeed, $d_{\mathcal{G}}(\mathbf{X}) = 0$ for all $\mathbf{X} \in \Lambda_k^n$, while it is strictly positive in general.

The next result, which is close in spirit to Proposition 1, relates the correlation clustering formulations (Q2) and (P). Specifically it provides a lower and upper bound for (P) in terms of (Q2) and $d_{\mathcal{G}}(\cdot)$ for all choices of k.

Proposition 2. *Let $\mathcal{G} = (V, E, p)$ be a random correlation graph variable. Then*

$$\varphi^*_{\mathcal{G},\Delta_k} \le \phi^*_{\mathcal{G},\Lambda_k} \le \varphi^*_{\mathcal{G},\Delta_k} + d_{\mathcal{G}}(\mathbf{Y}^*)$$

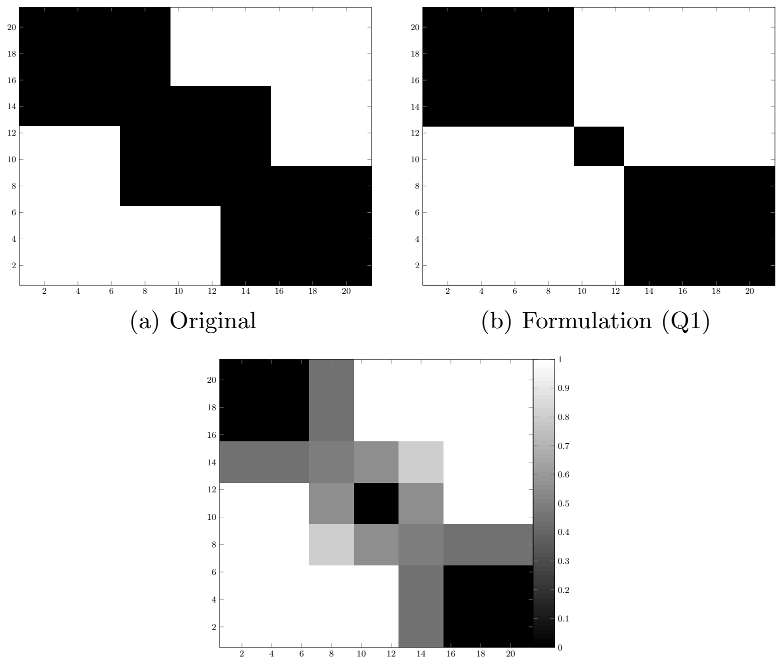
for all choices of $k > 0$, where $\mathbf{Y}^* \in \Delta_k^n$ is a solution of (Q2).

Proof. The first inequality $\varphi^*_{\mathcal{G},\Delta_k} \le \phi^*_{\mathcal{G},\Lambda_k}$ trivially holds because $\Lambda_k \subset \Delta_k$. The second inequality follows by noting that $\phi_{\mathcal{G}}(\mathbf{Y}) = \varphi_{\mathcal{G}}(\mathbf{Y}) + d_{\mathcal{G}}(\mathbf{Y})$, which implies $\phi^*_{\mathcal{G},\Delta_k} \le \varphi^*_{\mathcal{G},\Delta_k} + d_{\mathcal{G}}(\mathbf{Y}^*)$. By Proposition 1 the result derives.

From Proposition (2) we can see that if the solution of (Q2) is deterministic, then it is also a solution of (P). Otherwise, the higher the distance from a deterministic labelling, the larger the gap between $\varphi^*_{\mathcal{G},\Delta_k}$ and $\phi^*_{\mathcal{G},\Lambda_k}$ might be.

In Figure 1(c) we show the behaviour of formulation (Q2) with the toy example with 3 overlapping clusters, which has been previously introduced. We note that as opposed to (Q1), this formulation is indeed able to assign vertices to multiple classes, obtaining thereby a solution which reflects to the desired clustering.

Also for formulations (Q1) and (Q2) the model selection property of Remark 1 holds, clearly on the respective objective functions. It is worth mentioning that a formulation, which is equivalent to (Q2), has been used in [24] for communities



(a) Original

(b) Formulation (Q1)

(c) Formulation (Q2)

Fig. 1. Example of correlation clustering with 3 clear overlapping clusters. Left to right: Original correlation graph; $(\mathbf{Y}^*)^\top \mathbf{Y}^*$ with $\mathbf{Y}^*$ solution of (Q1); $(\mathbf{Y}^*)^\top \mathbf{Y}^*$ with $\mathbf{Y}^*$ solution of (Q2).

detection. However, the authors were not aware of the relation with correlation clustering and, thus, the automatic selection of the number of clusters.

Formulation (Q2) is, unfortunately, an highly non-convex minimization problem which is very difficult to attack with an exact algorithm working in a reasonable computational time. In the next section we propose to use non-exact algorithms based on two iterative formulations, for both (Q1) and (Q2), which ensure to return a locally minimizing solution.

4 Optimization Using the Baum-Eagon Inequality

In order to solve our optimization problem we shall use the following important result which is generally known as the Baum-Eagon inequality [5].

Theorem 1 (Baum-Eagon). *Let* $\mathbf{Y} \in \Delta_k^n$ *and* $Q(\mathbf{Y})$ *be a homogeneous polynomial in the variables* $y_{i\ell}$ *with nonnegative coefficients. Define the mapping* $\mathbf{Z} = \mathcal{M}(\mathbf{Y}) \in \Delta_k^n$ *as follows:*

$$z_{i\ell} = y_{i\ell} \frac{\partial Q(\mathbf{Y})}{\partial y_{i\ell}} \bigg/ \sum_{\ell' \in L_k} y_{i\ell'} \frac{\partial Q(\mathbf{Y})}{\partial y_{i\ell'}}\,, \tag{8}$$

for all $i = 1 \dots n$ and $\ell \in L_k$. Then $Q(\mathcal{M}(\mathbf{Y})) > Q(\mathbf{Y})$, unless $\mathcal{M}(\mathbf{Y}) = \mathbf{Y}$. In other words $\mathcal{M}$ is a growth transformation for the polynomial Q.

Although the original theorem applies to homogeneous polynomials only, the result has been generalized later by Baum and Sell [7] who proved that Theorem 1 still holds in the case of arbitrary polynomials with nonnegative coefficients, and further extended the result by proving that $\mathcal{M}$ increases Q *homotopically*, which means that for all $0 \leq \eta \leq 1$, $Q(\eta\mathcal{M}(\mathbf{Y}) + (1-\eta)\mathbf{Y}) \geq Q(\mathbf{Y})$ with equality if and only if $\mathcal{M}(\mathbf{Y}) = \mathbf{Y}$.

The Baum-Eagon inequality provides an effective iterative means for maximizing polynomial functions in probability domains, and in fact it has served as the basis for various statistical estimation techniques developed within the theory of probabilistic functions of Markov chains [6]. As pointed out in [7], we remark that the mapping $\mathcal{M}$ defined in Theorem 1 makes use of the first derivative only and yet is able to take finite steps while increasing Q. This contrasts sharply with classical gradient methods, for which an increase in the objective function is guaranteed only when infinitesimal steps are taken, and determining the optimal step size entails computing higher-order derivatives.

It is not difficult to show that, by starting from the interior of the simplex, the fixed points of the Baum-Eagon dynamics satisfy the first-order Karush-Kuhn-Tucker necessary conditions for local maxima and that strict local solutions are in correspondence to asymptotically stable points.

4.1 Algorithms for Correlation Clustering with Stochastic Labellings

We show now how the Baum-Eagon inequality can be used in order to optimize the relaxed formulations of correlation k-clustering introduced in Section 3. The theorem, however, cannot be applied directly as its hypothesis are not fulfilled. Indeed, the polynomials with variables in probability domain of (Q1) and (Q2) need to be minimized and not maximized, and they do not have in general nonnegative coefficients. Nevertheless, by exploiting the simplex constraints, we can transform the aforementioned formulations into equivalent ones, which can then be tackled by using the Baum-Eagon theorem. Hereafter, we denote with $\mathbf{E}$ a $k \times k$ matrix of all 1's, and with $\mathbf{I}$ the $k \times k$ identity matrix.

As for (Q1), by observing that $\mathbf{y}_i^\top \mathbf{E} \mathbf{y}_j = 1$ for all $(i, j) \in E$ and $\mathbf{Y} \in \Delta_k^n$, it is straightforward to rewrite $-\phi_\mathcal{G}(\mathbf{Y})$ as

$$-\phi_\mathcal{G}(\mathbf{Y}) = -|E| + \sum_{(i,j)\in E} \mathbf{y}_i^\top [\mathbf{E} + (2p_{ij} - 1)\mathbf{I}]\mathbf{y}_j - p_{ij}$$

which is a homogeneous polynomial with nonnegative coefficients (constant terms can be dropped), in probability domain Δ_k^n. This equivalence allows us to find a local solution of (Q1) by maximizing $-\phi_\mathcal{G}$. Hence, we can apply the Baum-Eagon theorem by using (8) with $Q = -\phi_\mathcal{G}$. This yields the following update rule for $\mathbf{Y} = (y_{i\ell})$:

$$y_{i\ell}^{(t+1)} = y_{i\ell}^{(t)} \frac{\left[\sum_{j\in E_i} 1-(1-2p_{ij})y_{j\ell}^{(t)}\right]}{\sum_{\ell\in L_k} y_{i\ell}^{(t)} \left[\sum_{j\in E_i} 1-(1-2p_{ij})y_{j\ell}^{(t)}\right]}, \tag{Alg-Q1}$$

where $E_i = \{j \mid (i,j) \in E\}$ and the starting labelling $\mathbf{Y}^{(0)}$ might be any point in the interior of Δ_k^n.

Similarly for (Q2), we can rewrite $-\varphi_{\mathcal{G}}(\mathbf{Y})$ as the following homogeneous polynomial with nonnegative coefficients:

$$-\varphi_{\mathcal{G}}(\mathbf{Y}) = -|E| + \sum_{(i,j)\in E} \left[\mathbf{y}_i^\top(\mathbf{E}-\mathbf{I})\mathbf{y}_j\right]^2 - p_{ij}$$

which can be locally maximized by means of the Baum-Eagon result obtaining a local solution of (Q2). This yields the following update rule:

$$y_{i\ell}^{(t+1)} = y_{i\ell}^{(t)} \frac{\sum\limits_{j\in E_i} \left(1-y_{j\ell}^{(t)}\right)(1-\mathbf{y}_i^\top\mathbf{y}_j)+2p_{ij}y_{j\ell}^{(t)}}{\sum\limits_{\ell\in L_k} y_{i\ell}^{(t)} \sum\limits_{j\in E_i} \left(1-y_{j\ell}^{(t)}\right)(1-\mathbf{y}_i^\top\mathbf{y}_j)+2p_{ij}y_{j\ell}^{(t)}}, \tag{Alg-Q2}$$

where the starting labelling $\mathbf{Y}^{(0)}$ might be any point in the interior of Δ_k^n.

Both update rules (Alg-Q1) and (Alg-Q2) satisfy the invariant property $\mathbf{Y}^{(t)} \in \Delta_k^n$ for all $t > 0$ if $\mathbf{Y}^{(0)} \in \Delta_k^n$ and lead to a local solution of the respective correlation clustering formulations.

5 Ensemble of Random Functions Sampled from Kernel Space

In this section we show how to construct a simple ensemble of agreement functions sampled from a reproducing kernel Hilbert space, which allows to obtain a random correlation graph variable for our algorithm from an arbitrary clustering dataset, without resorting on empirical estimations of the probability that two vertices will co-occur in the same class. This is an alternative approach to in [15].

A *kernel* is a symmetric function $K : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ such that for any dataset $(\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{X}^n$ the comparison matrix $\mathbf{K}$ with entries $k_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, regarded to as *Gram matrix*, is positive semidefinite, i.e. all its eigenvalues are nonnegative. A kernel uniquely determines a *reproducing kernel Hilbert space* [2]. This is a vector space $\mathcal{H}$ of functions $f : \mathbb{X} \to \mathbb{R}$ with the following properties:

- $f(\mathbf{x}) = \langle f, K(\mathbf{x}, \cdot)\rangle_{\mathcal{H}}$
- $\forall \mathbf{x} \in \mathbb{X}.\, K(\mathbf{x}, \cdot) \in \mathcal{H}$

where $\mathcal{H} = \overline{span\{K(\mathbf{x}, \cdot) : \mathbf{x} \in \mathcal{H}\}}$.

A *feature map* is a function $\Phi : \mathbb{X} \to \mathcal{H}$ associated to a kernel K such that $k_{ij} = \langle \Phi_i, \Phi_j \rangle_{\mathcal{H}}$, where $\Phi_i = \Phi(\mathbf{x}_i)$. By the reproducing kernel property, we can associate each function $f \in \mathcal{H}$ with an evaluating hyperplane w_f, such that $f(\mathbf{x}) = \langle w_f, \Phi(\mathbf{x}) \rangle_{\mathcal{H}}$. At the same time, a function $f \in \mathcal{H}$, can be regarded as a 2-class classifier mapping $\mathbf{x} \in \mathbb{X}$ to a label according to the sign of $f(\mathbf{x})$.

The probability p_{ij} that a randomly drawn function from $f \in \mathcal{H}$ with uniform distribution will put two data points $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{X}$ in the same class can be computed as a function of the angle θ_{ij} between Φ_i and Φ_j [18]:

$$p_{ij} = 1 - \frac{\theta_{ij}}{\pi}\,.$$

The angle here is given by $\theta_{ij} = \arccos(k_{ij}/\sqrt{k_{ii}k_{jj}})$. The matrix $\mathbf{P} = (p_{ij})$ of the probabilities can thus be computed as the following function of the kernel matrix $\mathbf{K}$:

$$\mathbf{P} = \mathbf{E} - \frac{1}{\pi} \arccos(\mathbf{D}_{\mathbf{K}}^{-\frac{1}{2}} \mathbf{K} \mathbf{D}_{\mathbf{K}}^{-\frac{1}{2}})\,,$$

where $\mathbf{D}_{\mathbf{K}}$ is the diagonal of $\mathbf{K}$. Note that for the Gaussian kernel the formula is simpler because the features have norm 1 as $K(\mathbf{x}, \mathbf{x}) = 1$. In this case indeed we obtain

$$\mathbf{P}_{\mathrm{rbf}} = \mathbf{E} - \frac{1}{\pi} \arccos(\mathbf{K}_{\mathrm{rbf}})\,.$$

Matrix $\mathbf{P}$ can be used to obtain a random correlation graph variable $\mathcal{G}$ representing the data to cluster by means of one of the correlation clustering approaches, previously described.

In order to account for classifiers with a larger number of classes, we consider the possibility of specifying the number of functions f that should be drawn from $\mathcal{H}$ for the classification. Under independence assumption the probability that two sample points $\mathbf{x}_i$ and $\mathbf{x}_j$ will be given the same class by each of the sampled functions, say d, is simply p_{ij}^d.

6 Experiments

In this section we assess the effectiveness of the relaxed formulations introduced in section 3 on both real and synthetic datasets.

For the experiments we considered the heuristics we introduced in Section 4, namely Alg-Q1 and Alg-Q2, which provide solutions to (Q1) and (Q2), respectively. We compared our algorithms against two heuristics for (P). The first is a randomized heuristic, called CC-Pivot, yielding a 11/7 approximation, which has been introduced in [1]. The second one is a local search heuristic, called Best One Element Move (BOEM), introduced in [17]. All four heuristics are repeated with 25 different random initializations and best results are returned.

We evaluated the algorithms on two real datasets from the UCI Machine Learning Repository: Iris and House-Votes. Iris consists of 150 data points in

4-dimensional space divided uniformly into 3 classes. House-Votes consists of 435 data points in 17-dimensional space divided into 2 classes (267/168). We also considered a synthetic dataset "4NG" composed by four overlapping gaussians with 50 points each and 50 points uniformly sampled in the hyperbox containing the data as outliers.

For each dataset we created a random correlation graph variable according to the method described in Section 5 in conjunction with a RBF kernel with manually tuned scale parameter, and with $d = 3$ sampled functions.

Note that as for our algorithms, we run them with a maximum number of classes $k = 20$, which was larger than the number of classes found in the datasets. By so doing, the algorithms were able to automatically find the number of clusters. The running time of Alg-Q1 and Alg-Q2 is comparable to other methods and take few minutes (< 15) with Matlab 7.8.0 [21] for Windows 7 ©Intel ®Core $^{\text{TM}}$Duo CPU T6600 2.20GHz, 4GB RAM.

We assessed the quality of the clusterings obtained from the algorithms by computing the *confusion error* [22]. Since confusion error does not penalize the selection of a number of clusters larger than the ground truth we report also the associated number of clusters.

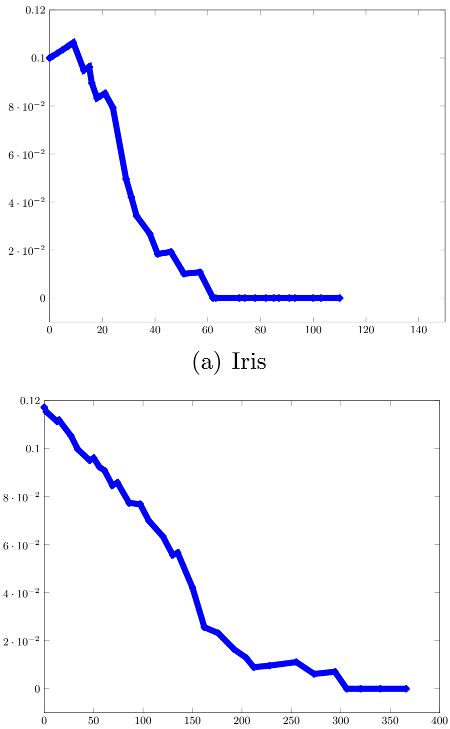
In Table 1 we report best results obtained by all the methods on all datasets. Beside the name of each dataset, we show the optimal number of classes. For each combination of dataset and algorithm we provide the number of classes obtained and the associated confusion error. As we can see among the four approaches, Alg-Q2 is the one achieving the best compromise between the automatic selection of the number of classes, and the confusion error, while the other approaches tend to overestimate the number of actual clusters in the data. Note that an advantage of having stochastic labelling is that we can measure the uncertainty in a label assignment. Since our algorithm is the only one which is able to capture such information, we report in Figure 2 the effect on the confusion error of the removal of points with the most uncertain label assignments obtained by it. As we can see, the error nicely decreases to zero. This indicates that the points where the algorithm exhibits uncertain label assignments are those leading to misclassification.

We also compared our method with the algorithm Left-Stochastic Decomposition (LSD) of [3] on datasets from [11] using the Misclassification Error [22].

Table 1. Results obtained on the datasets. We report for each combination of dataset and algorithm the number of clusters found by the algorithm and the confusion error of the solution found. We also report the optimal value of σ used for the experiment. For the Ten-Digits dataset both BOEM and CC-Pivot returned an high number of classes and their result are not significant.

Dataset (K)	σ	BOEM	CC-Pivot	Alg-Q1	Alg-Q2
Iris (3)	0.4	(31, 0.08)	(10, 0.10)	(11, 0.13)	(3, 0.10)
House-Votes (2)	0.8	(8, 0.11)	(5, 0.14)	(20, 0.37)	(2, 0.12)
Ten-Digits (10)	0.05	*	*	(20, 0.21)	(15, 0.17)
4NG (4)	0.1	(42, 0.13)	(56, 0.10)	(19, 0.13)	(7, 0.16)

As we can see from Table 2, both approaches perform comparably well, although our method achieves the best scores on most of the datasets that have been taken into account.



(a) Iris

(b) House-Votes

Fig. 2. Plot of the confusion error obtained by (Alg-Q2) by iteratively removing vertices with uncertain labels. On the x-axis we report the number of vertices removed from the dataset.

Table 2. A comparison with [3] on datasets of [11]. Number of used clusters in parenthesis.

Dataset (K)	Alg-Q2	LSD
Amazon Binary (2)	**.354**	.390
Aural Sonar (2)	**.120**	.140
Patrol (8)	**.253**	.440
Protein (4)	.347	**.200**
Voting (2)	**.094**	.100
Yeast Pfam 7-12 (2)	.380	**.360**
Yeast SW 5-7 (2)	.295	**.28**
Yeast SW 5-12 (2)	**.090**	**.090**
Yeast SW 7-12 (2)	**.095**	.100

7 Conclusions

The aim of this work is showing the relationship between classical Correlation clustering and a relaxed version which allows for stochastic labellings instead of hard ones. In proposition 1 we show that this relaxation is necessary, because Correlation clustering by itself cannot capture stochastic labellings. In proposition 2 the two functionals are put in relation. Moreover, we argue that the relaxation still preserves the property of model selection peculiar of Correlation Clustering. For both formulations we provide how to apply the Baum-Eagon inequality in order to obtain converging algorithms. As a further contribution, we show how we can practically build a simple ensemble of agreement functions sampled from a reproducing kernel Hilbert space of functions. In the experiments we obtain promising results compared to other, state-of-the-art, methods.

Acknowledgments. The work of Nicola Rebagliati was partially carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement 246016.

References

1. Ailon, N., Charikar, M., Newman, A.: Aggregating inconsistent information: ranking and clustering. In: STOC, pp. 684–693 (2005)
2. Aronszajn, N.: Theory of reproducing kernels. Trans. Amer. Math. Soc. 68, 337–404 (1950)
3. Arora, R., Gupta, M., Kapila, A., Fazel, M.: Clustering by left-stochastic matrix factorization. In: ICML, pp. 761–768 (2011)
4. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Machine Learning 56(1-3), 89–113 (2004)
5. Baum, L.E., Eagon, J.A.: An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. Bull. Amer. Math. Soc. 73, 360–363 (1967)
6. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Annals of Math. Statistics 41, 164–171 (1970)
7. Baum, L.E., Sell, G.R.: Growth transformations for functions on manifolds. Pac. J. Math. 27, 221–227 (1968)
8. Bonchi, F., Gionis, A., Ukkonen, A.: Overlapping correlation clustering. In: ICDM, pp. 51–60 (2011)
9. Bulò, S.R., Lourenço, A., Fred, A., Pelillo, M.: Pairwise probabilistic clustering using evidence accumulation. In: Hancock, E.R., Wilson, R.C., Windeatt, T., Ulusoy, I., Escolano, F. (eds.) SSPR&SPR 2010. LNCS, vol. 6218, pp. 395–404. Springer, Heidelberg (2010)
10. Bulò, S.R., Pelillo, M.: Probabilistic clustering using the baum-eagon inequality. In: ICPR, pp. 1429–1432 (2010)

11. Chen, Y., Garcia, E.K., Gupta, M.R., Rahimi, A., Cazzanti, L.: Similarity-based Classification: Concepts and Algorithms. Journal of Machine Learning Research 10, 747–776 (2009)
12. Coleman, T., Saunderson, J., Wirth, A.: Spectral clustering with inconsistent advice. In: ICML, pp. 152–159 (2008)
13. Demaine, E.D., Emanuel, D., Fiat, A., Immorlica, N.: Correlation clustering in general weighted graphs. Theor. Comput. Sci. 361(2-3), 172–187 (2006)
14. Downing, N., Stuckey, P.J., Wirth, A.: Improved consensus clustering via linear programming. In: ACSC, pp. 61–70 (2010)
15. Fern, X.Z., Brodley, C.E.: Random projection for high dimensional data clustering: A cluster ensemble approach. In: Fawcett, T., Mishra, N. (eds.) ICML, pp. 186–193. AAAI Press (2003)
16. Fred, A.L.N., Jain, A.K.: Combining multiple clusterings using evidence accumulation. IEEE Trans. Pattern Anal. Mach. Intell. 27(6), 835–850 (2005)
17. Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. In: Proceedings of the 21st International Conference on Data Engineering (ICDE), pp. 341–352 (2005)
18. Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J. ACM 42(6), 1115–1145 (1995)
19. Joachims, T., Hopcroft, J.E.: Error bounds for correlation clustering. In: ICML, pp. 385–392 (2005)
20. Mathieu, C., Schudy, W.: Bounding and comparing methods for correlation clustering beyond ILP. In: ILP-NLP (2009)
21. MATLAB: version 7.8.0 (R2009a). The MathWorks Inc., Natick, Massachusetts (2009)
22. Meilă, M.: Comparing Clusterings by the Variation of Information. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT/Kernel 2003. LNCS (LNAI), vol. 2777, pp. 173–187. Springer, Heidelberg (2003)
23. Monti, S., Tamayo, P., Mesirov, J.P., Golub, T.R.: Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. Machine Learning 52(1-2), 91–118 (2003)
24. Nepusz, T., Petróczi, A., Négyessy, L., Bazsó, F.: Fuzzy communities and the concept of bridgeness in complex networks. Physical Review E 77(1), 016107 (2008)
25. Strehl, A., Ghosh, J.: Cluster ensembles — a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3, 583–617 (2002)
26. Swamy, C.: Correlation clustering: maximizing agreements via semidefinite programming. In: SODA, pp. 526–527 (2004)

Break and Conquer: Efficient Correlation Clustering for Image Segmentation

Amir Alush and Jacob Goldberger

Bar-Ilan University, Ramat-Gan 52299, Israel

Abstract. We present a probabilistic model for image segmentation and an efficient approach to find the best segmentation. The image is first grouped into superpixels and a local information is extracted for each pair of spatially adjacent superpixels. The global optimization problem is then cast as correlation clustering which is known to be NP hard. This study demonstrates that in many cases, finding the exact global solution is still feasible by exploiting the characteristics of the image segmentation problem that make it possible to break the problem into subproblems. Each sub-problem corresponds to an automatically detected image part. We demonstrate a reduced computational complexity with comparable results to state-of-the-art on the BSDS-500 and the Weizmann Two-Objects datasets.

1 Introduction

Image segmentation is a fundamental process in many image, video, and computer vision applications. It is essentially the partitioning of an image into several constituent components. The basic task of image segmentation is thus to assign each pixel in the image to one of the image components. Many segmentation algorithms have been proposed and studied in recent decades and new algorithms are continuously emerging. These segmentation algorithms are usually based on various combinations of local low-level features and global optimization methods. In this paper we focus on the global optimization aspect of image segmentation.

Many visual tasks including segmentation can benefit from the complexity reduction achieved by transforming an image with millions of pixels into a few hundred or thousand "superpixels". Superpixels are small, homogeneous regions preserving almost all boundaries between different regions and are obtained by a low-level process based on cues such as color, edges and texture. The use of superpixels as primitive objects for clustering significantly reduce computational cost and allow feature extraction to be conducted from a larger homogeneous region. Given a superpixel graph we can first extract a local similarity measure for each pair of spatially adjacent superpixels and then find a global segmentation that is consistent with the local cues. This paradigm is common to many graph based image segmentation algorithms (e.g. [2,7,10]). However, current segmentation approaches, even when applied to superpixels, do not aim to find an exact optimal segmentation. Instead, they utilize approximation methods such

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 134–147, 2013.

as greedy hierarchical superpixels merging [7], LP-relaxation [24,17], dual decomposition [25] and spectral clustering algorithms that find an approximation of the optimal normalized-cut [22].

In this study we define a probabilistic model for image segmentation given a superpixel map that is based on correlation clustering [9,8]. Correlation clustering has recently been applied to image segmentation. In [17] the correlation clustering model is solved using higher order potentials and LP relaxation. [4] uses an integer linear programming (ILP) branch-and-cut strategy. It was also utilized for computing the ensemble segmentation from a given set of segmentations [3] that is based on the observation that segmentations of the same image are expected to agree on image parts that are clearly separated from the rest of the image and when the segmentations are projected on a superpixel map, the correlation clustering problem can be broken into non-overlapping parts and solved independently. The concept of decomposing image analysis to smaller subproblems is also related to dual decomposition optimization which was recently applied by [25] for image segmentation. In this work we show that unsupervised image segmentation that is based only on local cues can also benefit from decomposing the segmentation problem into sub-problems.

To find the optimal segmentation, based on correlation clustering model, we need to solve an Integer Linear Programm (ILP). The ILP problem is known to be NP hard which has prevented the algorithm from being applied to image segmentation problem. The main contribution of this study is showing that finding the exact global segmentation which is consistent with the local cues, is still tractable. This is done by a careful analysis of the implementation of the general ILP formulation to the image segmentation task.

The rest of this paper is organized as follows. In the next section we review correlation clustering and previous attempts to apply it to image segmentation. Section 3 presents an efficient method to solve the ILP problem and experimental results are shown in Section 4.

2 Correlation Clustering for Image Segmentation

Assume we are given an undirected graph $G = (V, E)$ such that V is the data points $\{1, ..., n\}$ we want to cluster. For each edge $ij \in E$ we are given a symmetric notion of similarity $w_{ij} \in (-\infty, \infty)$ such that a positive weight indicates a local tendency to group i and j into the same cluster and vice versa. The goal of clustering is to divide the data points into several groups such that points in the same group are similar and points in different groups are dissimilar to each other. We want to find a global clustering of the node set V that is most consistent with the local cues. A clustering of a set $\{1, ..., n\}$ can be transformed into a set of n-over-two binary decisions $x = \{x_{ij} | 1 \leq i < j \leq n)\}$ such that $x_{ij} = 1$ if i and j are in the same cluster and 0 otherwise. The correspondence between clusterings and binary decision sets is not one-to-one. Each clustering is represented by a different set of binary decisions but not every set of binary decisions corresponds to a valid clustering. The pairwise relation 'i and j are in the same cluster' is a transitive relation. If i, j and j, k are in the same cluster then necessarily i, k should be in the same cluster. It can be easily verified that the correspondence between clusterings and transitive binary decision sets is one-to-one.

Define the clustering score we want to maximize to be $\sum w_{ij}$, where the summation is over all data pairs that are in the same cluster. Observing that the transitivity constraints are linear, the optimal graph partition is obtained by solving the following Integer Linear Programm (ILP):

$$\begin{aligned} &\max_x \sum_{i<j} w_{ij} x_{ij} && && (1) \\ &s.t. \qquad x_{ij} + x_{jk} - x_{ik} \leq 1 && \forall i, j, k \\ &\qquad\qquad x_{ij} \in \{0, 1\} && \forall i, j \end{aligned}$$

The linear constraint $x_{ij} + x_{jk} - x_{ik} \leq 1$ on the binary variables, enforces transitivity on the binary decisions, i.e., $x_{ij} = x_{jk} = 1$, implies that $x_{ik} = 1$.

There is a simple probabilistic interpretation of the clustering approach described above that motivates the cluster score we optimize. Assume that for each edge $ij \in E$ we are given a probability $p_{ij}(1) = p(x_{ij} = 1)$ that i and j are in the same cluster (the probability that they are in different clusters is denoted by $p_{ij}(0) = 1 - p_{ij}(1)$). Assuming a uniform prior over the clusterings, the posterior probability of a clustering x is:

$$p(x) \propto \prod_{i<j} p_{ij}(x_{ij}) \tag{2}$$

Note that in this simplified probabilistic model the binary local information cues are assumed to be independent. The optimal global clustering which is consistent with the local pairwise evidence, can be found by computing $\arg\max_x p(x)$. It can be easily verified that:

$$\begin{aligned} \log p(x_{ij}) &= \log p_{ij}(1) 1_{\{x_{ij}=1\}} + \log p_{ij}(0) 1_{\{x_{ij}=0\}} \\ &= \log \frac{p_{ij}(1)}{p_{ij}(0)} 1_{\{x_{ij}=1\}} + \log p_{ij}(0) = \log \frac{p_{ij}(1)}{p_{ij}(0)} x_{ij} + \log p_{ij}(0) \end{aligned} \tag{3}$$

Hence,

$$\log p(x) = \sum_{i<j} \log p(x_{ij}) = \sum_{i<j} w_{ij} x_{ij} + \text{const} \tag{4}$$

such that 'const' is a scalar that is not dependent on x and

$$w_{ij} = \log \frac{p_{ij}(1)}{p_{ij}(0)} \tag{5}$$

The best clustering is $\arg\max_x p(x) = \arg\max_x \sum_{i<j} w_{ij} x_{ij}$ such that the maximization is done over all the sets of transitive binary decisions x. Hence the most likely clustering is obtained as the solution of the ILP maximization problem (1).

We can easily incorporate prior knowledge on the clustering x into the ILP framework. Let q be a prior probability that any two points are in the same cluster. For large values of q the optimal clustering tends to have a small number of clusters and vice versa. The modified weight function for the posterior probability is:

$$w_{ij} = \log \frac{p_{ij}(1)}{p_{ij}(0)} + \log \frac{q}{1-q}$$

The graph clustering problem (1) is known as "correlation clustering" [9,8]. This clustering approach has several advantages. It does not require users to specify a parametric form for the clusters, nor to pick the number of clusters. The main drawback of the ILP approach is its high complexity which impedes its applicability for clustering of large sets. The ILP problem (1) is known to be NP-hard [8].

Assume we are given a superpixel map of an image and a similarity measure between each two neighboring superpixels. We can form the segmentation problem as an instance of correlation clustering and solve the ILP (1) to find the optimal segmentation. This segmentation approach, however, is NP-hard and is not tractable for a graph of hundreds or more superpixels. Most of previously suggested graph-based methods for image segmentation try, explicitly or implicitly, to handle this NP-hardness of the ILP problem by either approximate solutions to the ILP clustering problem (e.g. greedy incremental superpixel merging [7]) or find optimal solutions to modified problems (e.g. minimal normalized cut [12]).

A simple approximation approach is to delete all the edges between dissimilar superpixels (i.e., with weights below a predefined threshold), and then look for connected components in the remaining graph. This approach, however, is too local since a single edge with weight above threshold is sufficient to cause two almost separately regions to be merged. Felzenszwalb and Huttenlocher [13] proposed an agglomerative global approach based on constructing a minimum spanning tree. A standard approximate solution of the global ILP problem (1) is obtained by an LP relaxation that replaces the binary constraint $x_{ij} \in [0, 1]$ with the linear constraint $0 \leq x_{ij} \leq 1$ [14,21,24,17]. The LP solution, however, is not binary and it is not clear how to convert it into a binary solution that satisfies transitivity. Given the solution of the relaxed LP problem, the segmentation can be found by considering the connected components obtained by eliminating edges with x_{ij} values below a specified threshold.

In the next sections we show that finding the exact solution for the NP-hard ILP problem (1) is still tractable for image segmentation applied to superpixels.

3 Efficiently Finding the Optimal Segmentation

In this section we describe an efficient method for solving the ILP problem (1) by breaking it into small sub-problems and by incrementally adding transitivity constraints that are not satisfied by the current solution. Assume we are given an undirected weighted graph $G = (V, E)$ such that the vertices $V = \{1, ..., n\}$ are the data points we want to cluster. For each undirected edge $ij \in E$ we are also given a weight $w_{ij} \in (-\infty, \infty)$ such that a positive weight indicates a local tendency to group i and j into the same cluster. The goal is to solve the ILP optimization problem (1). Our approach is based on dividing the problem into smaller problems in which applying standard ILP solvers is still feasible.

We use the following notation. Let $V_1, ..., V_k$ be a partition of V. For each $i, j \in \{1, ..., k\}$, denote $E \cap (V_i \times V_j)$ by E_{ij}. For $i \neq j$, an edge in E_{ij} is called a crossing edge; otherwise the edge is called an internal edge. Denote the set of all the crossing edges by $E_{cross} = \bigcup_{i \neq j} E_{ij}$.

Theorem 1. Assume V can be divided into disjoint subsets $V_1, ..., V_k$ such that there is no edge with a positive weight between members of different subsets (i.e., $w_{ij} \leq 0$ for every $ij \in E_{cross}$). Then the data clustering, which is the optimal solution of the ILP problem (1), is a refinement of the partition $V_1, ..., V_k$ and is obtained by separately applying the ILP optimization on each subset.

Proof. The cost function (1) can be written as a sum of two components:

$$\sum_{ij \in E} w_{ij} x_{ij} = \sum_{ij \in E_{cross}} w_{ij} x_{ij} + \sum_{t=1}^{k} \sum_{ij \in E_{tt}} w_{ij} x_{ij} \tag{6}$$

Eq. (6) decomposes the variables that appear in the cost function (1) into two disjoint subsets. The first set contains the crossing edges and the second set contains the internal edges. Hence, by separately maximizing each one of the two sub-problems, we get an upper bound on the solution of the ILP problem (1). Since we assume that $w_{ij} \leq 0$ for all $(i, j) \in E_{cross}$, the optimal zero-one solution of: max $\sum_{ij \in E_{cross}} w_{ij} x_{ij}$ is obtained by setting $x_{ij} = 0$ for all $(i, j) \in E_{cross}$. Solving an ILP problem on each sub-graph $G_t = (V_t, E_{tt}), \quad t = 1, .., k$ separately:

$$\begin{aligned} &\max \sum_{ij \in E_{tt}} w_{ij} x_{ij} && && (7) \\ &s.t. && x_{ij} + x_{jk} - x_{ik} \leq 1 && \forall i, j, k \in V_t \\ & && x_{ij} \in \{0, 1\} && \forall i, j \in V_t \end{aligned}$$

we get an upper bound on the optimal global solution. It can be easily verified that the combined solution (with $x_{ij} = 0$ for all crossing edges) satisfies all the transitivity constraints in (1) and hence it is optimal.

The most refined partition $V_1, ..., V_k$ that satisfies the requirement of Theorem 1 (no positive weight on crossing edges) can be found by utilizing a greedy approach. We begin with some vertex $v \in V$ defining the initial set of vertices $V_1 = \{v\}$. Then, at each iteration, we look for a positive weight edge (u, v), where $u \in V_1$ and $v \notin V_1$. Then vertex v is brought into V_1. This process is repeated until no vertex can be added to V_1. We next choose a vertex outside of V_1 and start constructing V_2 from the remaining vertices, etc. We call the members of the obtained partition the 'positively connected components' (they are actually the connected components of the graph obtained by eliminating all the non-positive weight edges in the original graph). The complexity of the

Algorithm 1. An efficient solver for the ILP problem (1).

```
Input: A weighted undirected graph $G = (V, E)$ with weights $\{w_{ij}\}$.
Output: A clustering of the graph nodes.

Break the graph into positively connected components $V_1, ..., V_k$.
for $i = 1, ..., k$ do
    Solve the ILP problem restricted to the subset $V_i$ using edge-based variables and
    the cutting-plane method.
end for
The clustering of $V$ is the union of the clusters of its positively connected components.
```

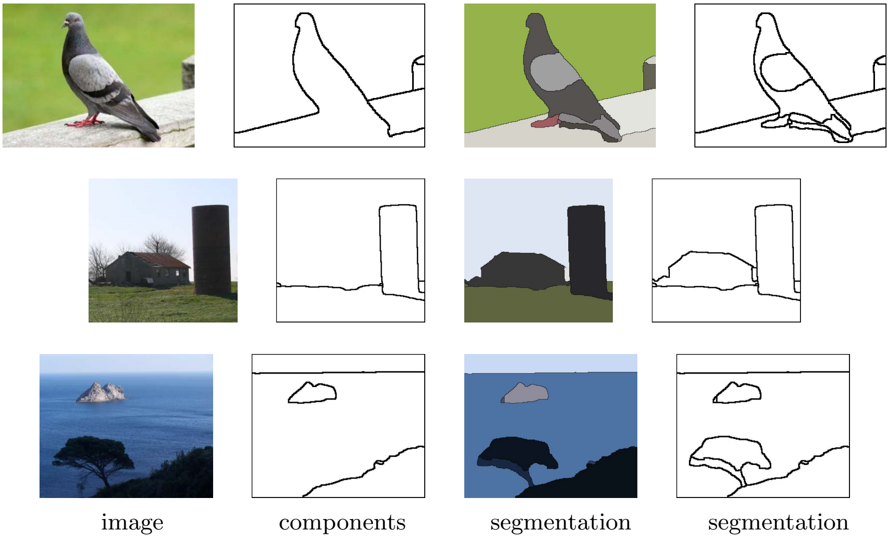


image components segmentation segmentation

Fig. 1. Examples of positively connected components and final segmentations from the Weizmann dataset

algorithm applied to a n-vertex graph is $O(n^2)$. As a result of Theorem 1, we can solve the ILP problem (1) for each positively connected component separately.

For each positively connected component we still need to solve an NP-hard ILP problem that corresponds to correlation clustering restricted to that component. In the case of image segmentation the graph we want to partition is sparse since it is planar and each node has only a small number of spatially adjacent nodes. In case of sparse graph we can formulate the ILP problem much more compactly by associating binary variables only to edges of the graph instead of all the node pairs [4]. The edge labeling consistency constraint can be enforced by adding a linear constraint for each pair of nodes that prevents the situation that the two adjacent nodes are belonging to different clusters but there is a path connecting them in which all the nodes along the path are labeled as connected. The exponential number of such constraints can be implemented using the cutting plane method [16]. The efficient ILP optimization algorithm is summarized in Algorithm-Box 1.

The success of applying the graph partitioning approach described above to image segmentation depends on the existence of image parts that can be separated from the rest of the image. Figures 1 and 3 demonstrates that this is indeed a common situation (implementation details are described in Section 4). In these images we show the positively connected components and the final segmentation that is obtained by solving an ILP problem for each component separately. Therefore, the obtained segmentation is a refinement of the positively connected components partition. We dub the proposed segmentation algorithm "Graph Decomposition ILP Segmentation" (GDIS). The GDIS algorithm was implemented in C. We used the Gurobi software (www.gurobi.com) to solve the ILP optimization sub-problems. Applying the GDIS algorithm on a an image where the size of the largest positively connected component is 1000 takes few seconds.

4 Experimental Results

We present visual and quantitative results of our algorithm for the Weizmann Two-Objects dataset [1] and for the Berkeley BSDS500 dataset [7]. We also show the effect of the efficient ILP algorithm on the segmentation procedure.

4.1 Extracting Superpixels and Local Weights

We used a state-of-the-art superpixel map suggested by Arbelaez et al. [7]. The first step is shifting from pixels to superpixels. The Oriented Watershed Transform (OWT) [7] is used to produce an over-segmentation of the image into a few hundred superpixels. It was observed in [5] that on the average it is enough to represent an image with few hundred superpixels to obtain almost full boundary recall for low enough thresholds.

For each pair of spatially adjacent superpixels we need to obtain (based on the image content) the probability that they are part of the same segment. Arbelaez et al. [7] proposed a similarity measure that combines multiple local cues into a globalization framework based on spectral clustering. The similarity measure takes the form of a logistic-regression that is optimized using an annotated training set. The outcome of this approach is an OWT superpixel map in which each arc pixel (a pixel separating two neighboring superpixels) has a score of being a boundary pixel (a pixel separating two neighboring segments). They refer to this score as the 'globalized probability of boundary' (gPb-owt) [6]. This pixel-level score can be converted into a score between adjacent superpixels by averaging all the scores of the pixels on the corresponding arc. The values of the gPb-owt score increase monotonically with the probability of existing a segmentation boundary but they are not probabilities in the strict sense. Monotonicity is enough for agglomerative clustering that iteratively merges the most similar regions [7]. However, for our approach which avoids agglomerative clustering and is based instead on a global optimization, we need the score to have

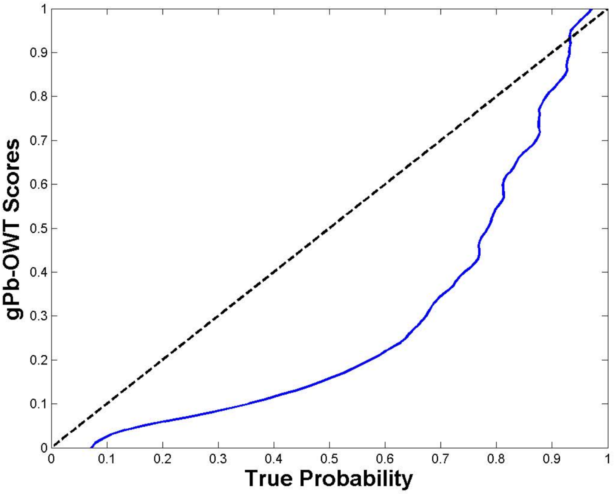


Fig. 2. True segment boundary probabilities vs. the gPb-owt scores

a strict probabilistic interpretation. To convert the gPb-owt score of an arc into a probability to be on a segment boundary, we apply the following procedure. For each arc pixel, using a ground-truth annotation, we can check whether it is on a segment boundary or not. Next, for each value of the gPb-owt score we compute the relative number of arc pixels that have that gPb-owt score and are part of a segmentation boundary. The result of this analysis, performed using the training part of Weizmann database [1]. is shown in Fig. 2. As can be seen, the gPb-owt score indeed increases monotonically but it does not coincide with the exact boundary probability. The graph in Fig. 2 can be used to convert the gPb-owt score into meaningful probability values. Using Eq. (5), the probabilities are converted to weights that are used for the ILP optimization (1) to obtain the final image segmentation.

It is not the focus of our work but there are many other features [10,5,2] and learning methods [17,14] to compute a similarity measure between two neighboring superpixels. Our efficient ILP optimization procedure is also relevant for all these cases.

4.2 Segmentations Results on Weizmann Two-Objects Dataset

The Weizmann Segmentation Dataset consists of 200 images; 100 images with a single object and 100 images with two objects [1]. We used the single object images as our training set for learning the true probability mapping as explained above. The two object images were used as the testing set. The testing procedure we describe next was similar to the one mentioned in [2] using their publicly available testing code [1]. The segmentation results were assessed by their consistency with ground truth segmentation using the F-measure [19]. As in [2]

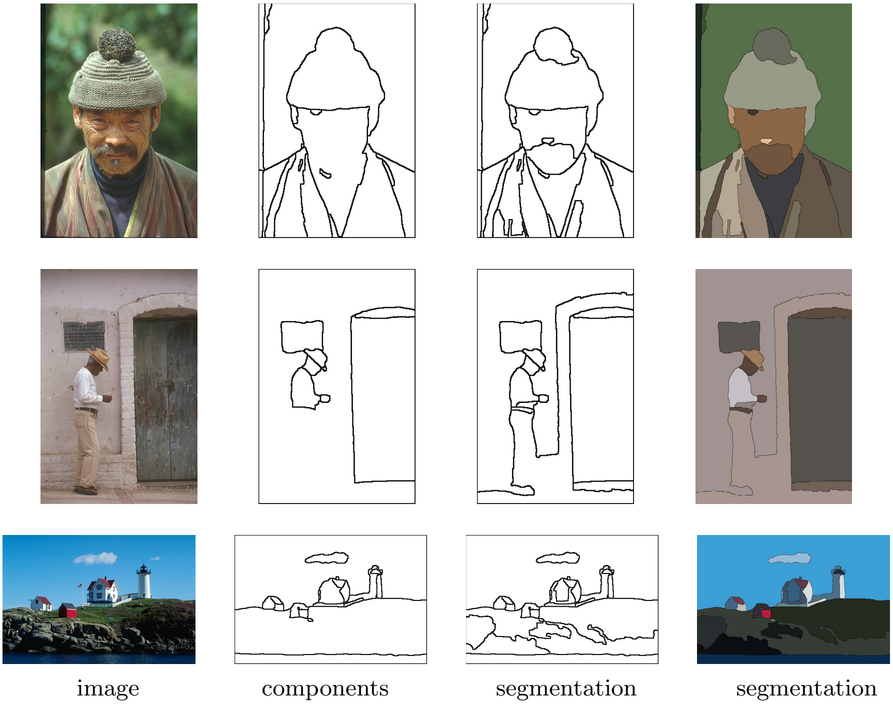


Fig. 3. Examples of positively connected components and final segmentations from the BSDS500 dataset

we selected for each segmentation algorithm the final score that gave the best performance on the Two-Objects Dataset.

In the segmentation experiment, in each run, for each object (of the two objects of each image) we selected separately the best segment that best fit the foreground. The averaged results for both objects are reported in Table 1. As can be seen, the GDIS algorithm scored the highest. Note that the only differences between the implementation of our optimization approach and the UCM [7] are the similarity weight scaling (Figure 2) and the global ILP optimization that we apply instead of the greedy superpixel merging procedure that is done in [7].

4.3 Segmentations Results on BSDS500

Before applying our method on the test portion of the BSDS500 dataset, we converted the gPb-owt scores [7] to probabilities based on the train set of the BSDS500 as explained in section 4.1. We used several standard methods for objective segmentation evaluation: the probabilistic Rand index (PRI) [23], the variation of information (VOI) [20] ,the boundary-based F-measure [19] and the Covering score.

Using the training set we chose the parameters set that scored the highest F-measure for each algorithm. Using the same parameter set, all four measures

Table 1. Average single segment coverage test results on the Weizmann Two-Objects Data Set. Higher is better.

Algorithm	Average F-measure
GDIS	**0.76**
UCM [7]	0.72
Alp [2]	0.68
SWA V1 [15]	0.66
SWA V2 [15]	0.61
Mean Shift [11]	0.61
N-Cuts [18]	0.58

Table 2. Comparison of our method and the UCM segmentations on the BSDS500 test set using four measures: F, PRI, VI(lower is better) and Covering(higher is better)

Algorithm	F	PRI	VI	Covering
GDIS	0.73	0.83	**1.95**	**0.59**
UCM [7]	0.73	0.83	1.97	0.58
PlanarCC [25]	0.72	-	-	-
Kim [17]	0.70	-	-	-

mentioned above were recorded for the testing set. The results for our algorithm and the UCM [7] are shown in Table 2. Table 2 shows that the GDIS also outperforms two other recently introduced approximated graph optimization methods [25,17]. Compared to the UCM, the GDIS scores similar results on F and PRI and only slightly better results with respect to VI and Covering.

To alleviate any confusion, when comparing the UCM results to the ones mentioned in [7], in [7] the different measures mentioned were recorded while optimizing for each measure separately using different results sets. Sample results for the BSDS500 test set are shown in Fig. 3. The fact that the GDIS results are very close to those of the UCM on the BSDS500 is because we use the same superpixels maps and the same underlying similarity score that was tuned on the BSDS500 dataset. It should be emphasized that in contradiction to the UCM which is based on greedy iterative merging, we find the exact global maximum, although it seems as though the UCM even though based on local mergin decision satisfies a global solution.

4.4 Efficiency Analysis of the ILP Algorithm

In this study we present an efficient method for solving the ILP problem (1) by breaking it into small sub-problems. Next, we demonstrate the efficiency contribution of these two elements when applied to an image segmentation task. The complexity of our ILP algorithms depends on the size of the largest component in the decomposition. We computed the following statistics. Next we constructed

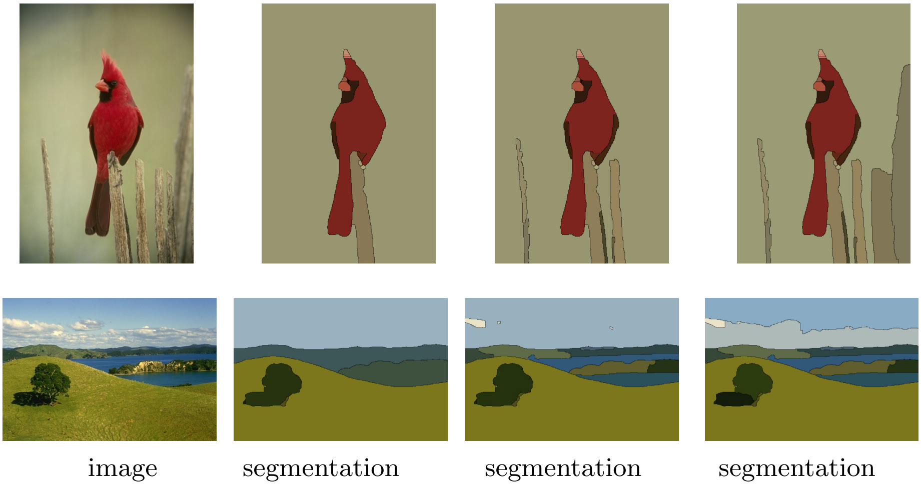


Fig. 4. From left to right: original image followed by three intermediate valid segmentations created as a result of adding more cutting plane constraints (moving to the right). The intermediate segmentations become more refined as we add constraints. Example images were taken from the BSDS500.

the positively connected components and measured the size of the largest component. Fig. 5 shows a histogram of the size of the largest component for the BSDS500. As can be seen, the average size of the largest component is smaller than the number of superpixels in the images. The average size of the superpixel graph for the for BSDS500 is 1160 while the average size of the largest component is 830.

To validate the effect of the cutting plane method we ran it on the BSDS500 dataset and for each instance of applying the (Gurobi) ILP software we measured the number of constraints at the last iteration. Fig. 6a shows the average number of constraints used by the cutting plane method as a function of the number of superpixels in the ILP problem. Note that the total number of constraints is exponential of the problem size. Fig. 6b shows the runtime statistics (measured on Intel Duo-Core, 2.5GHz, 4GB RAM) of the ILP Gurobi software combined with the cutting-plane method applied to positively connected components taken from the BSDS500 images.

4.5 Cutting Plane Intermediate Segmentation Results

The cutting plane algorithm produces an intermediate non-consistent solution. Figure 4 demonstrates on two examples from the BSDS500 the valid segmentations produced by computing the connected components of the intermediate solution. Each intermediate solution is less than the score of the optimal solution which is obtained at the end of the optimization process when the cutting plane method validates that no transitivity constraint is overruled. The intermediate

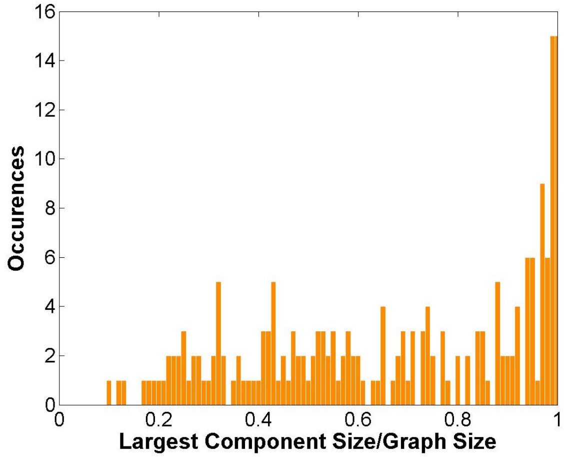


Fig. 5. Histogram of the ILP problem size. Statistics for the BSDS500 test part.

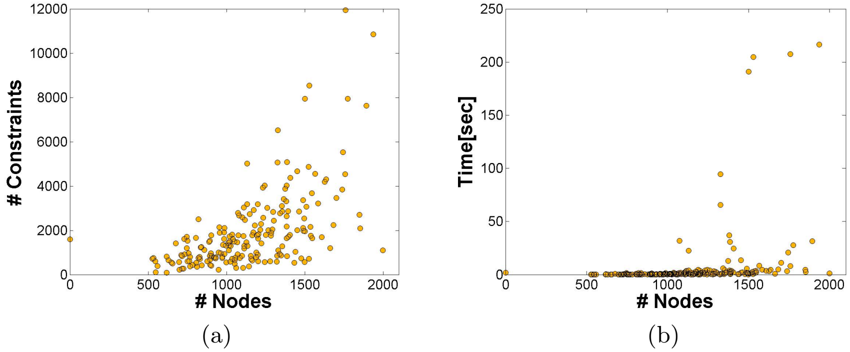


Fig. 6. (a) Number of constraints used in the cutting-plane method as a function of number of graph nodes. (b) Run time of the ILP solver (Gurobi) combined with the cutting-plane method as a function of number of graph nodes. Statistics for the BSDS500 test part.

segmentations usually become more refined at each iteration and as such can be considered as a hierarchical map of segmentations.

To conclude, we have presented a probabilistic modeling for image segmentation based on correlation clustering and an efficient algorithm for the ILP optimization problem. We showed that, given local scores on a map of several hundred superpixels, finding the global segmentation that is most consistent with the local evidence, is still tractable. We then applied the method to a dataset with manually segmented images and compared its performance to several recent algorithms obtaining favorable results. In recent years there was a lot of effort towards extracting better region based features between neighbor superpixels and developing novel machine learning methods to extract better informative similarity weights from those feature. In this study we focused on the global optimization aspect of image segmentation, based on a given superpixel map and local similarly scores between adjacent superpixels. In our implementation we used the probabilistic information score extracted from the gPb-owt score. Exploiting additional content based features from the superpixels as shown in [10,5,2], can be beneficial. The ideas presented in this study can be combined with recent approaches (e.g. [17,10,5]) to further improve segmentation and object detection results.

References

1. Alpert, S., Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. In: IEEE Conf. on Comp. Vision and Pattern Recognition (2007)
2. Alpert, S., Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. IEEE Trans. on Pattern Analysis and Machine Intelligence 34, 315–327 (2012)
3. Alush, A., Goldberger, J.: Ensemble segmentation using efficient integer linear programming. In: IEEE Trans. on Pattern Analysis and Machine Intelligence (2012)
4. Andres, B., Kroeger, T., Briggman, K.L., Denk, W., Korogod, N., Knott, G., Koethe, U., Hamprecht, F.A.: Globally optimal closed-surface segmentation for connectomics. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 778–791. Springer, Heidelberg (2012)
5. Arbelaez, P., Hariharan, B., Gu, C., Gupta, S., Bourdev, L., Malik, J.: Semantic segmentation using regions and parts. In: IEEE Conf. on Comp. Vision and Patt. Recog. (2012)
6. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. In: IEEE Conf. on Computer Vision and Pattern Recognition (2009)
7. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 898–916 (2011)
8. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Machine Learning Journal, 86–113 (2004)
9. Ben-Dor, A., Shamir, R., Yakhini, Z.: Clustering gene expression patterns. Journal of Computational Biology, 281–297 (1999)
10. Carreira, J., Sminchisescu, C.: CPMC: Automatic object segmentation using constrained parametric min-cuts. IEEE Trans. Pattern Anal. Mach. Intell, 1312–1328 (2012)
11. Comanicu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 603–619 (2002)
12. Dhillon, I., Guan, Y., Kulis, B.: Weighted graph cuts without eigenvectors: A multilevel approach. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), pp. 1944–1957 (2007)
13. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. Int. J. of Comp. Vision, 167–181 (2004)

14. Finley, T., Joachims, T.: Supervised clustering with support vector machines. In: Intl. Conf. on Machine Learning (2005)
15. Galun, M., Sharon, E., Basri, R., Brandt, A.: Texture segmentation by multiscale aggregation of filter responses and shape elements. In: IEEE Int. Conf. on Computer Vision (2003)
16. Kelley, J.E.: The cutting-plane method for solving convex programs. Journal of the Society for Industrial Applied Mathematics 8, 703–712 (1960)
17. Kim, S., Nowozin, S., Kohli, P., Yoo, C.D.: Higher-order correlation clustering for image segmentation. In: Neural Information Processing Systems (2011)
18. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. Int. J. Comput. Vision 43, 7–27 (2001)
19. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. In: IEEE Trans. on Pattern Analysis and Machine Intell., pp. 530–549 (2004)
20. Meila, M.: Comparing clusterings: An axiomatic view. In: Int. Conf. on Machine Learning (2005)
21. Nowozin, S., Jegelka, S.: Solution stability in linear programming relaxations: Graph partitioning and unsupervised learning. In: Int. Conf. on Machine Learning, ICML (2009)
22. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 888–905 (2000)
23. Unnikrishnan, R., Pantofaru, C., Hebert, M.: Toward objective evaluation of image segmentation algorithms. IEEE Trans. on Pattern Analysis and Machine Intell, 929–944 (2007)
24. Vitaladevuni, S.N., Basri, R.: Co-clustering of image segments using convex optimization applied to em neuronal reconstruction. In: IEEE Conf. on Computer Vision and Pattern Recognition (2010)
25. Yarkony, J., Ihler, A., Fowlkes, C.C.: Fast planar correlation clustering for image segmentation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 568–581. Springer, Heidelberg (2012)

Multi-task Averaging via Task Clustering

David Martínez-Rego[1] and Massimiliano Pontil[2]

[1] LIDIA Group, Department of Computer Science
University of A Coruña
Campus de Elviña, s/n 15071 A Coruña, Spain
dmartinez@udc.es

[2] Centre for Computational Statistics and Machine Learning
Department of Computer Science, University College London
Malet Place, Gower Street, London, WC1E 6BT
m.pontil@cs.ucl.ac.uk

Abstract. Multi-task averaging deals with the problem of estimating the means of a set of distributions jointly. It has its roots in the fifties when it was observed that leveraging data from related distributions can yield superior performance over learning from each distribution independently. Stein's paradox showed that, in an average square error sense, it is better to estimate the means of T Gaussian random variables using data sampled from all of them. This phenomenon has been largely disregarded and has recently emerged again in the field of multi-task learning. In this paper, we extend recent results for multi-task averaging to the n-dimensional case and propose a method to detect from data which tasks/distributions should be considered as related. Our experimental results indicate that the proposed method compares favorably to the state of the art.

Keywords: multi-task averaging, information theory, spectral clustering.

1 Introduction

Multi-task averaging (MTA) problem can be posed as follows: we have T datasets $\{\mathbf{x}_{t1}, \mathbf{x}_{t2}, \dots, \mathbf{x}_{tN_t}\}$, $t = 1, \dots, T$ each of which is sampled from a fixed but unknown probability distribution (N_t denotes the size of dataset t). Our goal is to estimate the means of each distribution. The first direct approach would be to estimate the means one at a time. However, it turns out that leveraging data from related distributions/tasks[1] can yield superior performance over learning each mean independently. Early evidence of this phenomenon dates back in the fifties from Stein's work, who showed that it is better (in an average square error sense) to estimate each of the means of T Gaussian random variables using data sampled from all of them, even if the random variables are independent

[1] Throughout the paper we use the words "distribution", "task" and "mean" interchangeably.

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 148–159, 2013.

and have different means. This surprising result is often referred to as Stein's paradox [3]. A recent work [4], studies MTA problem in one dimension (that is, taking $\mathbb{R}$ as input space) and presents different optimal results both for MTA mean estimator formula and its hyper-parameters. The proposed estimators are proved to be more accurate than those previously studied in the literature [6,7], but the study of their performance in an n-dimensional space is not treated and the proposed optimal hyper-parameter expression is only valid for the case when all the tasks are related to each other.

In this paper, we study MTA problem in $\mathbb{R}^n$ and also explore the impact of task grouping on the estimation accuracy of the estimation method. We propose optimal formulas for the n-dimensional case and a practical algorithm for task grouping based on information theoretic divergence measures and spectral clustering. When combining these two results, a practical algorithm for MTA in $\mathbb{R}^n$ is obtained. It will be showed that in certain circumstances, when not all the tasks at hand should be considered as related, then the optimal estimators presented in [4] have a null improvement when compared with independent mean estimation for each of the T tasks. On the other hand, we will demonstrate that the proposed method can improve estimation accuracy in an average mean square error sense. These findings may pave the way for more accurate algorithms in a multi-task scenario.

The paper is organized in the following manner. In Section 2, a summary of the key results in [4] are reviewed as the base for the present work. Section 3 presents the extension of the estimators in [4] for the n-dimensional case. Section 4 presents the proposed k-MTA method. In Section 5, we report on our numerical experiments with this method and with previous approaches. Finally, Section 6 contains concluding remarks and suggestions for future work.

2 Background

In the recent paper [4], MTA estimation in $\mathbb{R}$ is presented as the optimal solution to the following convex problem:

$$\mathbf{c}^* = \arg\min_{\mathbf{c}\in\mathbb{R}^T} \left\{ \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N_t} \frac{(x_{ti}-c_t)^2}{\sigma_t^2} + \frac{\gamma}{T^2}\sum_{s,t=1}^{T} A_{st}(c_s-c_t)^2 \right\}$$

where $x_{t1}, x_{t2}, \dots, x_{tN_t}$ are independent and identically distributed (iid) random samples for each task $t = 1, \dots, T$, σ_t^2 is the variance of t-th distribution and $\mathbf{c} = (c_1, \dots, c_T)$ is the vector of means we wish to estimate. Matrix $\mathbf{A} = (A_{st})_{s,t=1}^T$ describes the relatedness or similarity of any pair of the T tasks (with $A_{tt} = 0$ for all t without loss of generality because the diagonal self-similarity terms are canceled in the objective). It can be noted that the proposed MTA objective regularizes the estimates of each of the means, that is, it ties them together. The regularization parameter γ balances the empirical risk (error) and the multi-task regularizer. Note that if $\gamma = 0$, the MTA objective decomposes into T separate minimization problems, producing the simple separate sample averages

$\hat{x}_t = 1/N_t \sum_{i=1}^{N_t} x_{ti}$. Tasks' similarity matrix $\mathbf{A}$ for a specific problem at hand can be specified from the knowledge of a domain expert, but often this side information is not available or it may not be clear how to transform semantic notions of tasks' similarity into an appropriate choice for the values in $\mathbf{A}$. In addition to this difficulty, parameter γ has a great impact on the final result and an optimal choice from a mean square error perspective is desirable. However, the problem of finding an optimal formula for this parameter for a general form of matrix $\mathbf{A}$ is often analytically intractable. In [4], the optimal solution in cases when $\mathbf{A} = a\mathbf{1}\mathbf{1}'$ (called "constant MTA") was found. We restate this result for completeness:

Lemma 1 (constant MTA). *Assume that $\mathbf{A} = a\mathbf{1}\mathbf{1}'$ and $0 < \frac{\sigma_t^2}{N_t} < \infty$ for all t. The optimal $\mathbf{c}^*$ (in terms of mean square error) is given by the formula*

$$\mathbf{c}^* = (I_T + \frac{a}{T}\mathbf{\Sigma} L(\mathbf{1}\mathbf{1}'))^{-1}\hat{\mathbf{x}}$$

where

$$a = \frac{2}{\frac{1}{T(T-1)} \sum_{s,t=1}^{T} (\mu_s - \mu_t)^2}. \tag{1}$$

In the above formula $\mathbf{\Sigma} = \mathrm{diag}(\frac{\sigma_1^2}{N_1}, \ldots, \frac{\sigma_T^2}{N_T})$, $L(\mathbf{A})$ is the Laplacian of matrix $\mathbf{A}$ and μ_t is the true mean of task t. Note that in this result γ is considered equal to 1 without loss of generality.

There are two main issues when applying this lemma in a practical situation. First, the result involves σ_t^2 and μ_t, both quantities which are not known in practice (the second quantity is indeed the one that we are trying to estimate). This issue is solved in [4] using empirical estimates for both quantities and proved to be accurate in practice. Therefore such approach is also used in this paper. The second issue has to do with the form of matrix $\mathbf{A}$ considered in Lemma 1. With $\mathbf{A} = a\mathbf{1}\mathbf{1}'$ we are assuming that all the T tasks are mutually related, which is very unlikely to happen in practice. An analytical result for the case when $T = 2$ proves that the proposed MTA estimation is better than single task estimation only if the true means are close with respect to the variances of their distributions. This observation will be experimentally observed in Section 5. In addition, a closer look at formula (1) shows us that, if far apart tasks are considered as related, the optimal value of parameter a will approximate 0, so that the MTA estimator will bring no benefit.

In order to use the above results in a general case, in addition to extend them to $\mathbb{R}^n$, it is necessary to devise a strategy that, directly from data, estimates which tasks should be considered as related. In the remaining part of the paper we will tackle these problems and demonstrate experimentally that our strategy yields improved results in an average mean square error sense when compared to previous strategies.

3 MTA in High Dimensional Spaces

In this section, we extend the problem presented in [4] to $\mathbb{R}^n$ in a straightforward manner. This will be the first step towards the general MTA algorithm presented in Section 4. MTA in $\mathbb{R}^n$ consist in finding the optimal solution to the problem

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \left\{ \sum_{t=1}^{T} \sum_{i=1}^{N_t} \frac{\|\mathbf{x}_{ti} - \mathbf{c}_t\|^2}{\sigma_t^2} + \frac{\gamma}{2T} \sum_{s,t=1}^{T} A_{st} \|\mathbf{c}_s - \mathbf{c}_t\|^2 \right\} \tag{2}$$

where $\mathbf{c} \in \mathbb{R}^{Tn}$ denotes the vector with all the means $\mathbf{c}_t$, $t = 1, \ldots, T$ concatenated and γ is a hyper-parameter that balances the weighting of the two terms. Problem (2) is similar to equation proposed in [4] but including the 2-norm in $\mathbb{R}^n$ instead of in $\mathbb{R}$. The next two lemmas will be proved in the appendix.

Lemma 2 (MTA in $\mathbb{R}^n$). *The optimal solution of problem (2) is given by*

$$\mathbf{c}^* = ((I_T + \frac{\gamma}{T} \boldsymbol{\Sigma} L(\mathbf{A}))^{-1} \otimes I_n)\hat{\mathbf{x}} \tag{3}$$

where I_T (resp. I_n) is the $T \times T$ (resp. $n \times n$) identity matrix, $\boldsymbol{\Sigma} = \mathrm{diag}(\frac{\sigma_1^2}{N_1}, \ldots, \frac{\sigma_T^2}{N_T})$, $L(\mathbf{A})$ is the Laplacian of matrix $\mathbf{A}$ and $\hat{\mathbf{x}} \in \mathbb{R}^{Tn}$ is the vector of independent means $\hat{\mathbf{x}}_t = \frac{1}{N_t} \sum_{i=1}^{N_t} \mathbf{x}_{ti}$ concatenated in the same order as in $\mathbf{c}^*$.

Lemma 3 (constant MTA in $\mathbb{R}^n$). *Assume that $\mathbf{A} = a\mathbf{1}\mathbf{1}'$ and $0 < \frac{\sigma_t^2}{N_t} < \infty$ for all t. The optimal (in a mean square error sense) mean estimator is given by*

$$\mathbf{c}^* = ((I_T + \frac{a}{T} \boldsymbol{\Sigma} L(\mathbf{1}\mathbf{1}'))^{-1} \otimes I_n)\hat{\mathbf{x}} \tag{4}$$

for

$$a = \frac{2n}{\frac{1}{T(T-1)} \sum_{s,t=1}^{T} \|\mu_s - \mu_t\|^2} \tag{5}$$

where n is the dimension of the input space and μ_t are the true mean vectors of the distributions of each task.

Note that the obtained formulas for the estimator involve the inverse of a matrix which depends neither on the dimension of the space nor on the sample sizes. Hence, its calculation can be done in a very efficient way. Estimators from data of the actual values of μ_t and σ_t^2 in equations (4) and (5) will be used in the practical implementation of these formulas.

4 *k*-MTA: Multi-task Averaging via Information Theoretic Clustering

In this section, k-MTA algorithm is proposed. It is divided in two phases: (a) first, the sets of tasks which should be considered as related are detected via spectral clustering; (b) for each cluster of tasks, equations (5) and (4) are applied separately in order to find the means of each task in the cluster. This approach aims at tackling the limitations of the direct application of the results in [4] when a clustered set of tasks is presented and their respective means are required. Following the results sketched in [4], MTA is only effective when the distance between the true means of the tasks is small when compared to the variance of their distributions. So, for this first phase, we need a measure of divergence between tasks which is able to detect (from data samples) whether the supports of probability distributions largely overlap or not. Based on those similarities, tasks are subsequently clustered and their means estimated. In the next section, we present the divergence measure which will be used in this paper. Subsequently, the spectral clustering algorithm used to construct the groups is described.

4.1 Information Theoretic Tasks' Similarity Measure

The work in [10] considers the quadratic Renyi's entropy as the basic expression for building cost functions for clustering, linear models, and other machine learning problems. The cost and divergence measures developed under the Renyi's entropy framework have been proved effective when dealing with these different learning problems. In particular, the divergence measure between probability density functions (pdfs) called *euclidean pdf distance* is given by

$$D_{ED} = \int\limits_{\mathbb{R}^n} (f(\mathbf{x}) - g(\mathbf{x}))^2 d\mathbf{x}. \tag{6}$$

In the absence of an expression for both f and g, in [10], a parzen estimation using a Gaussian Kernel [11] of both is considered. Using these approximations for f and g, this quantity can be rewritten as:

$$\begin{aligned} D_{ED} &= \int\limits_{\mathbb{R}^n} (f(\mathbf{x}) - g(\mathbf{x}))^2 d\mathbf{x} = \int\limits_{\mathbb{R}^n} f(\mathbf{x})^2 d\mathbf{x} + \int\limits_{\mathbb{R}^n} g(\mathbf{x})^2 d\mathbf{x} - 2\int\limits_{\mathbb{R}^n} f(\mathbf{x})g(\mathbf{x})d\mathbf{x} \\ &= \int\limits_{\mathbb{R}^n} \left(\frac{1}{N}\sum_{i=1}^{N} G_\sigma(\mathbf{x} - \mathbf{x}_i^f)\right)^2 d\mathbf{x} + \int\limits_{\mathbb{R}^n} \left(\frac{1}{M}\sum_{i=1}^{M} G_\sigma(\mathbf{x} - \mathbf{x}_i^g)\right)^2 d\mathbf{x} \\ &- 2\int\limits_{\mathbb{R}^n} \Big(\frac{1}{N}\sum_{i=1}^{N} G_\sigma(\mathbf{x} - \mathbf{x}_i^f)\Big)\Big(\frac{1}{M}\sum_{i=1}^{M} G_\sigma(\mathbf{x} - \mathbf{x}_i^g)\Big) d\mathbf{x} \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G_{\sqrt{2}\sigma}(\mathbf{x}_j^f - \mathbf{x}_i^f)^2 + \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} G_{\sqrt{2}\sigma}(\mathbf{x}_j^g - \mathbf{x}_i^g)^2 \\ &- \frac{2}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M} G_{\sqrt{2}\sigma}(\mathbf{x}_j^g - \mathbf{x}_i^f)^2 = \hat{V}_f + \hat{V}_g - 2\hat{V}_c \end{aligned} \tag{7}$$

where σ is the width of the gaussian kernel and has to be selected. This measure has proven to be an effective way of computing the divergence between two pdfs represented by a sample in many learning scenarios and in this work will be used as the similarity measure between tasks. Specifically, a normalized version of this measure is used

$$D_{ED}^{N}(f,g) = 2 - 2\hat{V}_c/\hat{V}_f\hat{V}_g. \tag{8}$$

This expression still maintains the properties of a divergence and has the advantage of being normalized in the interval $[0,2]$ which will be useful for graph construction in the k-MTA algorithm. Since the clustering technique presented below requires a similarity measure, we transform the aforementioned divergence D_{ED}^{N} into the following similarity measure in the interval $[0,1]$:

$$S_{ij} = \frac{2 - D_{ED}^{N}(f_i,f_j)}{2} \tag{9}$$

4.2 Spectral Clustering

Spectral clustering [14] aims at clustering similar objects $o_i,\ i = 1,\ldots,T$ into k groups given a similarity graph G between all these objects. It can be used virtually with a sample of any kind of items as long as we are given a similarity measure between them. These similarities are used to build a similarity graph G which subsequently is fed into the clustering subroutine. When constructing similarity graphs the goal is to model the local neighborhood relationships between the data points. There are several popular constructions to transform a given set $o_1,\ldots,o_n$ of objects with pairwise similarities S_{ij} into a graph: (a) ϵ-neighborhood, where all points whose pairwise similarities are greater than ϵ are connected; (b) k-nearest neighbor graphs, where if a vertex v_i is among the k-nearest neighbors of v_j those two vertex are connected, and (c) fully connected graph, in which all points are connected with positive similarity given by S_{ij}. In this work we will use ϵ-neighborhood strategy to build the similarity graph.

Once we have the similarity graph G, the graph Laplacian of matrix G is constructed. At this point three main algorithms are proposed in the literature depending on the kind of Laplacian used: unnormalized spectral clustering [14] and the works in [9,13] which use a normalized Laplacian. In this work we will use the version of [13] since it has proved more accurate and stable in practice. Algorithm 1 summarizes the steps of this algorithm (more details can be found in [14]).

4.3 Proposed Algorithm

In this section, we combine the results and components described in previous sections in the proposed algorithm k-MTA. Algorithm 2 summarizes its main steps. First, the task clusters are detected combining the similarity measure presented in Section 4.1 with the spectral clustering algorithm of Section 4.2. Thanks to this step, we will apply the MTA formula derived in Section 3 to the task groups which are similar to each other and we will not blend in tasks which are completely dissimilar, thus avoiding negative transfer.

Algorithm 1. Spectral clustering main steps

```
Input: Similarity matrix $\mathbf{S} \in \mathbb{R}^{T \times T}$, number of clusters $k$, barrier $\epsilon$.
Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j \mid o_j \in C_i\}$

1. Construct a similarity graph $G$ by $\epsilon$-neighborhood based on $S$.
2. Compute the unnormalized Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{G}$.
3. Compute the first $k$ generalized eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ of the genera lized eigenproblem $\mathbf{Lu} = \lambda \mathbf{Du}$.
4. Let $\mathbf{U} \in \mathbb{R}^{T \times k}$ be the matrix containing the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ as columns.
5. Let $\mathbf{y}_i \in \mathbb{R}^k$ be the vector corresponding to the $i$-th row of $\mathbf{U}$ (each $\mathbf{y}_i$ corresponds to each object $o_i$).
6. Cluster the points $\mathbf{y}_i \in \mathbb{R}^k,\ i = 1, \ldots, T$ with the $k$-means algorithm into clusters $A_1, \ldots, A_k$.
```

Algorithm 2. k-MTA algorithm

```
Input: Similarity matrix $\mathbf{S} \in \mathbb{R}^{T \times T}$, number of clusters $k$, barrier $\epsilon$.
Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j \mid o_j \in C_i\}$

1. Construct a similarity graph $G$ by $\epsilon$-neighborhood based on $S$.
2. Compute the unnormalized Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{G}$.
3. Compute the first $k$ generalized eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ of the genera lized eigenproblem $\mathbf{Lu} = \lambda \mathbf{Du}$.
4. Let $\mathbf{U} \in \mathbb{R}^{T \times k}$ be the matrix containing the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ as columns.
5. Let $\mathbf{y}_i \in \mathbb{R}^k$ be the vector corresponding to the $i$-th row of $\mathbf{U}$ (each $\mathbf{y}_i$ corresponds to each object $o_i$).
6. Cluster the points $\mathbf{y}_i \in \mathbb{R}^k,\ i = 1, \ldots, T$ with the $k$-means algorithm into clusters $A_1, \ldots, A_k$.
```

5 Experimental Results

In this section, we explore the performance of k-MTA when compared to its predecessor MTA in [4] and with the single task mean calculation method. To this end, we test the methods on both an artificial dataset which exhibits the behavior of all the methods when clusters of tasks are present, as well as a real dataset where final marks of groups of students are to be predicted.

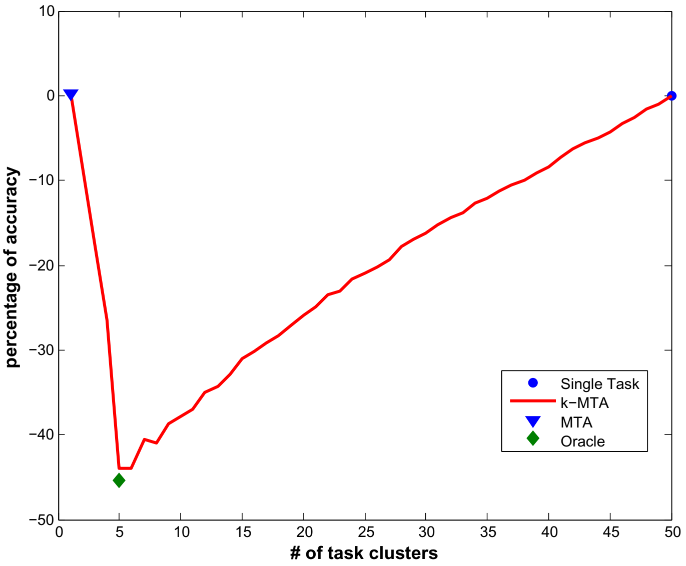


Fig. 1. Mean square distance to the actual means compared to single task result

5.1 Artificial Dataset

The artificial generated dataset has the following properties:

- Number of tasks: 50;
- Number of clusters of tasks: 5;
- Number of task per cluster: 10;
- Input Space: $\mathbb{R}^{10}$;
- Distribution of data: first the means of the Gaussians are selected from 5 Gaussians $\mu_t \sim N(\mu_c \mathbf{1}, \sigma I_d))$ for $\mu_c = \{-10, -5, 0, 5, 10\}$ and $\sigma = 0.1$, where I_d denotes the $d \times d$ identity matrix. Ten centers are selected for each value of μ_c. Then, for each task, a set of iid random data points are generated as $\mathbf{x}_{ti} \sim N(\mu_t, I_d)$.

In this selection, we obtain a convenient distribution of the data for k-MTA since the task are clustered in 5 distant clusters and the expected distance between their centers is small compared to the variance of each task. Figure 1 depicts the average mean square distance from the estimated means to the actual ones compared to the average mean distances obtained with single task means. The results in the figure are the averages of 30 random runs, having 5 data points per task (a scarce sample when compared to the number of parameters to be estimated). The value k of the x axis is the number of clusters k that were configured for k-MTA (optimal ϵ was selected from the interval $[0, 0.5]$). It can be observed how MTA directly applied to the data does not bring any benefit when compared with the single task means while k-MTA obtains an increment of up to -45% when the exact number of clusters is given. Also, the mean result for the oracle, when the correct clustering is always provided to the k-MTA, is shown. It can be observed how the risk of the k-MTA is similar to the oracle one when the correct number of clusters is given as value for k. In addition, it should be noted that even when the number of asked clusters is not exactly the number of actual clusters in the dataset, k-MTA is indeed able to obtain very good accuracy increments.

5.2 School Dataset

The goal of this application is to predict the final class grades $\mu_1, \ldots, \mu_T$ of T students, given only each student's N homework grades $y_{ti},\ i = 1, \ldots, N$. The final class grades include all tests and final exams made by the students but only homework grades are used to predict the final grade. The 16 anonymized datasets were provided by instructors at the University of Washington Department of Electrical Engineering. We consider each class as an experiment and the students in that class the tasks. All grades are normalized in the interval $[0,\ 100]$ and never handed homework was assigned 0 points. For each class, a single pooled variance estimate was used for all tasks. In other words $\sigma_t^2 = \sigma^2$, for every $t = 1, \ldots, T$. Table 1 shows the results obtained when compared with MTA. The reported results are the gains in percentage in final marks prediction when compared with single task means, thus lower value is better.

Table 1. School dataset results

# of stud.	68	69	72	44	50	50	47	16
k-MTA	-37.29	**-38.73 (*)**	-26.92	-36.91	**-18.14 (*)**	-26.58	-8.62	**-1.80 (*)**
MTA	-37.29	-38.42	-26.94	-36.91	3.33	-26.58	-8.62	1.0
# of stud.	29	36	57	48	58	39	149	110
k-MTA	-10.26	-13.99	**-3.82 (*)**	**-12.80 (*)**	-12.35	-5.38	-9.15	-11.52
MTA	-10.26	-13.99	-3.47	-11.53	-12.35	-5.38	-9.15	-11.52

In the table it can be observed that, since k-MTA includes MTA as an special case (when $k = 0$) it has always an equal or better performance than MTA. It is important to note that k-MTA performs better in 5 out of 16 classes and that it always presents a gain with respect to single task means. It is able to obtain a gain even when MTA can not improve single task means. This may be due the presence of clusters in those classes, which are not treated by MTA. In this case, optimal values were selected from the intervals $k = [1, 30]$ and $\epsilon = [0, 0.5]$.

6 Conclusions and Future Work

We have proposed a new algorithm for multi-task averaging. It extends the work in [4] to a n-dimensional space and tackles a key issue when dealing with real data, namely the presence of clusters of related tasks. The algorithm is based on two steps. First, tasks are clustered based on their samples and subsequently MTA is applied for each cluster of tasks. Experimental results show that direct application of MTA in a case where tasks are clustered is useless compared with the results obtained by the single task means. On the other hand, k-MTA is able to detect the underlying clusters of tasks and obtains a significant increment of accuracy. The experiments also suggest that, when dealing with more than two tasks, their relatedness should reflect the similarity between their distributions and this issue should be taken into account when building algorithms like for example multitask one-class classifiers [5,15]. In the future it would be interesting to study extension of the ideas presented here to learn multiple mean embeddings in reproducing kernel Hilbert spaces (see e.g. [2]). Another interesting direction of research is to consider different models of task relatedness and groupping such as in [1,8].

Acknowledgements. This work was supported by Secretaría de Estado de Investigación of the Spanish Government under project TIN 2009-02402, and by the Xunta de Galicia through projects CN2011/007 and CN2012/211, all partially supported by the European Union ERDF. David Martinez-Rego is supported by the FPU Grant Program of the Ministry of Education of the Spanish Government.

A Appendix: Proof of Lemmas

Proof of Lemma 2

We first rewrite the objective function in equation (2) as

$$\sum_{t=1}^{T}(a_t + \frac{N_t}{\sigma_t^2}\|\mathbf{c}_t\|^2 - 2\frac{N_t}{\sigma_t^2}\mathbf{c}_t'\mathbf{c}_t) + \frac{\gamma}{2T}\sum_{s,t=1}^{T} A_{st}(\|\mathbf{c}_s\|^2 + \|\mathbf{c}_t\|^2 - 2\mathbf{c}_s'\mathbf{c}_t)$$

where $a_t := \sum_{i=1}^{N_t} \frac{\|\mathbf{x}_{ti}\|^2}{\sigma_t^2}$

Next we rewrite this equation as in terms of $\mathbf{c} \in \mathbb{R}^{Tn}$ and $\hat{\mathbf{x}} \in \mathbb{R}^{Tn}$ as

$$\sum_{t=1}^{T} \mathbf{a}_t + \mathbf{c}'(\mathbf{\Sigma}^{-1} \otimes I_n)\mathbf{c} - 2\mathbf{c}'(\mathbf{\Sigma}^{-1} \otimes I_n)\hat{\mathbf{x}} + \frac{\gamma}{T}\mathbf{c}'(L(\mathbf{A}) \otimes I_d)\mathbf{c}.$$

Taking the derivative with respect to $\mathbf{c}$ and setting it equal it to $\mathbf{0}$ yields that

$$\mathbf{c}^* = (I_{Tn} + \frac{\gamma}{T}(\mathbf{\Sigma} \otimes I_n)(L(\mathbf{A}) \otimes I_n))^{-1}\hat{\mathbf{x}}.$$

Applying the mixed-product property of the kronecker product to the second term of the inverse, then the associativity of the kronecker product and the inverse property we find that

$$(I_T \otimes I_n + \frac{\gamma}{T}(\mathbf{\Sigma}L(\mathbf{A})) \otimes I_n)^{-1} = ((I_T + \frac{\gamma}{T}\mathbf{\Sigma}L(\mathbf{A}))^{-1} \otimes I_n).$$

The result follows.

A Proof of Lemma 3

Without loss of generality we assume that $\gamma = 1$. Let $\sigma = \frac{\mathrm{tr}(\mathbf{\Sigma})}{T}$ and observe that

$$\begin{aligned}
\mathbf{c}^* &= ((I_T + \frac{a}{T}\mathbf{\Sigma}L(\mathbf{11}'))^{-1} \otimes I_n)\hat{\mathbf{x}} \\
&= ((I_T + \frac{a}{T}\mathbf{\Sigma}(TI_T - \mathbf{11}')^{-1} \otimes I_n)\hat{\mathbf{x}} \\
&= ((I_T + a\mathbf{\Sigma}I_T - \frac{a}{T}\mathbf{\Sigma}\mathbf{11}')^{-1} \otimes I_n)\hat{\mathbf{x}} \\
&= ((I_T + a\mathbf{\Sigma}I_T)^{-1} + \frac{(I_T + a\mathbf{\Sigma}I_T)^{-1}\frac{a}{T}\mathbf{\Sigma}\mathbf{11}'(I_T + a\mathbf{\Sigma}I_T)^{-1}}{1 - \frac{a}{T}\mathbf{1}'(I_T + a\mathbf{\Sigma}I_T)^{-1}\mathbf{\Sigma}\mathbf{1}}) \otimes I_n)\hat{\mathbf{x}} \\
&= (\frac{1}{a\sigma + 1}\left(I_T + a\frac{\sigma}{T}\mathbf{11}'\right) \otimes I_n)\hat{\mathbf{x}}
\end{aligned}$$

where we have made use of the Sherman-Morrison formula for the inverse and omitted some tedious algebra. We will call the matrix on the right-hand side $\mathbf{Z}$ when substituting.

Next, we define the expression for the expected mean square error of an estimator of the form $\mathbf{W}\hat{\mathbf{x}}$ of a mean vector μ, where $\hat{\mathbf{x}}$ is the simple average of each task. We have that:

$$\begin{aligned}
R(\mathbf{W}\hat{\mathbf{x}}, \mu) &= E(\|\mathbf{W}\hat{\mathbf{x}} - \mu\|^2) \\
&= E((\mathbf{W}\hat{\mathbf{x}} - \mu)'(\mathbf{W}\hat{\mathbf{x}} - \mu)) \\
&= \mathrm{tr}(\mathbf{W\Sigma W}') + \mu'(\mathbf{W} - I)'(\mathbf{W} - I)\mu
\end{aligned}$$

where the expected value is taken with respect to the random sample and μ and $\mathbf{\Sigma}$ are the actual mean and covariance of the distribution. In this work we will suppose that all the distributions have an isotropic diagonal covariance matrix so we can use this expression with $\mu \in R^{Tn}$ and covariance matrix $\mathbf{\Sigma} = \mathbf{\Sigma}_T \otimes I_n$ with $\mathbf{\Sigma_T} = \mathrm{diag}(\frac{\sigma_1^2}{N_1}, \ldots, \frac{\sigma_T^2}{N_T})$. If we substitute the optimal expression for $\mathbf{W}$ in this expression we have that:

$$\begin{aligned}
R(\mathbf{W}\hat{\mathbf{x}}, \mu) &= \mathrm{tr}((\mathbf{Z} \otimes I_n)\mathbf{\Sigma}(\mathbf{Z} \otimes I_n)') + \mu'((\mathbf{Z} \otimes I_n) - I_{Tn})'((\mathbf{Z} \otimes I_n) - I_{Tn})\mu \\
&= \mathrm{tr}((\mathbf{Z\Sigma Z}') \otimes I_n) + \mu'((\mathbf{Z} - I_T)'(\mathbf{Z} - I_T) \otimes I_n)\mu \\
&= \mathrm{tr}(\mathbf{Z\Sigma Z}')\mathrm{tr}(I_n) + \mu'((\mathbf{Z} - I_T)'(\mathbf{Z} - I_T) \otimes I_n)\mu \\
&= n\left[\frac{\sigma}{(a\sigma + 1)^2}(T + 2a\sigma + (a\sigma)^2)\right] + \frac{(a\sigma)^2}{(a\sigma + 1)^2}\mu'\left[L(\frac{1}{T}\mathbf{11}') \otimes I_n\right]\mu
\end{aligned}$$

where $\sigma = \frac{\mathrm{tr}(\mathbf{\Sigma})}{T}$, we have used the idempotency of matrix $L(\frac{1}{T}\mathbf{11}')$ and omitted some tedious algebra in the last step. The derivative of this expression with respect to a is given by

$$\frac{\delta R((\mathbf{Z} \otimes I_n)\hat{\mathbf{x}}, \mu)}{\delta a} = \frac{2\sigma^2[(1 - T)n + a\mu'\left[L(\frac{1}{T}\mathbf{11}') \otimes I_n\right]\mu]}{(a\sigma + 1)^3}. \tag{10}$$

In order for this expression to be equal to zero, the numerator must be zero. The result follows.

References

1. Argyriou, A., Maurer, A., Pontil, M.: An algorithm for transfer learning in a heterogeneous environment. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 71–85. Springer, Heidelberg (2008)
2. Gretton, A., Sriperumbudur, B., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., Fukumizu, K.: Optimal kernel choice for large-scale two-sample tests. In: Advances in Neural Information Processing Systems 25, pp. 1214–1222 (2012)
3. Efron, B., Morris, C.N.: Stein's paradox in statistics. Scientific American 236(5), 119–127 (1977)
4. Feldman, S., Gupta, M.R., Frigyik, B.A.: Multi-task averaging. In: Advances in Neural Information Processing Systems 25, pp. 1178–1186 (2012); MMM This paper seems obscure. can you add volume, page number etc? I would otherwise remove this citation unless strictly needed PPP
5. He, X., Mourot, G., Maquin, D., Ragot, J.: One-class SVM in multi-task learning. In: Advances in Safety, Reliability and Risk Management (2012)
6. James, W., Stein, C.: Estimation with quadratic loss. In: Proc. Fourth Berkeley Symposium on Mathematical Statistics and Probability, pp. 361–379 (1961)
7. Lehmann, E.L., Casella, G.: Theory of Point Estimation. Springer, New York (1998)
8. Maurer, A., Pontil, M., Romera-Paredes, B.: Sparse coding for multitask and transfer learning. In: Proceedings of the 30th International Conference on Machine Learning (2013)
9. Ng, A.Y., Jordan, M., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems (2002)
10. Principe, J.C.: Information Theoretic Learning. Renyi's Entropy and Kernel Perspectives. Springer (2000)
11. Parzen, E.: On Estimation of a Probability Density Function and Mode. Annals of Mathematics and Statistics 33(3), 1065–1076 (1962)
12. Stein, C.: Inadmissibility of the usual estimator for the mean of a multivariate distribution. In: Proc. Third Berkeley Symposium on Mathematical Statistics and Probability, pp. 197–206 (1956)
13. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
14. von Luxburg, U.: A Tutorial on Spectral Clustering. Statistics and Computing 17(4), 395–416 (2007)
15. Yang, H., King, I., Lyu, M.R.: Multi-task Learning for One-class Classification. In: International Joint Conference on Neural Networks (2010)

Modeling and Detecting Community Hierarchies

Maria Florina Balcan and Yingyu Liang

School of Computer Science
Georgia Institute of Technology
Atlanta, GA 30332

Abstract. Community detection has in recent years emerged as an invaluable tool for describing and quantifying interactions in networks. In this paper we propose a theoretical model that explicitly formalizes both the tight connections within each community and the hierarchical nature of the communities. We further present an efficient algorithm that provably detects all the communities in our model. Experiments demonstrate that our definition successfully models real world communities, and our algorithm compares favorably with existing approaches.

Keywords: community detection, hierarchical structure.

1 Introduction

The structure of networks has been extensively studied over the past several years in many disciplines, ranging from mathematics and computer science to sociology and biology. A significant amount of recent work in this area has focused on the development of community detection algorithms. The community structure reflects how entities in a network form meaningful groups such that interactions within the groups are more active compared to those between the groups and the outside world. The discovery of these communities is useful for understanding the structure of the underlying network, or making decisions in the network [8,9,28,29].

Generally, a community should be thought of as a subset whose members have more interactions with each other than with the remainder of the network. This intuition is captured by some recently proposed models [2,3,1,12,15]. Additionally, recent studies show that networks often exhibit hierarchical organization, in which communities can contain groups of sub-communities, and so forth over multiple scales. For example, this can be observed in ecological niches in food webs, modules in biochemical networks or groups of common interest in social websites [31,19,7]. It is also shown empirically and theoretically that hierarchical structures can simultaneously explain and quantitatively reproduce many commonly observed topological properties of networks [6,32,10]. This suggests that the hierarchical structure should also be reflected when modeling real world communities.

Although some heuristic approaches [10,21] have been proposed to detect community hierarchies, few works have formalized this hierarchical property,

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 160–175, 2013.

and there are no theoretical performance guarantees for the algorithms. Inspired by the related work in clustering [4], in this paper we define a notion of communities that both reflects the tight connections within communities and explicitly models the hierarchy of communities. In our model, each member of a community falls into a sub-community, and the sub-communities within this community have active interactions with each other while entities outside this community have fewer interactions with members inside. Given this formalization, we then propose an efficient algorithm that detects all the communities in this model, and prove that all the communities form a hierarchy. Empirical evaluations demonstrate that our formalization successfully models real world communities, and our algorithm compares favorably with existing approaches.

In the remainder of the paper, we formalize our model in Section 2, and then describe and analyze our algorithm in Section 3. We then present the results of our experiments in Section 4, and conclude our paper in Section 5.

2 Hierarchical Community Model

A network is typically represented as a graph $G = (V, E)$ on a set of $n = |V|$ points[1], where the edges could be undirected or directed, unweighted or weighted. The graph implicitly specifies a neighborhood structure on the points, i.e. for each point there is a ranking of all other points according to the level of possible interaction. More precisely, we assume that we have a neighborhood function N which given a point p and a threshold t outputs a list $N_t(p)$ containing the t nearest neighbors of p in V.

The neighborhood function can be used to formalize a model of hierarchical communities. Using this neighborhood function, the tight connections within communities can be naturally rephrased as follows: for suitable t, most points p in the community have most of the nearest neighbors $N_t(p)$ from the community while points outside have just a few nearest neighbors from the community. Besides this, we also want to formalize the hierarchical structure that sub-communities in a lower, more local level actively interacting with each other form a community in a higher, more global level. The connections between the sub-communities can also be rephrased using the language of neighborhood: a majority of points in each sub-community have most of the nearest neighbors from the sub-communities in the same community.

In the remainder of the section, we specify our model based on the neighborhood function. We begin with the following notion of compact blobs, which will serve as a building block for our model.

Definition 1. *A subset A of points is called an α-compact blob, if out of the $|A|$ nearest neighbors:*

- *any point $p \in A$ has at most αn neighbors outside A, i.e. $|N_{|A|}(p) \setminus A| \leq \alpha n$;*
- *any point $q \notin A$ has at most αn neighbors inside A, i.e. $|N_{|A|}(q) \cap A| \leq \alpha n$.*

[1] We distinguish the nodes in the hierarchy our algorithm builds from the points in the graph.

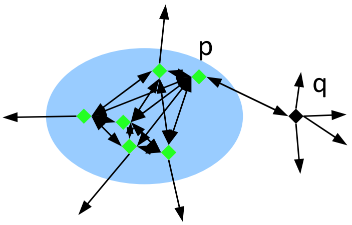


Fig. 1. Illustration of an α-compact blob. An edge (x, y) means that y is one of x's nearest neighbors.

Note that the notion of compact blobs is the same as the clusters that satisfy the α-good neighborhood property defined in [4]. The notion captures the desired property of communities to be detected: members in the community have many more interactions with other members inside the community and have fewer interactions with those outside. However, in practice, the notion may seem somewhat restricted. First, it requires all the members in the community have most interactions with other members inside the community, which may not be the case in real life. For example, some members in the boundary may have more interactions with the outside world, i.e. they have more than αn neighbors from outside. Based on this consideration, we define the (α, β)-stable property as follows.

Definition 2. *A community C is (α, β)-stable if*

- *any point $p \in C$ falls into a α-compact blob $A_p \subseteq C$ of size greater than $6\alpha n$,*
- *for any point $p \in C$, at least β fraction of points in A_p have all but at most αn nearest neighbors from C out of their $|C|$ nearest neighbors,*
- *any point q outside C has at most αn nearest neighbors from C out of their $|C|$ nearest neighbors.*

Informally, the first condition means that every point falls into a sufficiently large compact blob in its community. This condition formalizes the local neighborhood structure that each member interacts actively with sufficiently many members in the community. Note that the compact blob should be large enough so that the membership of the point is clearly established, i.e. it should have size comparable to αn, the number of connections to points outside. Here we choose a minimum size of $6\alpha n$ mainly because it guarantees that our algorithm can still identify the blob in the worst case. The second condition means that at least β fraction of points in these compact blobs have most of their nearest neighbors from the community. This condition formalizes more global neighborhood structure about how the compact blobs interact with each other to form a community. The third condition formalizes how the community is separated from the outside.

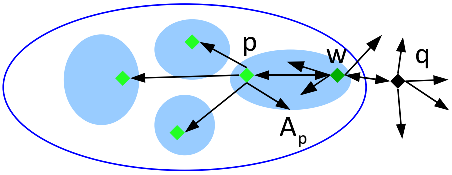


Fig. 2. Illustration of an (α, β)-stable community. An edge (x, y) means that y is one of x's nearest neighbors. Note that point w lies on the "boundary" of the community. It falls into the compact blob A_p, but does not have most of its nearest neighbors from the community.

Note that we no longer require all the members in the community have most interactions inside; we only require each member interacts with sufficiently many members and a majority of members in these local groups interact actively. Also note that the definition is hierarchical in nature: sufficiently large compact blobs clearly satisfy the definition of (α, β)-stable property and thus can be viewed as communities in lower levels. Furthermore, in the next section we will show that all the (α, β)-stable communities form a hierarchy. We show this by presenting an algorithm and proving that each (α, β)-stable community is a node in the hierarchy output by the algorithm. So our formulation explicitly models the hierarchical structure of communities observed in networks.

Next we propose a further generalization that considers possible noise in real world data. There may be some abnormal points that do not exhibit clear membership to any community, in the presence of which our definition above does not model the communities well. For example, suppose there is a point that has connections to all other points in the network, then no non-trivial subsets satisfy our definition above. We call such points bad since they do not fit into our community model above. To deal with the noise, we can naturally relax the (α, β)-stable property to the (α, β, ν)-stable property defined as follows. Informally, it requires that the target community satisfies the (α, β)-stable property after removing a few bad points B. For convenience, we call the other points in $S \setminus B$ good points.

Definition 3. *A community C is (α, β, ν)-stable if there exist a subset of bad points B of size at most νn, such that*

- *any good point $p \in G = C \setminus B$ falls into a compact blob $A_p \subseteq C$ of size greater than $6(\alpha + \nu)n$,*
- *for any point $p \in G$, at least β fraction of points in A_p have all but at most αn nearest neighbors from G out of their $|G|$ nearest neighbors in $S \setminus B$,*
- *any good point q outside $C \cup B$ has at most αn nearest neighbors from G out of their $|G|$ nearest neighbors in $S \setminus B$.*

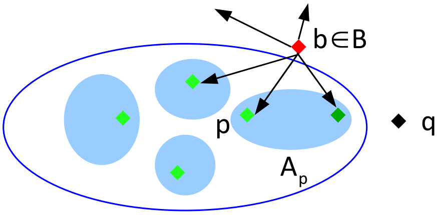


Fig. 3. Illustration of an (α, β, ν)-stable community. An edge (x, y) means that y is one of x's nearest neighbors. Point b is a bad point and does not exhibit clear membership to any community.

Note 1. The parameters α, ν are defined globally, i.e. they are defined as ratios with respect to the total number of points. So a local change to some community can affect the values of these parameters for the other communities. For example, suppose we add Kn new points to some community, with all the new points having neighbors only inside this special community. Since the number of points increases to $(K+1)n$, the communities outside the modified community are now $(\alpha/(1+K), \beta, \nu/(1+K))$-stable. However, the local change does not affect the identifiability of these communities. Our algorithm described in the next section can still detect these communities, given the value of $(\alpha + \nu)n$.

Note 2. The input of the community detection task is usually a graph representing the network, and there are different ways to lift the graph to a neighborhood function. The simplest one is to directly sort for each point p all the other points q according to the weights of the edges (p, q) and break ties randomly (we assume without loss of generality that the weights are in $[0, 1]$ and the weight of an edge not in E is regarded as 0). However, as pointed out in [3], we also have alternative approaches to convert the observed graph into a neighborhood function. More specifically, we assume the observed graph reflects some underlying unobserved set of relations, and thus we can lift the graph to an affinity system based on various beliefs about the connection between the latent relations and the observed graph, and then sort the points according to the affinity system to get the neighborhood function. For example, based on the belief that random walks on the graph can reflect the similarities between entities, we can define the affinity to be the diffusion kernel $\exp\{\lambda A\}$ where A is the adjacent matrix and λ is a parameter. Note that the results of appropriate lifting procedures can better reflect the true relationships between entities, and thus the conversion can address the challenging issue of sparsity in the observed graph.

3 Hierarchical Community Detection Algorithm

In the section, we propose an algorithm for detecting communities satisfying the (α, β, ν)-stable property. The goal of our algorithm is to output a set of

Algorithm 1. Hierarchical Community Detection Algorithm

```
Input: neighborhood function $N$ on a set of points $V$, $n = |V|$, $\alpha > 0, \nu > 0$.

Step 1  Initialize $\mathcal{C}'$ to be a set of singleton points, and $t = 6(\alpha+\nu)n + 1$.
        while $|\mathcal{C}'| > 1$ do
Step 2    Build $F_t$ on $V$ as follows.
          for any $x, y \in V$ that satisfy $|N_t(x) \cap N_t(y)| \geq t - 2(\alpha+\nu)n$ do
            Connect $x, y$ in $F_t$.
Step 3    Build $H_t$ on $\mathcal{C}'$ as follows. Let $N_F(x)$ denote the neighbors of $x$ in $F_t$.
          for any $U, W \in \mathcal{C}'$ do
            if $U, W$ are singleton subsets, i.e. $U = \{x\}, W = \{y\}$ then
              Connect $U, W$ in $H_t$, if $|N_F(x) \cap N_F(y)| > (\alpha+\nu)n$.
            else
              Set $S_t(x, y) = |N_F(x) \cap N_F(y) \cap (U \cup W)|$, $\forall x \in U, y \in W$.
              Connect $U, W$ in $H_t$, if $\text{median}_{x \in U, y \in W} S_t(x, y) > \frac{|U|+|W|}{4}$.
Step 4    for any component $R$ in $H_t$ that satisfies $|\bigcup_{C \in R} C| \geq 4(\alpha+\nu)n$ do
            Update $\mathcal{C}'$ by merging subsets in $R$ into one subset.
Step 5    $t = t + 1$.
        end while

Output: Hierarchy $T$ with single points as leaves and internal nodes corresponding
to the merges performed.
```

communities such that each community satisfying the (α, β, ν)-stable property is close to one in the output. To be precise, we say that a community C is ν-close to another community C' if $|C \setminus C'| + |C' \setminus C| \leq \nu n$. We first describe the details in Algorithm 1, and then present the analysis in Theorem 1.

Now we prove that the algorithm successfully outputs a hierarchy such that any community satisfying the (α, β, ν)-stable property with sufficiently large β is close to one of the nodes in the hierarchy. Formally,

Theorem 1. *Algorithm 1 outputs a hierarchy such that any community satisfying the (α, β, ν)-stable property with $\beta \geq 5/6$ is ν-close to a node in the hierarchy. The algorithm runs in time $O(n^{\omega+1})$, where $O(n^{\omega})$ is the state of the art for matrix multiplication.*

The correctness of the theorem follows from Lemma 3 and the running time follows from Lemma 4. In the following analysis, we always assume $\beta \geq 5/6$. Before presenting the analysis for the general communities in Lemma 3, we first prove a lemma for the base case of compact blobs, showing that for any compact blob, a node close to it will be formed.

Lemma 1. *For any good point p, when $t \leq |A_p|$, good points from A_p will not be merged with good points outside A_p. At the end of the threshold $t = |A_p|$, all points in A_p have been merged into a subset.*

Proof. We prove this by induction on t. The claim is clearly true initially. Now assume for induction that at the beginning of a threshold $t \leq |A_p|$, in $\mathcal{C}'$ good points from A_p are not merged with good points outside A_p, i.e. any subset can contain good points from only one of A_p and $V \setminus B \setminus A_p$. We now analyze the properties of the graphs F_t and H_t, and show that at the end of the current threshold, the claim is still true.

First, as long as $t \leq |A_p|$, the graph F_t has the following properties.

– No good point x in A_p is connected to a good point y outside A_p. By the definition of compact blobs, out of the t nearest neighbors, x has at most $(\alpha + \nu)n$ neighbors outside A_p. For $y \in V \setminus B \setminus A_p$, y has at most $(\alpha + \nu)n$ neighbors in A_p. Then x, y have at most $2(\alpha + \nu)n < t - 2(\alpha + \nu)n$ common neighbors, so they are not connected.

– No bad point z is connected to both a good point x in A_p and a good point y outside A_p. We know that out of the t nearest neighbors, x has at most $(\alpha + \nu)n$ neighbors outside A_p. So if z is connected to x, then z must have more than $t - 3(\alpha + \nu)n$ neighbors in A_p and less than $3(\alpha + \nu)n$ neighbors outside A_p. Since y has at most $(\alpha + \nu)n$ neighbors in A_p, we have that y, z share less than $3(\alpha + \nu)n + (\alpha + \nu)n < t - 2(\alpha + \nu)n$ neighbors, so they are not connected.

Based on the properties of F_t and the inductive assumption that any subset can contain good points from only one of A_p and $V \setminus B \setminus A_p$, we show that the graph H_t has the following properties.

– No subset U containing good points from A_p is connected to a subset W containing good points outside A_p. This is clearly true if they are singleton subsets. In the other cases, note that the fraction of bad points in U or W is at most 1/4. Then the number of pairs (x, y) with good points $x \in U$ and $y \in W$ is at least $\frac{3}{4}|U| \times \frac{3}{4}|W| > |U||W|/2$, i.e. more than half of the pairs (x, y) with $x \in U$ and $y \in W$ are pairs of good points. This means there exist good points $x^* \in U, y^* \in W$ such that $S_t(x^*, y^*)$ is no less than $\text{median}_{x \in U, y \in W} S_t(x, y)$. By the properties of F_t, x^*, y^* have no common neighbors. Therefore, U and W are not connected.

– If a subset W contains only bad points, then it cannot be connected to both a subset containing good points from A_p and a subset containing good points outside A_p. Suppose it is connected to U which contains good points from A_p. Note that since W contains only bad points, it must contain only a single point z. If $U = \{x\}$ is singleton, then x, z share more than $(\alpha + \nu)n$ neighbors in F_t. Since in F_t, x is only connected to good points from A_p and bad points, z and x must share some common neighbors from A_p, then z must be connected to some good points in A_p. In the other cases, note that the fraction of bad points in U is at most 1/4. So there exists a good point $x^* \in U$ such that $S_t(x^*, z) \geq \text{median}_{x \in U} S_t(x, z)$. Then we have $S_t(x^*, z) > (|U| + |W|)/4 > \nu n$, and thus z must also be connected to some good points in A_p. Similarly, if W is connected to a subset containing good points outside A_p, then the point in W must connect to some good point outside A_p. But this is contradictory to the fact that in F_t no bad point is connected to both a good point in A_p and a good point outside A_p.

By the properties of H_t, no connected component contains both good points in A_p and good points outside A_p. So at the end of this threshold t, the claim is still true. Then by induction, we know that when $t \leq |A_p|$, we will not merge good points from A_p with good points outside A_p.

Next we show that at the end of the threshold $t = |A_p|$, we will merge all points in A_p into a subset. First, at this threshold, all good points in A_p are connected in F_t. Any good point in A_p has at most $(\alpha + \nu)n$ neighbors outside A_p, so when $t = |A_p|$, any two good points x, y in A_p are connected, and thus they share at least $|A_p|$ common neighbors in F_t. Second, all subsets containing good points in A_p are connected in H_t. If no good points in A_p have been merged, then these singleton points will be connected in H_t since they share at least $|A_p|$ singleton subsets as common neighbors in F_t. If some good points in A_p have already been merged into non-singleton subsets, we can show that in H_t these non-singleton subsets will be connected to each other and connected to singleton subsets containing good points from A_p. For any such pair of subsets U and W, the fraction of bad points in U or W is at most $1/4$, so there exist good points $x^* \in U, y^* \in W$ such that $\text{median}_{x \in U, y \in W} S_t(x, y)$ is no less than $S_t(x^*, y^*)$. Since x^*, y^* are connected to all good points in A_p in F_t, $S_t(x^*, y^*)$ is no less than the number of good points in U and W. So $\text{median}_{x \in U, y \in W} S_t(x, y) \geq S_t(x^*, y^*) > (|U| + |W|)/4$, and thus U, W are connected in H_t. Therefore, all points in A_p are merged into a subset. □

The following is a consequence of Lemma 1, which will be used in the analysis for the general communities in Lemma 3.

Lemma 2. *In Algorithm 1, if a subset U satisfies that for any good point $p \in U$, $A_p \subseteq U$, then there exist a subset of good points $P \subseteq U$, such that $\{A_p : p \in P\}$ is a partition of $U \setminus B$.*

Proof. We have $U \setminus B = \cup_{p \in U \setminus B} A_p$. We only need to show that sets in $\{A_p : p \in U \setminus B\}$ are laminar, i.e. for any $p, q \in U \setminus B$, either $A_p \cap A_q = \emptyset$ or $A_p \subseteq A_q$ or $A_q \subseteq A_p$. Assume for contradiction that there exist A_p and A_q such that $A_p \setminus A_q \neq \emptyset, A_q \setminus A_p \neq \emptyset$ and $A_p \cap A_q \neq \emptyset$. Without loss of generality, suppose $|A_p| \leq |A_q|$. Then by Lemma 1, at the end of the threshold $t = |A_p|$, we have merged all good points in A_p into a subset. Specifically, this means that we have merged $A_p \cap A_q$ with $A_p \setminus A_q$. So for $t \leq |A_q|$, we have merged good points in A_q with good points outside A_q, which is contradictory to Lemma 1. □

By the above lemmas, for any good point p, the subset A_p will be formed before points in it are merged with good points outside. Once these subsets are formed, we can show that subsets in the same target community will be merged together before they are merged with those from other communities, and thus the hierarchy produced has a node close to the target community. Formally, we have the following result.

Lemma 3. *For any community C satisfying the (α, β, ν)-stable property with $\beta \geq 5/6$, $\mathcal{C}' \setminus B$ in Algorithm 1 is always laminar to $C \setminus B$, i.e. for any $C' \in \mathcal{C}'$,* *either* $(C' \setminus B) \cap (C \setminus B) = \emptyset$ *or* $(C' \setminus B) \subseteq (C \setminus B)$ *or* $(C \setminus B) \subseteq (C' \setminus B)$. *Furthermore, there is a node* U *in the hierarchy produced such that* $U \setminus B = C \setminus B$.

Proof. we will show by induction on t that: for any community C satisfying the (α, β, ν)-stable property with $\beta \geq 5/6$,

- at the end of threshold t, $\mathcal{C}' \setminus B$ is laminar to $C \setminus B$,
- at the end of threshold t, for any C such that $|C \setminus B| \leq t$, we have merged all points in $C \setminus B$ into a subset.

These claims are clearly true initially. Assume for induction that they are true for the threshold $t - 1$, we now show that they are also true for the threshold t.

We first show that the laminarity is preserved. The laminarity is broken only when we connect in H_t two subsets U, W such that U is a strict subset of C after removing the bad points, and W is a subset containing good points from outside. If there is a good point $p \in U$ such that $A_p \not\subseteq U$, then by Lemma 1, they cannot be connected. So we only need to consider the other case when for any good point $p \in U, A_p \subseteq U$. For convenience, we call a point great if it is a good point in C, and it has less than αn neighbors outside $C \setminus B$ out of the $|C \setminus B|$ nearest neighbors in $V \setminus B$. We now show that U, W are not connected in H_t. Since $U \setminus B$ is a strict subset of $C \setminus B$, by induction on the second claim, we have $t \leq |C \setminus B|$. Then great points in U and points in W share at most $2(\alpha + \nu)n < t - 2(\alpha + \nu)n$ common neighbors, so they are not connected in F_t. By Lemma 2 and the second condition of the (α, β, ν)-stable property, we know that at least 5/6 fraction of points in $U \setminus B$ are great points. Then there exist a great point $x^* \in U$ and a point $y^* \in W$ such that $S_t(x^*, y^*)$ is no less than $\mathrm{median}_{x \in U, y \in W} S_t(x, y)$. Since in F_t great points in U are not connected to points in W, we have $S_t(x^*, y^*) \leq (|U| + |W|)/4$. So $\mathrm{median}_{x \in U, y \in W} S_t(x, y) \leq (|U| + |W|)/4$ and U, W are not connected in H_t. Therefore, the laminarity is preserved.

Next we show that at the end of the threshold $t = |C \setminus B|$, all points in $C \setminus B$ are merged into a subset. By Lemma 1, all good points in $C \setminus B$ are now in sufficiently large subsets. We claim that any two of these subsets U, W are connected in H_t, and thus will be merged. Again by Lemma 2, we know at least 5/6 fraction of points in $U \setminus B$ or $W \setminus B$ are great points, and thus there exist great points $x^* \in U, y^* \in W$ such that $S_t(x^*, y^*)$ is no more than $\mathrm{median}_{x \in U, y \in W} S_t(x, y)$. Notice that all great points in U are connected to great points in W in F_t, since they share at least $t - 2(\alpha + \nu)n$ neighbors. Then $S_t(x^*, y^*) \geq 3(|U| + |W|)/4 > (|U| + |W|)/4$, and thus $\mathrm{median}_{x \in U, y \in W} S_t(x, y) > (|U| + |W|)/4$. Therefore, any two subsets containing good points from $C \setminus B$ are connected in H_t and thus are merged.

So the two claims hold for all t, specially for $t = n$. Then the algorithm must stop after this threshold, and we have the lemma as desired. □

Lemma 4. *Algorithm 1 has a running time of* $O(n^{\omega+1})$.

Proof. To implement the algorithm, we introduce some data structures. For any $x \in V$, if y is within the t nearest neighbors of x, let $I_t(x, y) = 1$, otherwise

$I_t(x, y) = 0$. Initializing I_t takes $O(n^2)$ time. Next we compute $CN_t(x, y)$, the number of common neighbors between x and y. Notice that $CN_t(x, y) = \sum_{z \in V} I_t(x, z) I_t(y, z)$, so $CN_t = I_t I_t^T$. Then we can compute the adjacent matrix F_t (overloading notation for the graph F_t) from CN_t. These take $O(n^{\omega})$ time.
To compute the graph H_t, we introduce the following data structures. Let $FS_t(x, y) = 1$ if x, y are singleton subsets and $F_t(x, y) = 1$, and let $FS_t(x, y) = 0$ otherwise. Let $NS_t = FS_t(FS_t)^T$, then for two singleton subsets x, y, $NS_t(x, y)$ is the number of singleton subsets they share as neighbors in common in F_t. Similarly, let $FC_t(x, y) = 1$ if x and y are in the same subset and $F_t(x, y) = 1$, and let $FC_t(x, y) = 0$ otherwise. Let $S_t(x, y) = NS_t(FC_t)^T + FC_t(NS_t)^T$, then for two points $x \in U, y \in W$ where U, W are two non-singleton subsets, $S_t(x, y)$ is the number of points in $U \cup W$ they share as neighbors in common in F_t. Based on NS_t and S_t we can build the graph H_t. All these take $O(n^{\omega})$ time.

When we perform merge or increase the threshold, we need to update the data structures, which takes $O(n^{\omega})$ time. Since there are $O(n)$ merges and $O(n)$ thresholds, Algorithm 1 takes time $O(n^{\omega+1})$ in total. □

4 Experiments

In this section, we present our experimental results on evaluating our model and algorithm. While our main concern is building theoretical model for communities, empirical study is valuable in verifying the model and providing guidance for further improvement. Therefore, we applied our algorithm on both real world and synthetic data sets.

Note that the networks are represented as graphs, and we need to lift the graphs to get neighborhood functions for our algorithm. We use two lifting approaches for our experiments. The first approach is direct lifting: first, for any x, y set the affinity between x and y to be 1 if $(x, y) \in E$ and 0 otherwise; then for each x, sort all the other points according to the affinities; break ties randomly to avoid bias. The second approach is diffusion lifting: first set the affinity matrix K between entities to be $K = \exp\{\lambda A\}$ where $\lambda = 0.05$ and A is the adjacent matrix of the graph; then for each x, sort all the other points according to the affinities.

For comparison, we implemented two other algorithms: the lazy random walk algorithm (LRW [34]) and the Girvan-Newman algorithm (GN [10]). The lazy random walk algorithm performs truncated random walk from a seed point in the network and outputs selected communities where the selection is guided by the walk distribution and conductance. The conductance has been widely used as a criterion for quantifying the tight connections within communities, and thus the comparison to the lazy random walk algorithm provides an evaluation on how well our model and algorithm capture this intuition. The GN algorithm repeatedly removes the edge with the maximum edge-betweenness and regards the created connected components as communities. Although no theoretical model of hierarchical communities is targeted, the algorithm builds a hierarchy during its execution. It has been shown that the algorithm performs remarkably well on modeling communities in real-world data sets [10,29]. We use the code from [5] for fast computation of edge betweenness in the algorithm.

For algorithms with parameters, we run them multiple times with different values of parameters, and report the best result. More specifically, we run our algorithm using parameters $(\alpha + \nu) = \frac{i}{5n}(i = 1, 2, \ldots, 5)$. For the lazy random walk algorithm, we enumerate the parameters $\theta_0 = 0.05i(i = 1, \ldots, 4)$ and $b = 1, 2, \ldots, \lceil \log m \rceil$. In each run, 100 seed points are generated uniformly at random, each of which leads to a community. Since not all communities are meaningful (e.g. a singleton subset or the entire set of points), communities containing less than 10 points or containing more than $n - 10$ points are removed, and the rest communities are regarded as the output communities. We then evaluate the average error of the output communities. The error for a ground-truth community C with respect to a set $\mathcal{C}$ of output communities is defined as

$$\mathrm{error}(C, \mathcal{C}) = \min_{C' \in \mathcal{C}} \frac{|C \setminus C'| + |C' \setminus C|}{n}.$$

This criterion measures how well the ground-truth communities are recovered by the algorithm. We further note that our algorithm outputs fewer communities than the other algorithms in all the conducted experiments, and thus has advantage when they achieve similar performance.

4.1 Evaluation on Real-World Networks

To assess the performance of the proposed method in terms of accuracy, we conduct experiments on the following real world data sets[2] : karate [36], dolphins [23], polbooks [18], and football [10].

Figure 4 shows the average error and running time of the algorithms. We observe that our algorithm with diffusion lifting achieves the best performance on 3 out of 4 data sets, and achieves performance comparable to the GN algorithm on the football data set. It recovers the ground truth communities remarkably well over all the data sets. Our algorithm with direct lifting does not achieve good results. Note that this is due to the fact that diffusion lifting reflects the true neighborhood structure more accurately than direct lifting. More precisely, when we sort neighbors for a point p in direct lifting, all points not adjacent to p are ranked randomly. In fact some of them can be reached by a few steps and thus should be ranked as close neighbors, while others are actually far away from the point p. On the other hand, diffusion lifting leads to a neighborhood function that more accurately reflects the neighborhood information. The LRW algorithm has the worst performance, though it is the fastest. Our algorithm, especially with the diffusion lifting, runs 10-100 faster than the GN algorithm. Therefore, our algorithm with suitable neighborhood functions is the most favorable for detecting real world communities.

[2] Detailed descriptions and links for download can be found on `http://www-personal.umich.edu/~mejn/netdata/`

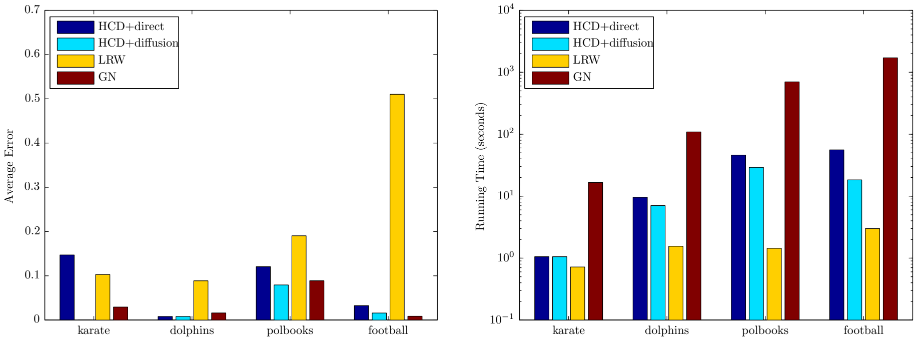


Fig. 4. The average error and running time using our hierarchical community detection algorithm with direct lifting (HCD+direct) or diffusion lifting neighborhood function (HCD+diffusion), Lazy Random Walk (LRW [34]) and the Girvan-Newman algorithm (GN [10]). Note that the running time is in log scale.

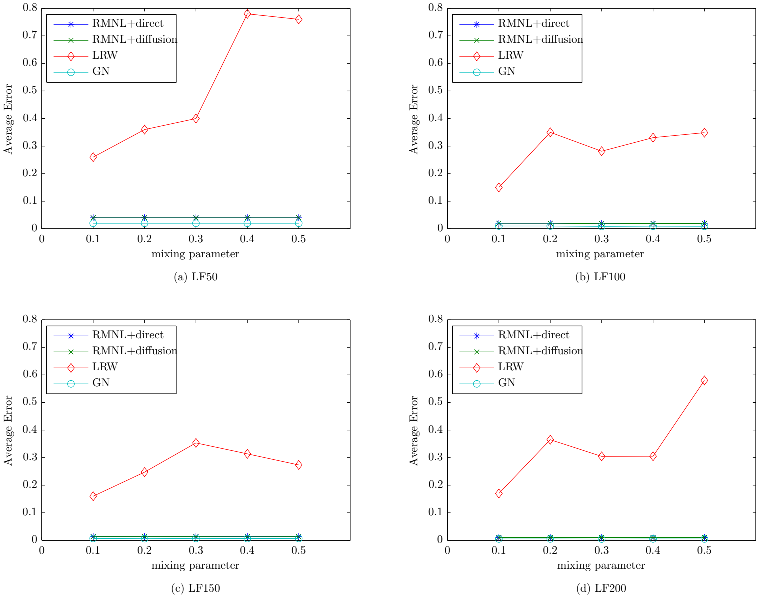
4.2 Evaluation on Synthetic Networks

Table 1. The parameters of the synthetic data sets for performance evaluation. n/m: number of nodes/edges; $k/maxk$: average/maximum degree of the nodes; $minc/maxc$: minimum/maximum size of the lower level communities; $minC/maxC$: minimum/maximum size of the higher level communities.

Data set	n	m	k	$maxk$	$minc$	$maxc$	$minC$	$maxC$
LF50	50	≈500	10	15	10	15	20	30
LF100	100	≈1500	15	20	15	20	30	40
LF150	150	≈3000	20	30	20	30	40	60
LF200	200	≈6000	30	40	30	40	60	80

Besides real-world networks, we further use the Lancichinetti-Fortunato (LF) benchmark[3] graphs [20] to evaluate the performance of the algorithms. By varying the parameters of the networks, we can analyze the behavior of the algorithms in detail. We generate four unweighted undirected benchmark networks with two level community hierarchies. The numbers of nodes are $50, 100, 150$ and 200 respectively, and some important parameters of the networks are given in Table 1. For each type of dataset, we range the mixing parameter μ from 0.1 to 0.5 with a span of 0.1, and set the low-level mixing parameter $\mu_1 = \mu/4$ and the high-level mixing parameter $\mu_2 = \mu - \mu_1$, resulting in five networks. Generally, the higher the mixing parameter of a network is, the more difficult it is to reveal the community structure.

[3] The source code we use and details about the parameters can be found on `https://sites.google.com/site/andrealancichinetti/software`



(a) LF50

(b) LF100

(c) LF150

(d) LF200

Fig. 5. The average error on the synthetic data sets

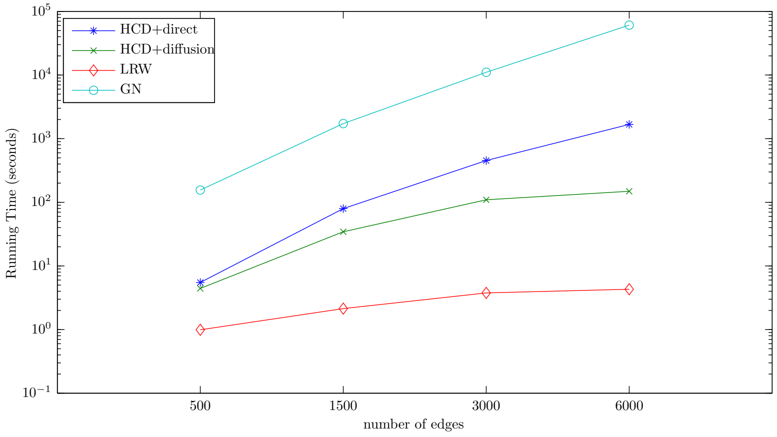


Fig. 6. The running time on the synthetic data sets

Figure 5 shows the average errors of the algorithms and Figure 6 shows the running time. Our algorithm with direct or diffusion lifting and the GN algorithm achieve similar results on all the benchmark networks. The errors of these algorithms are below 5%, and hardly increase with the mixing parameter. This suggests that they recover the ground truth communities remarkably well even in the hard case when the members of the communities have significant connections with the outside. In contrast, the LRW algorithm does not recover the communities well, even though it runs much faster than the other algorithms. Our algorithm runs about 50 times faster than the GN algorithm over all the data sets. These results are consistent with those observed on real world data sets, and again demonstrate the advantage of our algorithm.

5 Conclusion

In this paper we propose a model of communities that both reflects the tight connections within communities and explicitly models the hierarchy of communities. We present an efficient algorithm that provably detects all the communities in this model. Experiments demonstrate that our definition successfully models communities arising in the real world, and our algorithm compares favorably with existing approaches.

For future work, we plan to perform systematic empirical study of our model and algorithm using more neighborhood functions and on more real-world data sets. Another direction would be to speed up the computation of the neighborhood function and the algorithm and adapt them to large-scale scenarios.

Acknowledgements. This work was supported in part by NSF grant CCF-0953192, AFOSR grant FA9550-09-1-0538, and a Microsoft Research Faculty Fellowship.

References

1. Airoldi, E.M., Blei, D.M., Fienberg, S.E., Xing, E.P.: Mixed membership stochastic blockmodels. J. Mach. Learn. Res. 9, 1981–2014 (2008)
2. Arora, S., Ge, R., Sachdeva, S., Schoenebeck, G.: Finding overlapping communities in social networks: toward a rigorous approach. In: Proceedings of the 13th ACM Conference on Electronic Commerce (2012)
3. Balcan, M.F., Borgs, C., Braverman, M., Chayes, J., Teng, S.H.: Finding endogenously formed communities. In: SODA 2013 (2013)
4. Balcan, M.F., Gupta, P.: Robust hierarchical clustering. In: Proceedings of the Conference on Learning Theory, COLT (2010)
5. Bounova, G., de Weck, O.L.: Overview of metrics and their correlation patterns for multiple-metric topology analysis on heterogeneous graph ensembles. Phys. Rev. E 85(016117) (2012)
6. Clauset, A., Moore, C., Newman, M.E.J.: Hierarchical structure and the prediction of missing links in networks. Nature 453(7191), 98–101 (2008)
7. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Physical Review E 70(6), 066111+ (2004)
8. Fortunato, S.: Community detection in graphs. Physics Reports 486(3-5), 75–174 (2010)
9. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences 99(12), 7821–7826 (2002)

10. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences 99(12) (2002)
11. Gleiser, P., Danon, L.: Community Structure in Jazz. Advances in Complex Systems 6(4), 565–573 (2003)
12. He, J., Hopcroft, J., Liang, H., Suwajanakorn, S., Wang, L.: Detecting the structure of social networks using (α, β)-communities. In: Frieze, A., Horn, P., Prałat, P. (eds.) WAW 2011. LNCS, vol. 6732, pp. 26–37. Springer, Heidelberg (2011)
13. Ho, Q., Parikh, A.P., Xing, E.P.: A Multiscale Community Blockmodel for Network Exploration. Journal of the American Statistical Association 107(499), 916–934 (2012)
14. Huang, J., Sun, H., Han, J., Deng, H., Sun, Y., Liu, Y.: Shrink: a structural clustering algorithm for detecting hierarchical communities in networks. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 219–228. ACM (2010)
15. Kim, M., Leskovec, J.: Latent multi-group membership graph model. In: ICML (2012)
16. Kleinberg, J.: Complex networks and decentralized search algorithms. In: Proceedings of the International Congress of Mathematicians, ICM (2006)
17. Knuth, D.E.: The Stanford GraphBase: a platform for combinatorial computing. ACM, New York (1993)
18. Krebs, V.: `http://www.orgnet.com/` (unpublished)
19. Cosentino Lagomarsino, M., Jona, P., Bassetti, B., Isambert, H.: Hierarchy and feedback in the evolution of the Escherichia coli transcription network. Proceedings of the National Academy of Sciences 104(13), 5516–5520 (2007)
20. Lancichinetti, A., Fortunato, S.: Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Physical Review E 80(1), 016118 (2009)
21. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. New Journal of Physics 11(3), 033015 (2009)
22. Leskovec, J., Lang, K.J., Mahoney, M.: Empirical comparison of algorithms for network community detection. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 631–640. ACM, New York (2010)
23. Lusseau, D., Schneider, K., Boisseau, O.J., Haase, P., Slooten, E., Dawson, S.M.: Behavioral Ecology and Sociobiology 54, 396–405 (2003)
24. McAuley, J., Leskovec, J.: Learning to discover social circles in ego networks. In: Advances in Neural Information Processing Systems 25, pp. 548–556 (2012)
25. Mishra, N., Schreiber, R., Stanton, I., Tarjan, R.E.: Clustering social networks. In: Bonato, A., Chung, F.R.K. (eds.) WAW 2007. LNCS, vol. 4863, pp. 56–67. Springer, Heidelberg (2007)
26. Mishra, N., Schreiber, R., Stanton, I., Tarjan, R.E.: Finding strongly knit clusters in social networks. Internet Mathematics 5(1), 155–174 (2008)
27. Narasimhan, G., Smid, M.: Geometric spanning networks. Cambridge University Press (2007)
28. Newman, M.E.J.: Detecting community structure in networks. The European Physical Journal B - Condensed Matter and Complex Systems 38(2), 321–330 (2004)
29. Newman, M.E.J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences 103(23), 8577–8582 (2006)
30. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proceedings of the National Academy of Sciences of the United States of America 101(9), 2658–2663 (2004)

31. Ravasz, E., Somera, A.L., Mongru, D.A., Oltvai, Z.N., Barabási, A.L.: Hierarchical Organization of Modularity in Metabolic Networks. Science 297(5586), 1551–1555 (2002)
32. Schweinberger, M., Snijders, T.A.B.: Settings in social networks: A measurement model. Sociological Methodology 33, 307–341 (2003)
33. Shen, K., Song, L., Yang, X., Zhang, W.: A hierarchical diffusion algorithm for community detection in social networks. In: 2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 276–283. IEEE (2010)
34. Spielman, D.A., Teng, S.-H.: Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In: Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing, STOC 2004, pp. 81–90. ACM, New York (2004)
35. Yang, B., Di, J., Liu, J., Liu, D.: Hierarchical community detection with applications to real-world network analysis. Data & Knowledge Engineering (2012)
36. Zachary, W.W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452–473 (1977)

Graph Characterization Using Gaussian Wave Packet Signature

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. In this paper we present a new approach for characterizing graphs using the solution of the wave equation. The wave equation provides a richer and potentially more expressive means of characterizing graphs than the more widely studied heat equation. Unfortunately the wave equation whose solution gives the kernel is less easily solved than the corresponding heat equation. There are two reasons for this. First, the wave equation can not be expressed in terms of the familiar node-based Laplacian, and must instead be expressed in terms of the *edge-based Laplacian.* Second, the eigenfunctions of the edge-based Laplacian are more complex than that of the node-based Laplacian. In this paper we present a solution to the wave equation, where the initial condition is Gaussian wave packets on the edges of the graph. We propose a global signature of the graph which is based on the amplitudes of the waves at different edges of the graph over time. We apply the proposed method to both synthetic and real world datasets and show that it can be used to characterize graphs with higher accuracy.

Keywords: Edge-based Laplacian, Wave Equation, Gaussian wave packet, Graph Characterization.

1 Introduction

Graphs-based methods are frequently used to solve problems in many areas including computer vision machine learning and pattern recognition. This is due to the fact that most real world data can be conveniently represented by graphs or meshes. For example a color or a gray-scale image can be represented using a planar graph, where vertices are corners of the objects and edges represent some geometric relationship between the vertices. Similarly a chemical data structure can be represented using a graphs, where vertices represent atoms and edges represent bonds between the edges. A three-dimensional shapes can be conveniently represented using a mesh that approximates the bounding surface of the body. Once the graph of the object is extracted, we can use these graphs to find both the local and global properties of the object itself.

One of the most popular way of characterizing graph structure is to use spectral methods, which make use of the eigenvalues and eigenvectors of the Laplacian matrix. The Laplacian matrix is defined using the adjacency matrix of the

[⋆] Edwin Hancock was supported by a Royal Society Wolfson Research Merit Award

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 176–189, 2013.

graph and can be used to link equations from analysis to graph. Over the recent years many researchers have successfully used the solutions of partial differential equations defined using the Laplacian matrix to characterize graphs. For example, Xiao et al [1] have used heat kernel, which is derived from graph Laplacian, to embed the nodes of a graph in Euclidean space. Zhang et al[2] have used the heat kernel for anisotropic image smoothing. Sun et al[3] have used the heat kernel on mesh for defining signatures for 3D shapes and this is referred to as Heat Kernel Signature. Aubry et al[4] have used the solution of Schrödinger equation to define Wave Kernel Signature, which represents the average probability of measuring a quantum mechanical particle at a specific location. There are many other applications of graph Laplacian in the literature.

The discrete Laplacian defined over the vertices of a graph, however, cannot link most results in analysis to a graph theoretic analogue. For example the wave equation $u_{tt} = \Delta u$, defined with discrete Laplacian, does not have finite speed of propagation. In [5,6], Friedman and Tillich develop a calculus on graph which provides strong connection between graph theory and analysis. Their work is based on the fact that graph theory involves two different volume measures. i.e., a "vertex-based" measure and an "edge-based" measure. This approach has many advantages. It allows the application of many results from analysis directly to the graph domain.

While the method of Friedman and Tillich leads to the definition of both a divergence operator and a Laplacian (through the definition of both vertex and edge Laplacian), it is not exhaustive in the sense that the edge-based eigenfunctions are not fully specified. In a recent study we have fully explored the eigenfunctions of the edge-based Laplacian and developed a method for explicitly calculating the edge-interior eigenfunctions of the edge-based Laplacian [7]. This reveals a connection between the eigenfunctions of the edge-based Laplacian and both the classical random walk and the backtrackless random walk on a graph. The eigensystem of the edge-based Laplacian contains eigenfunctions which are related to both the adjacency matrix of the line graph and the adjacency matrix of the oriented line graph.

As an application of the edge-based Laplacian, we have recently presented a new approach to characterizing points on a non-rigid three-dimensional shape[8]. This is based on the eigenvalues and eigenfunctions of the edge-based Laplacian, constructed over a mesh that approximates the shape. This leads to a new shape descriptor signature, called the Edge-based Heat Kernel Signature (EHKS). The EHKS was defined using the heat equation, which is based on the edge-based Laplacian. This has applications in shape segmentation, correspondence matching and shape classification.

Wave equation provides potentially richer characterisation of graphs than heat equation. Initial work by Howaida and Hancock [9] has revealed some of its potential uses. They have proposed a new approach for embedding graphs on pseudo-Riemannian manifolds based on the wave kernel. However, there are two problems with the rigourous solution of the wave equation; a) we need to compute the edge-based Laplacian, and b) the solution is more complex than the heat equation. Recently we [10] have presented a solution of the edge-based wave equation on a graph. We assume that initial condition is a Gaussian wave packet on the edge of the graph, and show the evolution of this wave packet over time.

In this paper we propose a new signature for characterizing graphs, which is based on the solution of edge-based wave equation. The signature is constructed by assuming a Gaussian wave packet on a single edge of the graph and use the amplitude of the wave on different edges over different times to construct a unique signature for the graph.The remainder of this paper is organized as follows. We commence by introducing graphs and some definitions. In section 3, we introduce the eigensystem of the edge-based Laplacian. In section 4, we give a general solution of the wave equation, and the solution for the Gaussian wave packet as initial condition. In section 5, we define the proposed wave packet signature for the graph. Finally, in the experiment section, we apply the proposed method to both synthetic and real-world dataset.

2 Graphs

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 *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 *line graph* $L(G) = (\mathcal{V}_L, \mathcal{E}_L)$ 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. Therefore

$$\mathcal{V}_L = \{(u, v) \in D(G)\}$$

$$\mathcal{E}_L = \{((u, v), (v, w)) : (u, v) \in D(G), (v, w) \in D(G)\}$$

The *oriented line graph* $OL(G) = (\mathcal{V}_O; \mathcal{E}_O)$ is constructed in the same way as the $L(G)$ except that reverse pairs of arcs are not connected, i.e. $((u, v), (v, u))$ is not an edge. The vertex and edge sets of $OL(G)$ are therefore

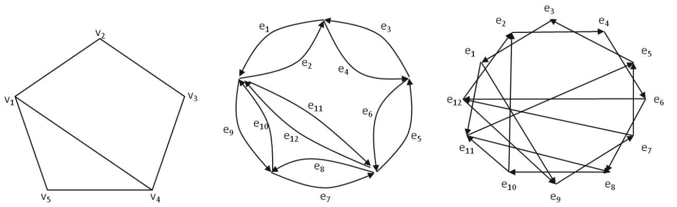
$$\mathcal{V}_O = \{(u, v) \in D(G)\}$$

$$\mathcal{E}_O = \{((u, v), (v, w)) : (v, w)), (u, v) \in D(G), (v, w) \in D(G), u \neq w\}$$

The *complement* or *inverse* of a graph G is a graph with the same vertex set but whose edge set consists of the edges not present in G. The complement is denoted by $\overline{G} = (\overline{\mathcal{V}}, \overline{\mathcal{E}})$, where

$$\overline{\mathcal{V}} = \mathcal{V}$$

$$\overline{\mathcal{E}} = \{(u, v) : (u, v) \notin \mathcal{E}\}$$



(a) Graph (b) Diagraph (c) Oriented line graph

Fig. 1. Graph, its digraph, and its oriented line graph

Figure 1(a) shows a simple graph, 1(b) its digraph, and 1(c) the corresponding oriented line graph. A random walk on the vertices of $L(G)$ represents the sequence of edges traversed in a random walk on the original graph G. Similarly, a random walk on the $OL(G)$ represents the sequence of edges traversed in a random walk on G where backtracking steps are not allowed (a backtrackless walk).

3 Edge-Based Eigensystem

In this section we review the eigenvalues and eigenfunction of the edge-based Laplacian[5][7]. 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 $l_e = 1$.

3.1 Vertex Supported Edge-Based Eigenfunctions

The vertex-supported eigenpairs of the edge-based Laplacian 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 edge-based Laplacian 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.

3.2 Edge-Interior Eigenfunctions

The edge-interior eigenfunctions are those eigenfunctions which are zero on vertices and therefore must have a principle frequency of $\omega \in \{\pi, 2\pi\}$. Recently we have shown that these eigenfunctions can be determined from the eigenvectors of the adjacency matrix of the oriented line graph[7]. We have shown that the eigenvector corresponding to eigenvalue $\lambda = 1$ of the oriented line graph provides a solution in the case $\omega = 2\pi$. In this case we obtain $|\mathcal{E}| - |\mathcal{V}| + 1$ linearly independent solutions. Similarly the eigenvector corresponding to eigenvalue $\lambda = -1$ of the oriented line graph provides a solution in the case $\omega = \pi$. In this case we obtain $|\mathcal{E}| - |\mathcal{V}|$ linearly independent solutions. This comprises all the principal eigenpairs which are only supported on the edges.

3.3 Normalization of Eigenfunctions

Note that although these eigenfunctions are orthogonal, they are not normalized. To normalize these eigenfunctions we need to find the normalization factor corresponding to each eigenvalue and divide each eigenfunction with the corresponding normalization factor. Let $\rho(\omega)$ denotes the normalization factor corresponding to eigenvalue ω. Then

$$\rho^2(\omega) = \sum_{e \in \mathcal{E}} \int_0^1 \phi^2(e, x_e)\, dx_e$$

Evaluating the integral, we get

$$\rho(\omega) = \sqrt{\sum_{e \in \mathcal{E}} C(e)^2 \left[\frac{1}{2} + \frac{\sin(2\omega + 2B(e))}{4\omega} - \frac{sin(2B(e))}{4\omega}\right]}$$

Once we have the normalization factor to hand, we can compute a complete set of orthonormal bases by dividing each eigenfunction with the corresponding normalization factor. Once normalized, these eigenfunctions form a complete set of orthonormal bases for $L^2(\mathcal{G}, \mathcal{E})$.

4 Solution of the Wave Equation

Let a graph coordinate $\mathcal{X}$ defines an edge e and a value of the standard coordinate on that edge x. The eigenfunctions of the edge-based Laplacian are

$$\phi_{\omega,n}(\mathcal{X}) = C(e,\omega)\cos\left(B(e,\omega) + \omega x + 2\pi n x\right)$$

The edge-based wave equation is

$$\frac{\partial^2 u}{\partial t^2}(\mathcal{X}, t) = \mathit{\Delta}_E u(\mathcal{X}, t) \tag{1}$$

We look for separable solutions of the form $u(\mathcal{X}, t) = \phi_{\omega,n}(\mathrm{X})g(t)$. This gives

$$\phi_{\omega,n}(\mathcal{X})g''(t) = g(t)\left(\omega + 2\pi n\right)^2 \phi(\omega, n)$$

which gives a solution for the time-based part as

$$g(t) = \alpha_{\omega,n}\cos\left[(\omega + 2\pi n)t\right] + \beta_{\omega,n} sin\left[(\omega + 2\pi n)t\right]$$

By superposition, we obtain the general solution

$$\begin{aligned} u\left(\mathcal{X}, t\right) = \sum_{\omega}\sum_{n} & C(e,\omega)\cos\left[B(e,\omega) + \omega x + 2\pi n x\right] \\ & \{\alpha_{\omega,n}\cos\left[(\omega + 2\pi n)t\right] + \beta_{\omega,n} sin\left[(\omega + 2\pi n)t\right]\} \end{aligned} \tag{2}$$

4.1 Initial Conditions

Since the wave equation is second order partial differential equation, we can impose initial conditions on both position and speed

$$u(\mathcal{X}, 0) = p(\mathcal{X})$$

$$\frac{\partial u}{\partial t}(\mathcal{X}, 0) = q(\mathcal{X})$$

and we obtain

$$p(\mathcal{X}) = \sum_{\omega}\sum_{n} \alpha_{\omega,n} C(e,\omega)\cos\left[B(e,\omega) + \omega x + 2\pi n x\right]$$

$$q(\mathcal{X}) = \sum_{\omega}\sum_{n} \beta_{\omega,n}(\omega + 2\pi n) C(e,\omega)\cos\left[B(e,\omega) + \omega x + 2\pi n x\right]$$

We can obtain these coefficients using the orthogonality of the eigenfunctions. So we get

$$\alpha_{\omega,n} = \sum_e C(e,\omega)\frac{1}{2}\left[F_{\omega,n} + F^*_{\omega,n}\right]$$

where

$$F_{\omega,n} = e^{iB}\int_0^1 dxp(e,x)e^{i\omega x}e^{i2\pi n}$$

similarly

$$\beta_{\omega,n}(\omega + 2\pi n) = \sum_e C(e,\omega)\frac{1}{2}\left[G_{\omega,n} + G^*_{\omega,n}\right]$$

where

$$G_{\omega,n} = e^{iB}\int_0^1 dxq(x,e)e^{i(\omega+2\pi n)x} = e^{iB}\int_0^1 dxp'(x,e)e^{i(\omega+2\pi n)x}$$

4.2 Gaussian Wave Packet

Let the initial position be a Gaussian wave packet $p(e,x) = e^{-a(x-\mu)^2}$ on one particular edge and zero everywhere else. Then we have

$$\begin{aligned}F_{\omega,n} &= e^{iB}\int_0^1 dxe^{-a(x-\mu)^2}e^{i\omega x}e^{i2\pi nx}\\ &= e^{iB}e^{i\mu\omega}e^{-\frac{\omega^2}{4a}}\int_0^1 dxe^{-a\left(x-\mu-\frac{i\omega}{2a}\right)^2}e^{i2\pi nx}\end{aligned}$$

Let the Gaussian is fully contained on one edge. i.e., $p(x,e)$ is only supported on this edge, then

$$F_{\omega,n} = e^{iB}e^{i\mu\omega}e^{-\frac{\omega^2}{4a}}\int_{-\infty}^{\infty} dxe^{-a\left(x-\mu-\frac{i\omega}{2a}\right)^2}e^{i2\pi nx}$$

Solving, we get

$$F_{\omega,n} = \sqrt{\frac{\pi}{a}}e^{i[B+\mu(\omega+2\pi n)]}e^{-\frac{1}{4a}(\omega+2n\pi)^2}$$

Similarly we obtain

$$F^*_{\omega,n} = \sqrt{\frac{\pi}{a}}e^{-i[B+\mu(\omega+2\pi n)]}e^{-\frac{1}{4a}(\omega+2n\pi)^2}$$

and so

$$\alpha_{\omega,n} = \sqrt{\frac{\pi}{a}}e^{-\frac{1}{4a}(\omega+2n\pi)^2}C(e,\omega)\cos[B+\mu\left(\omega+2\pi n\right)] \tag{3}$$

Since $p(x,e)$ is zero at both ends the coefficients β can be found straightforwardly.

$$\beta_{\omega,n} = \sqrt{\frac{\pi}{a}}e^{-\frac{1}{4a}(\omega+2n\pi)^2}C(e,\omega)\sin[B+\mu\left(\omega+2\pi n\right)] \tag{4}$$

4.3 Complete Reconstruction

Let f be the edge on which the initial function is non-zero. Let the Gaussian is fully contained on one edge. Then

$$u(\mathcal{X},t) = \sum_{\omega} \sqrt{\frac{\pi}{a}} C(\omega,e)C(\omega,f) \sum_{n} e^{-\frac{1}{4a}(\omega+2\pi n)^2}$$
$$\cos\left[B(\omega,e)+\omega x+2\pi n x\right]\cos\left[B(\omega,f)+(\omega+2\pi n)(t+\mu)\right]$$

For a particular sequence with principal eigenvalue ω, we need to calculate

$$u_\omega = \sum_{n} \sqrt{\frac{\pi}{a}} e^{-\frac{1}{4a}(\omega+2\pi n)^2} \cos\left[B(\omega,e)+\omega x+2\pi n x\right]\cos\left[B(\omega,f)+(\omega+2\pi n)(t+\mu)\right]$$

Writing the cosine in exponential form, we obtain

$$\begin{aligned} u_w = &\sum_{n} \sqrt{\frac{\pi}{a}} e^{-\frac{1}{4a}(\omega+2\pi n)^2} \\ &\times \frac{1}{4}\left[e^{i[B(e,\omega)+B(f,\omega)]}e^{i(\omega+2\pi n)(x+t+\mu)} + e^{-i[B(e,\omega)+B(e,\omega)]}e^{-i(\omega+2\pi n)(x+t+\mu)}\right. \\ &\left.+e^{i[B(e,\omega)-B(f,\omega)]}e^{i(\omega+2\pi n)(x-t-\mu)} + e^{-i[B(e,\omega)-B(e,\omega)]}e^{-i(\omega+2\pi n)(x-t-\mu)}\right] \end{aligned}$$

We need to evaluate terms like terms like$\sum_n \frac{\pi}{a} e^{-\frac{1}{4a}} e^{i[B(e,\omega)+B(f,\omega)]} e^{i(\omega+2\pi n)(x+t+\mu)}$, where the values of ω and n depend on the particular eigenfunction sequence under evaluation.

Let $\mathcal{W}(z)$ be z wrapped to the range $[-\frac{1}{2},\frac{1}{2})$, i.e.,

$$\mathcal{W}(z) = z - \left\lfloor z+\frac{1}{2}\right\rfloor$$

Solving for all cases, the complete solution becomes

$$\begin{aligned} u(\mathcal{X},t) = &\sum_{\omega\in\Omega_a} \frac{C(\omega,e)C(\omega,f)}{2}\left(e^{-a\mathcal{W}(x+t+\mu)^2}\cos\left[B(e,\omega)+B(f,\omega)+\omega\left\lfloor x+t+\mu+\frac{1}{2}\right\rfloor\right]\right. \\ &\left.+\,e^{-a\mathcal{W}(x-t-\mu)^2}\cos\left[B(e,\omega)-B(f,\omega)+\omega\left\lfloor x-t-\mu+\frac{1}{2}\right\rfloor\right]\right) \\ &+\frac{1}{2|E|}\left(\frac{1}{4}e^{-a\mathcal{W}(x+t+\mu)^2}+\frac{1}{4}e^{-a\mathcal{W}(x-t-\mu)^2}\right) \\ &+\sum_{\omega\in\Omega_c}\frac{C(\omega,e)C(\omega,f)}{4}\left(e^{-a\mathcal{W}(x-t-\mu)^2}-e^{-a\mathcal{W}(x+t+\mu)^2}\right) \\ &+\sum_{\omega\in\Omega_c}\frac{C(\omega,e)C(\omega,f)}{4}\left((-1)^{\left\lfloor x-t-\mu+\frac{1}{2}\right\rfloor}e^{-a\mathcal{W}(x-t-\mu)^2}\right. \\ &\left.-(-1)^{\left\lfloor x+t+\mu+\frac{1}{2}\right\rfloor}e^{-a\mathcal{W}(x+t+\mu)^2}\right) \end{aligned} \tag{5}$$

where Ω_a represents the set of vertex-supported eigenvalues and Ω_b and Ω_c represent the set of edge-interior eigenvalues respectively. i.e., π and 2π.

5 Wave Packet Signatures

Once a complete solution of the edge-based wave equation is known, we can use it to define both local and global signatures for graphs and meshes. In this paper we define a global signature for characterizing graphs which is based on amplitudes of waves on the edges of the graph over time. To define the signature we assume that the initial condition is a Gaussian wave packet on a single edge of the graph. For this purpose we select the edge $(u, v) \in E$, such that u is the highest degree vertex in the graph and v is the highest degree vertex in the neighbours of u. We define the local signature of an edge as

$$WPS(\mathcal{X}) = [u(\mathcal{X}, t_0), u(\mathcal{X}, t_1), u(\mathcal{X}, t_2), ...u(\mathcal{X}, t_n)] \tag{6}$$

Given a graph G, we define its global wave packet signature as

$$GWPS(G) = hist\left(WPS(\mathcal{X}_1), WPS(\mathcal{X}_2), , ..., WPS(\mathcal{X}_{|E|})\right) \tag{7}$$

where hist(.) is the histogram operator which bins the list of arguments $WPS(\mathcal{X}_1)$, $WPS(\mathcal{X}_2), , ..., WPS(\mathcal{X}_{|E|})$.

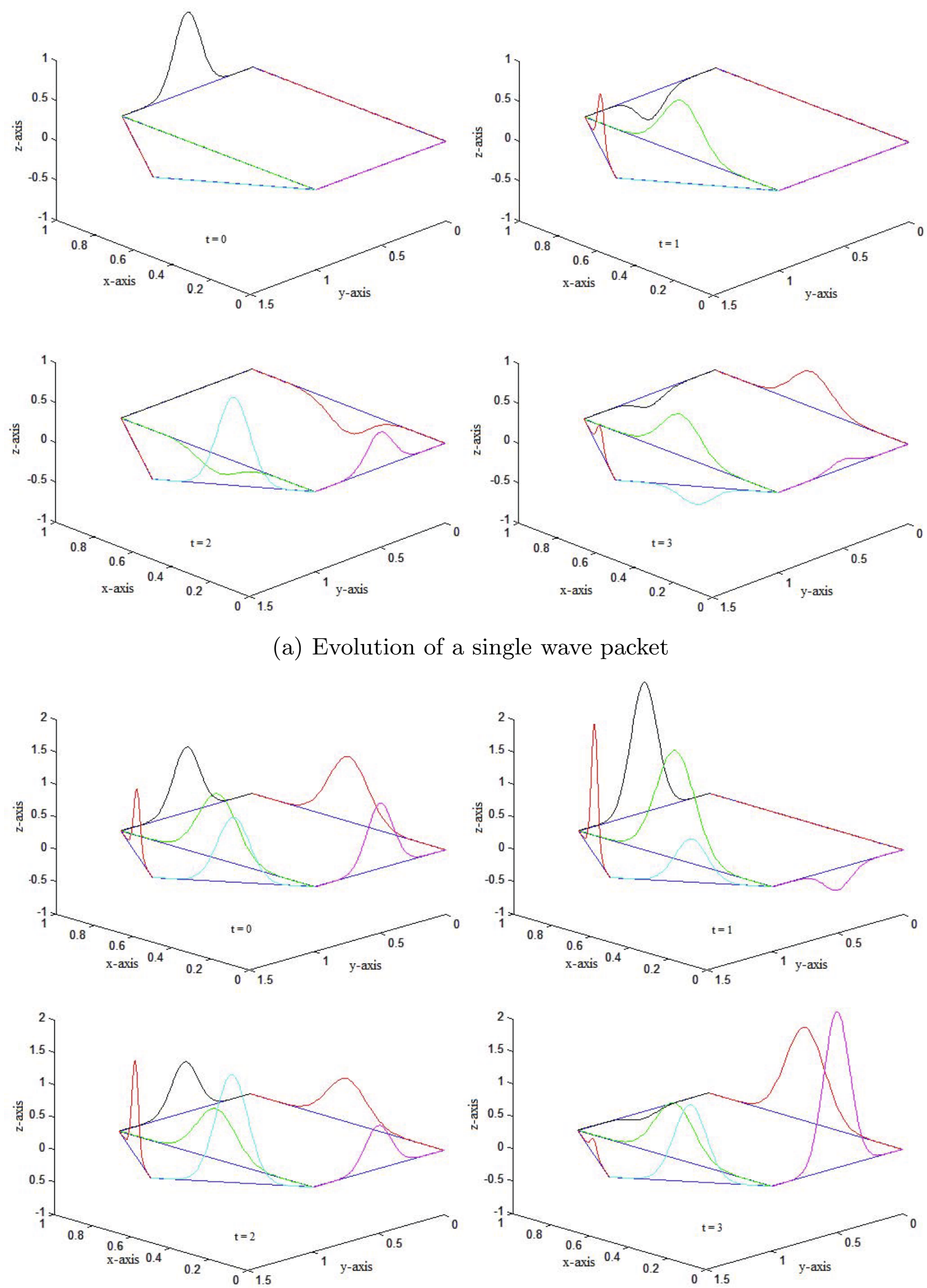
6 Experiments

In this section we apply our proposed method on both synthetic and real world datasets.

6.1 Synthetic Dataset

To show the evolution of Gaussian wave packet on a graph, we take a simple graph with 5 nodes and 6 edges. We assume the initial condition as a Gaussian wave packet on a single edge and zero everywhere else. Figure 2(a) shows the results for the times $t = 0$, $t = 1$, $t = 2$ and $t = 3$ in a three dimensional space. Note that when the wave packet hits a node with degree greater than 2, some part of the packet is reflected back while the other part is equally distributed to the connecting edges. Figure 2(b) shows a similar analysis but with a different initial condition. Here we assume that initially a Gaussian wave packet exist on every edge of the graph and show its evolution for the times $t = 0$, $t = 1$, $t = 2$ and $t = 3$.

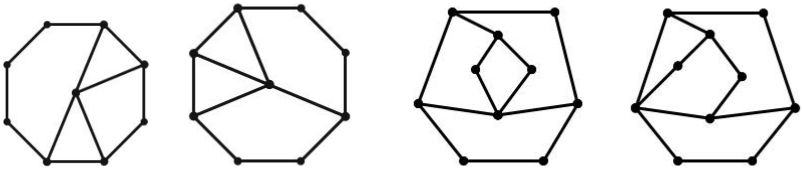
One of the advantage of using the solution of equations defined using edge-based Laplacian is that it is less prone to the problem of failing to distinguish graphs due to cospectrality of the Laplacian or adjacency matrices. This is due to the fact that the structure of edge-interior eigenfunctions of the edge-based Laplacian are determined by the eigenvectors of the oriented line graph which is closely related to discrete time quantum walk on a graph [11]. Figure 3(a) and Figure 3(b) show two pairs of graphs with 9 and 10 vertices respectively, which are cospectral with respect to both their adjacency matrices and the adjacency matrices of their complements. Figure 4(a) and Figure 4(b) show the global wave



(a) Evolution of a single wave packet

(b) Evolution of multiple wave packets

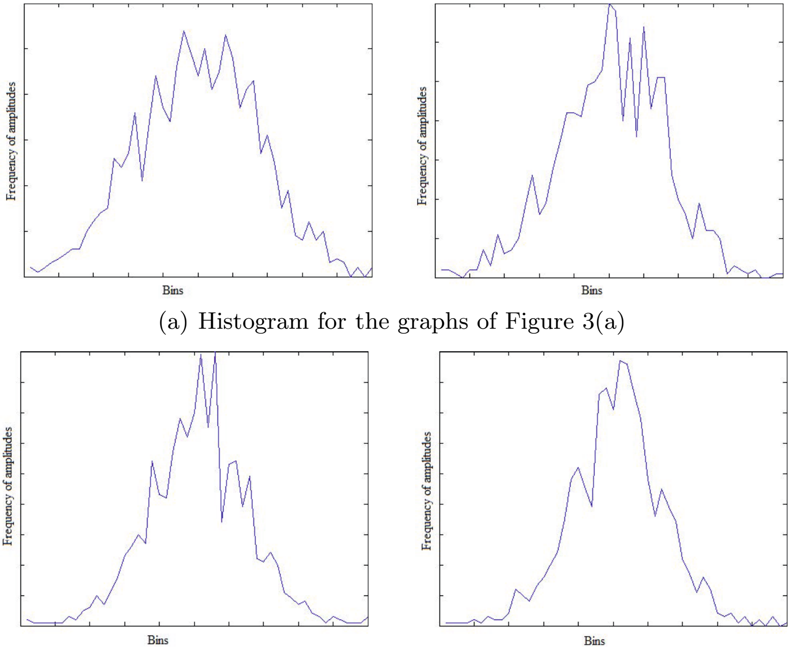
Fig. 2. Solution of wave equation on a graph with 6 vertices and 8 edges



(a) cospectral graphs with 9 nodes

(b) cospectral graphs with 10 nodes

Fig. 3. Examples of cospectral graphs



(a) Histogram for the graphs of Figure 3(a)

(b) Histogram for the graphs of Figure 3(b)

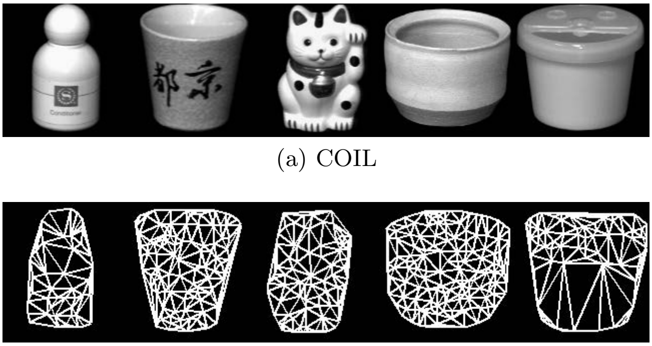
Fig. 4. Histograms for cospectral graphs

packet signature for the graphs of Figure 3(a) and Figure 3(b). Results show the ability of the wave equation to distinguish cospectral graphs. This is due to the fact that although these graphs cannot be distinguished by random walks on the graph, backtrackless walks on the other hand can distinguish such graphs[12].

6.2 Real-World Dataset

Finally, we apply the proposed method on real world dataset. Our dataset consists of graphs extracted from the images in the Columbia object image library

(COIL) dataset [13]. This dataset contains views of 3D objects under controlled viewer and lighting condition. For each object in the database there are 72 equally spaced views. The objective here is to cluster different views of the same object onto the same class. To establish a graph on the images of objects, we first extract feature points from the image. For this purpose, we use the Harris corner detector [14]. We then construct a Delaunay graph using the selected feature points as vertices. Figure 5(a) shows some of the object views (images) used for our experiments and Figure 5(b) shows the corresponding Delaunay triangulations.



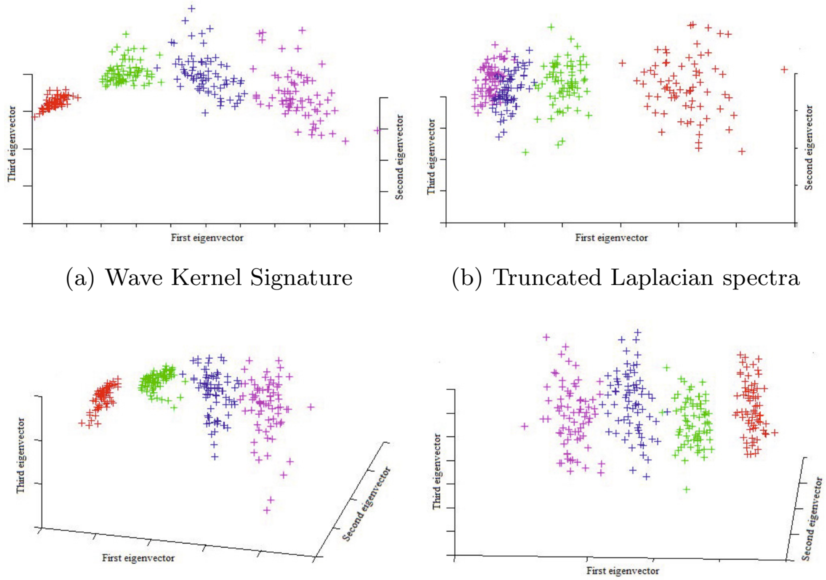
(a) COIL

(b) Delaunay triangulation

Fig. 5. COIL objects and their Delaunay triangulations

We compute the wave signature for an edge by taking $t_{min} = 10$, $t_{max} = 100$ and $x_e = 0.5$. We then compute the GWPS for the graph by fixing 100 bins for histogram. To visualize the results, we have performed principal component analysis (PCA) on GWPS. PCA is mathematically defined [15] as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. Figure 6(a) shows the results of the embedding of the feature vectors on the first three principal components.

To measure the performance of the proposed method we compare it with truncated Laplacian, random walk [16] and Ihara coefficients [17]. Figure 6 shows the embedding results for different methods. To compare the performance, we cluster the feature vectors using *k-means clustering* [18]. *k*-means clustering is a method which aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean. We compute *Rand index* [19] of these clusters which is a measure of the similarity between two data clusters. The rand indices for these methods are shown in Table 1. It is clear from the table that the proposed method can classify the graphs with higher accuracy.



(a) Wave Kernel Signature (b) Truncated Laplacian spectra

(c) Random Walk Kernel (d) Ihara Coefficients

Fig. 6. Graph, its digraph, and its oriented line graph

Table 1. Experimental results on Mutag dataset

Method	Accuracy
Wave Kernel Signature	0.9965
Random Walk Kernel	0.9526
Truncated Laplacian Spectra	0.8987
Ihara Coefficients	0.9864

7 Conclusion and Future Work

In this paper we have used the solution of the wave equation on a graph to characterize graphs. The wave equation is solved using the edge-based Laplacian of a graph. We assume the initial distribution be a Gaussian wave packet and shown its evolution with time on different graphs. We use the amplitudes of the wave over different edges to define a signature for graph characterization. The advantage of using the edge-based Laplacian over vertex-based Laplacian is that it allows the direct application of many results from analysis to graph theoretic domain. For example it allows the study of non-dispersive solutions or solitons. In future our goal is to use the solution of other equations defined using the edge-based Laplacian for defining local and global signatures for graphs.

References

1. Xiao, B., Yu, H., Hancock, E.R.: Graph matching using manifold embedding. In: Campilho, A., Kamel, M. (eds.) ICIAR 2004. LNCS, vol. 3211, pp. 352–359. Springer, Heidelberg (2004)
2. Zhang, F., Hancock, E.R.: Graph spectral image smoothing using the heat kernel. Pattern Recognition, 3328–3342 (2008)
3. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. Comp. Graph Forum, 1383–1392 (2010)
4. Aubry, M., Schlickewei, U., Cremers, D.: The wave kernel signature: A quantum mechanical approach to shape analysis. Tech. rep., TU München, Germany (2011)
5. Friedman, J., Tillich, J.P.: Wave equations for graphs and the edge based laplacian. Pacific Journal of Mathematics, 229–266 (2004)
6. Friedman, J., Tillich, J.P.: Calculus on graphs. CoRR (2004)
7. Wilson, R.C., Aziz, F., Hancock, E.R.: Eigenfunctions of the edge-based laplacian on a graph. Linear Algebra and its Applications 438, 4183–4189 (2013)
8. Aziz, F., Wilson, R.C., Hancock, E.R.: Shape analysis using the edge-based laplacian. 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. 382–390. Springer, Heidelberg (2012)
9. ElGhawalby, H., Hancock, E.R.: Graph embedding using an edge-based wave kernel. In: Hancock, E.R., Wilson, R.C., Windeatt, T., Ulusoy, I., Escolano, F. (eds.) SSPR&SPR 2010. LNCS, vol. 6218, pp. 60–69. Springer, Heidelberg (2010)
10. Aziz, F., Wilson, R.C., Hancock, E.R.: Gaussian wave packet on a graph. Graph-based Representation (2013)
11. Emms, D., Severini, S., Wilson, R.C., Hancock, E.R.: Coined quantum walks lift the cospectrality of graphs. Pattern Recognition (2009)
12. Aziz, F., Wilson, R.C., Hancock, E.R.: Backtrackless walks on a graph. IEEE Transaction on Neural Networks and Learning Systems 24, 977–989 (2013)
13. Murase, H., Nayar, S.K.: Visual learning and recognition of 3-d objects from appearance. International Journal of Computer Vision 14, 5–24 (1995)
14. Harris, C., Stephens, M.: A combined corner and edge detector. In: Fourth Alvey Vision Conference, Manchester, UK, pp. 147–151 (1988)
15. Jolliffe, I.T.: Principal component analysis. Springer, New York (1986)
16. Gärtner, T., Flach, P., 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)
17. Ren, P., Wilson, R.C., Hancock, E.R.: Graph characterization via Ihara coefficients. IEEE Tran. on Neural Networks 22, 233–245 (2011)
18. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations, vol. 1, pp. 281–297. University of California Press (1967)
19. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66, 846–850 (1971)

Analysis of the Schrödinger Operator in the Context of Graph Characterization

Pablo Suau[1], Edwin R. Hancock[2], and Francisco Escolano[1]

[1] Mobile Vision Research Lab, University of Alicante, Spain
{pablo,sco}@dccia.ua.es

[2] Department of Computer Science, University of York, UK
edwin.hancock@york.ac.uk

Abstract. In this paper, we apply the solution of the Schrödinger equation, i.e. the Schrödinger operator, to the graph characterization problem. The motivation behind this approach is two-fold. Firstly, the mathematically similar heat kernel has been used in the past for this same problem. And secondly, due to the quantum nature of the Schrödinger equation, our hypothesis is that it may be capable of providing richer sources of information. The two main features of the Schrödinger operator that we exploit in this paper are its non-ergodicity and the presence of quantum interferences due to the existence of complex amplitudes with both positive and negative components. Our proposed graph characterization approach is based on the Fourier analysis of the quantum equivalent of the heat flow trace, thus relating frequency to structure. Our experiments, performed both on synthetic and real-world data, demonstrate that this new method can be successfully applied to the characterization of different types of graph structures.

Keywords: graph characterization, heat flow, Schrödinger equation, quantum walks.

1 Introduction

Many physical, biological or social systems may be represented by means of a network or a graph. The analysis of graph structure and features thus becomes significant as a way of understanding the structure and dynamics of these systems. This fact hence motivated the appearance of several graph characterization techniques reported in the literature. The aim of graph characterization is to provide a way to distinguish and compare different types of graph structures without applying subgraph isomorphism, a procedure that is NP-complete. Among these graph characterization algorithms several are based on random walks [1], the Ihara zeta function [2] or the spectral radius [3]. In a recent paper, Escolano *et al.* [4] introduced an alternative technique based on the analysis of the heat flow. The heat flow is derived from the heat kernel [5], which is the solution of the heat diffusion equation, and provides a method to represent the heat transfer between nodes of a graph over time.

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 190–203, 2013.

The Schrödinger equation is mathematically similar in structure to the heat diffusion equation [6]. However, they describe rather different physical phenomena. While the heat equation describes how heat is transfered in a system, the Schrödinger equation characterizes the dynamics of a particle in a quantum system. The quantum nature of the Schrödinger equation and its complex-valued solutions give rise to many interesting non-classical effects, including quantum interferences. These interferences have proved to be useful in several applications, including the detection of symmetric motifs in graphs via continuous-time quantum walks [7] and graph embedding by means of quantum commute times [8]. Motivated by previous works on graph characterization from the solution of the heat diffusion equation, in this paper we demonstrate that the solution of the Schrödinger equation, i.e. the Schrödinger operator, may also be useful for this task. We exploit the non-ergodicity of dynamic quantum systems based on the Schrödinger equation and propose a new frequency domain characterization, based on the Fourier analysis of the quantum equivalent to the heat flow trace [4]. The resulting characterization relates frequency and graph structure. Our experiments both on synthetic and real-world datasets demonstrate that such an approach successfully distinguishes different types of network structures.

The remainder of this paper is structured as follows. In Section 2 we summarize the concept of heat flow for graph characterization. In Section 3 the Schrödinger operator is introduced. The main contributions of this paper are presented in Section 4, in which we formally analyze the Schrödinger operator and propose a new graph characterization technique based on an equivalent of the heat flow. Then, in Section 5, we show some experimental results. Finally we draw some conclusions and point out ways in which this work can be further extended.

2 Heat Flow

Let $G = (V, E)$ be an undirected graph where V is its set of nodes and $E \subseteq V \times V$ is its set of edges. The Laplacian matrix $L = D - A$ is constructed from the $|V| \times |V|$ adjacency matrix A, in which the element $A(u, v) = 1$ if $(u, v) \in E$ and 0 otherwise, where the elements of the diagonal $|V| \times |V|$ degree matrix are $D(u, u) = \sum_{v \in V} A(u, v)$. The $|V| \times |V|$ heat kernel matrix K_t is the fundamental solution of the heat equation

$$\frac{\partial K_t}{\partial t} = -LK_t, \tag{1}$$

and depends on the Laplacian matrix L and time t. It describes how information flows across the edges of a graph with time, and its solution is $K_t = e^{-Lt}$.

The heat kernel K_t is a doubly stochastic matrix. Double stochasticity implies that diffusion conserves heat. In [4], a graph is characterized from the constraints it imposes to heat diffusion due to its structure. This characterization is based on the normalized instantaneous flow $F_t(G)$ of graph G, that accounts the edge-normalized heat flowing through the graph at a given instant t, and it is defined as:

$$F_t(G) = \frac{2|E|}{n} \sum_{i=1}^{n} \sum_{j \neq i} A(i,j) \left(\sum_{k=1}^{n} \phi_k(i)\phi_k(j)e^{-\lambda_k t} \right). \tag{2}$$

A more compact definition of the edge-normalized instantaneous flow is $F_t(G) = (2|E|/n)A : K_t$, where $X : Z = trace(XZ^T)$ is the Frobenius inner product. The heat flow trace describing the graph is constructed by computing Eq. 2 on the interval $[0, t_{max}]$.

3 Heat Kernel vs. Schrödinger Operator

The Schrödinger equation describes how the complex state vector $|\psi_t\rangle \in \mathbb{C}^{|V|}$ of a continuous-time quantum walk varies with time [9]:

$$\frac{\partial|\psi_t\rangle}{\partial t} = -iL|\psi_t\rangle. \tag{3}$$

Given an initial state $|\psi_0\rangle$ the latter equation can be solved to give $|\psi_t\rangle = \Psi_t|\psi_0\rangle$, where $\Psi_t = e^{-iLt}$ is a complex $|V| \times |V|$ unitary matrix. In this paper we refer to Ψ_t as the *Schrödinger operator*. Our attention in this paper will be focused on the operator itself and not on the quantum walk process. As can be seen, Eq. 3 is similar to Eq. 1. However, the physical dynamics induced by the Schrödinger equation are totally different, due to the existence of oscillations and interferences.

In this section we address the question of whether the Schrödinger operator may be used to characterize the structure of a graph. Empirical analysis on different graph structures shows that both the heat kernel and the Schrödinger operator evolve with time in a manner which strongly depends on graph structure. [1] However, the underlying physics and the dynamics are different (see Fig. 1). In the case of heat flow heat diffuses between nodes through the edges, eventually creating transitive links (energy exchanges between nodes that are not directly connected by an edge), until reaching a stationary energy equilibrium state. The Schrödinger operator yields a faster energy distribution through the system (e.g. for a 100 nodes line graph, it takes $t = 50$ time steps for the Schrödinger operator to reach every possible position on the graph, taking more than twice this time in the case of the heat kernel [4]). Moreover, due to negative components of the complex amplitudes, interferences are created, producing energy waves. The main difference is that the Schrödinger operator never reaches an equilibrium state. In other words, it is non-ergodic. Graph connectivity imposes constraints on the distribution of energy. In the case of the heat kernel, a higher number of energy distribution constraints implies the creation of more transitive links with time [4]. This is also true in the case of the Schrödinger operator, for which lower frequency and more symmetrical energy distribution patterns are also observed.

[1] Videos showing the evolution of both heat kernel and Schrödinger operator are available at `http://www.dccia.ua.es/~pablo/downloads/schrodinger_operator.zip`

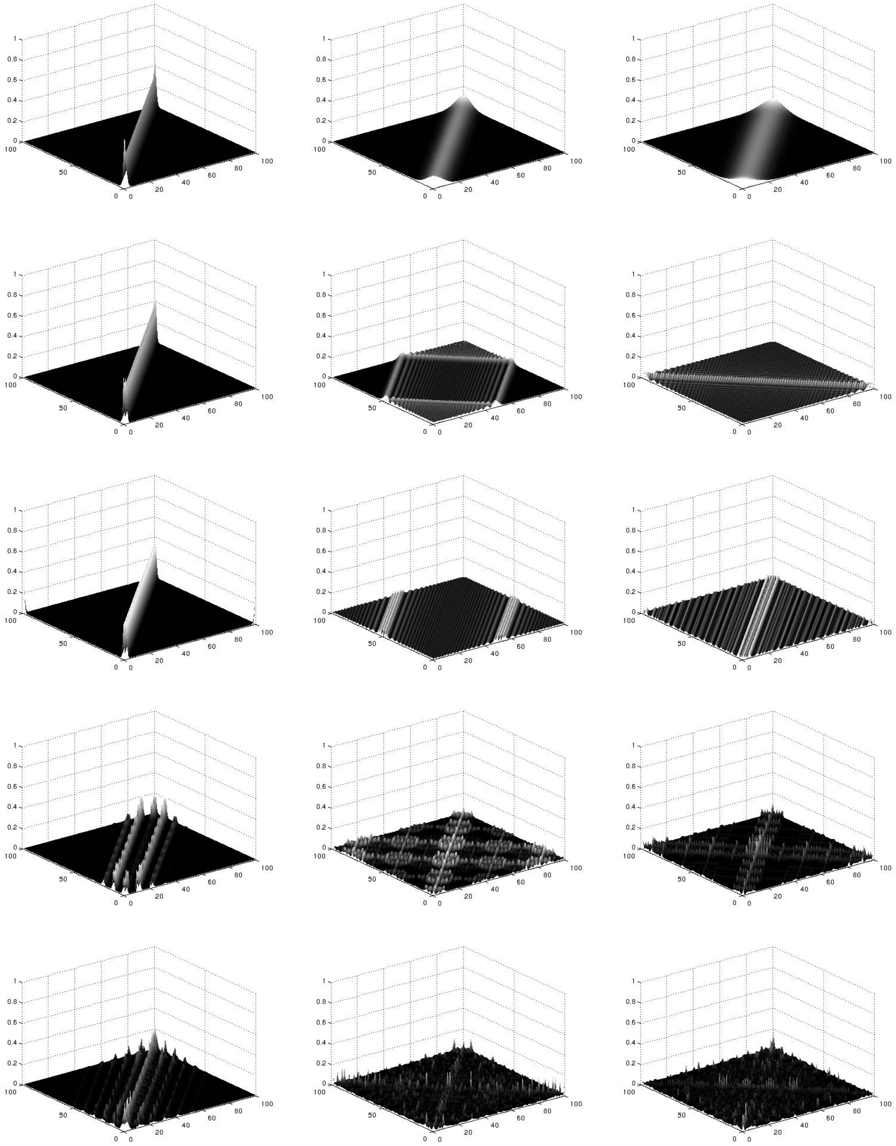


Fig. 1. Evolution with time ($t = 1, 25$ and 100). From top to bottom: heat kernel for a 100 node line graph, Schrödinger operator for a 100 node line graph, Schrödinger operator for a 100 node circle graph, Schrödinger operator for a 10×10 grid graph with 4 neighbour connectivity and Schrödinger operator for a 10×10 grid graph with 8 neighbour connectivity.

3.1 Analysis of the Schrödinger Operator

Further formal analysis of the Schrödinger operator supports the empirical evidence stated above. We first consider the Schrödinger operator when t tends to zero. Its Taylor expansion is given by:

$$\Psi t = e^{-iLt} = \cos Lt - i \sin Lt = I_{|V|} - iLt - \frac{t^2}{2!}L^2 + i\frac{t^3}{3!}L^3 + \frac{t^4}{4!}L^4 \cdots, \quad (4)$$

where $I_{|V|}$ is the $|V| \times |V|$ identity matrix. Hence

$$\lim_{t \to 0} \Psi_t \approx I_{|V|} - iLt, \quad (5)$$

where $\Psi_t = K_t$ when $t = 0$. At this time instant every node conserves its energy (as in the case of the Heat Kernel). The role of the identity matrix is to make the Schrödinger operator unitary. Due to the $-iLt$ term, it can be seen that energy spreads as a wave even for t values close to zero. Thus, the Schrödinger operator causes energy to distribute in a waveform from the initial time instant.

In order to explore the ergodicity of the Schrödinger operator we consider both its spectral decomposition and that of the heat kernel:

$$K_t = \sum_{p=1}^{n} e^{-t\lambda_p} \phi_p \phi_p^T \text{ and} \quad (6)$$

$$\Psi_t = \sum_{p=1}^{n} e^{-it\lambda_p} \phi_p \phi_p^T, \quad (7)$$

where λ_p is the p-th eigenvalue of the Laplacian L and ϕ_p its corresponding eigenvector.

The spectral decomposition of the heat kernel demonstrates that it is dominated by the lowest eigenvalues, due to the fact that $e^{-t\lambda_p}$ tends to zero as t tends to infinity. However, the limit of $e^{-it\lambda_p}$ when t tends to minus infinity is infinite. Thus, there are two important differences with the heat kernel. Firstly, the Schrödinger operator never converges (it is non-ergodic), and secondly, it is not dominated by any particular eigenvalue (i.e. there is more dependence on global graph structure as t tends to infinity).

Finally, we can compare the Euler equation based Schrödinger operator Ψ_t with the wave equation formula

$$\psi = \upsilon e^{i(kx - wt + \epsilon)}, \quad (8)$$

where υ is the amplitude, ϵ is the initial phase, k is the wavenumber, and w is the angular frequency. The Schrödinger operator can be interpreted as a wave with $\upsilon = 1$, $k = \epsilon = 0$ and $w = L$. In fact, Eq. 7 expresses the Schrödinger operator as a linear combination of $p = 1 \ldots n$ waves with different frequencies λ_p.

3.2 The Quantum Energy Flow

As stated in Section 2 the heat flow characterizes a graph by means of a trace that accounts for the information flowing on the graph with time. Due to the similarity between the heat diffusion and the Schrödinger equations, we could define the quantum energy flow (QEF) as

$$Q_t(G) = A : \Psi_t, \tag{9}$$

and the quantum energy trace (the equivalent of heat flow) as the evolution of Q_t with time. It must be noted that the Hamiltonian of the quantum system defined by Ψ_t is given by the graph Laplacian L. The adjacency matrix A in Eq. 9 causes the QEF to only account for the energy distributing through edges. In Fig. 2 we compare the heat flow and the QEF traces for two different types of graphs. In [4], graph structure is characterized by the heat flow's phase transition point (PTP). The overall information transmitted in the system increases until reaching a PTP, and then decreases until convergence. This is illustrated inf Fig. 2 (left). A PTP based characterization can not be applied in the case of the Schrödinger operator, due to its non-ergodicity and the existence of several PTPs. However, we observe again a difference in phase transition frequency depending on the structure of the graph.

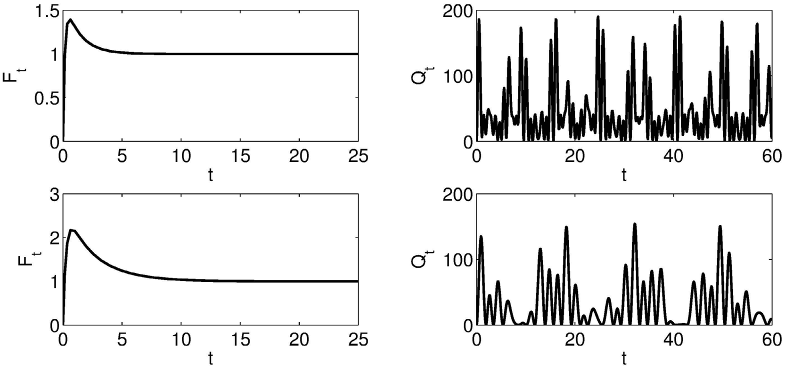


Fig. 2. Heat flow (left) and QEF (right) for two different 10 node graphs: a 2×5 grid graph with 8 neighbour connectivity (top) and a circle graph (bottom). In both cases, the x axis represents time.

3.3 Frequency Domain Analysis of the Schrödinger Operator

The results and analysis above suggest a correlation between graph structure and both the Schrödinger operator and the QEF frequency patterns. We therefore propose a graph characterization based on the QEF in the frequency domain. In order to obtain this characterization, we consider the QEF as a non-periodic signal: we select a time interval $[0, T]$ and we apply the Fast Fourier Transform to the QEF. We refer to this representation as the *frequency domain trace*. The frequency domain trace for the graphs in Fig. 2 can be seen in Fig. 3. The first conclusion from these plots is that the more complex graphs are characterized by the presence of higher frequencies.

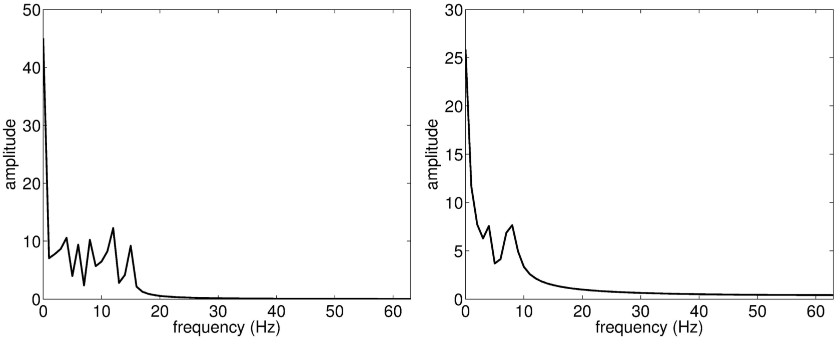


Fig. 3. Frequency domain trace obtained from the quantum energy flow of the grid graph (left) and the circle graph (right) in Fig. 2

However, this representation depends on graph size. Fig. 4 (left) shows the frequency domain trace for four differently sized line graphs. This plot demonstrates that the maximum spectral amplitude is proportional to the graph size. In order to compare arbitrarily sized graphs we apply a simple frequency domain trace normalization based on its maximum amplitude. The result of this normalization can be seen in Fig. 4 (right).

During our experiments we will represent graphs by means of a *cumulative frequency domain trace*, obtained by accumulating the normalized amplitudes

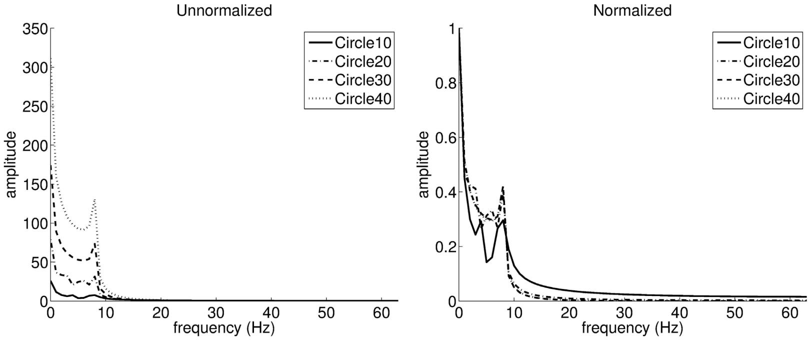


Fig. 4. Unnormalized (left) and normalized (right) frequency domain traces for four different size circle graphs (10, 20, 30 and 40 nodes)

from lower to higher frequencies of their corresponding frequency domain traces. In Fig. 5 we compare the cumulative frequency domain trace obtained from five graphs and their corresponding heat flows. In the case of the cumulative frequency domain trace, the area under the curve provides a good estimate of graph complexity. Simpler graphs yield larger areas. The PTPs of the corresponding heat flow traces also provide a good complexity estimate. In this case, the PTP for simple graphs is reached later in time. However, in this particular example, the heat flow trace estimates the complexity of the line graph to be lower than that of the circle graph. That is not the case of the cumulative frequency domain trace, for which the complexity of the line graph is higher.

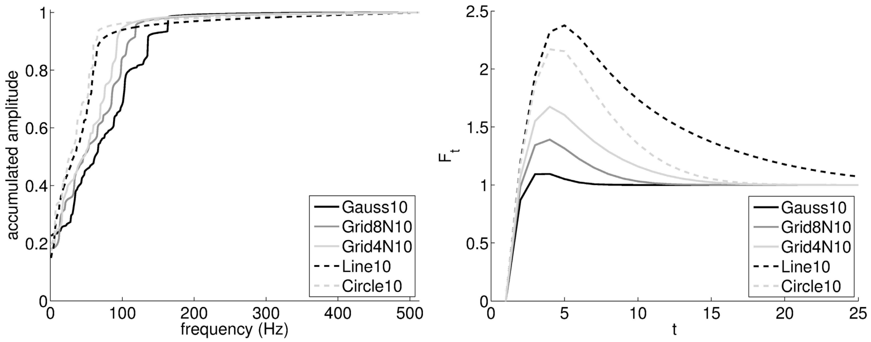


Fig. 5. Cumulative frequency domain traces (left) and heat flow (right) for five simple 10 node graphs: a random graph (Gauss10), a 8-connected 2x5 grid (Grid8N10), a 4-connected 2x5 grid (Grid4N10), a line graph (Line10) and a circular graph (Circle10)

4 Experimental Results

4.1 Noise Sensitivity

The aim of this first experiment is to show the sensitivity of frequency domain traces to graph noise. We first constructed a base 400 nodes random graph by means of the Erdös-Rényi model [10]. We then compared the frequency domain trace of the base graph to those obtained after applying random edit operations on it. In this experiment we only applied edge removal operations, and thus, in each iteration, we remove a random edge from the base graph and we compute the Euclidean distance between the unnormalized traces. The results are shown in Fig. 6. Four experiments were performed, using four different time intervals $[0..T]$ to construct the frequency domain traces.

From Fig. 6, it is clear that the final trace is not strongly affected by small disturbances. For larger time intervals there appears to be a significant sensitivity to noise. However, difference between traces is still low. The remainder of the experiments in this paper are conducted after setting $T = 1024$.

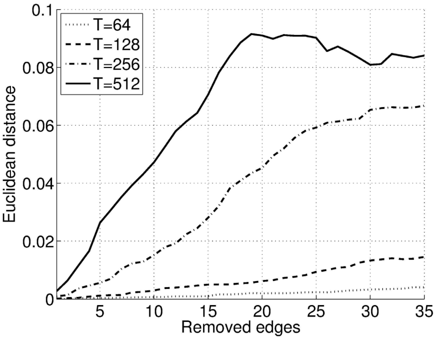


Fig. 6. Results of the noise sensitivity experiment. Number of edit operations (edge removals) versus distance between edited graph's frequency domain trace and base graph's one, for four different T values.

4.2 Characterization of Synthetic Data

In order to test the discriminative power of our characterization we constructed a dataset of synthetic graphs. The dataset consists of three groups of 32 graphs, each group characterized by a different graph structure. All of the graphs in the dataset have 90 nodes. The graphs in the first group are random graphs constructed using the Erdös-Rényi [10] model, in which each pair of nodes is linked by an edge with probability given by p. In our experiments we set $p = 0.1$. The graphs in the second group belong to the category of scale free graphs (i.e. graphs for which its degree distribution follows a power law), and were constructed using the Barabási and Albert's model [11]. In this model we have set $m_0 = 5$ for the initial size of the graphs and $m = 2$ for the number of links to add during each iteration, following the addition of a node. Finally, the graphs in the third group correspond to small world graphs (i.e. graphs in which most nodes are not neighbours of each other, but in which average path length between a graph pair of nodes is small). These small world graphs are generated by means of the Watts and Strogatz algorithm [12]. In this case we set the mean degree value to $K = 10$ and the rewiring probability to $p = 0.2$.

A cumulative frequency domain trace was computed for all graphs in the set, and the results are shown in Fig. 7. The first conclusion of our experiment is that these traces clearly discriminate between different graph structures. This conclusion is supported by a Multidimensional Scaling analysis (MDS) of the traces (also shown in Fig. 7). The aim of MDS is to apply dimensionality reduction on data while preserving relative distances between patterns. If we project the traces onto a 2D space, the graphs in the three groups are clearly split into three different clusters with high intra-cluster homogeneity and high-inter cluster separability.

In Fig. 7 we explore the relationship between frequency and structure. The frequency spectrum of random graphs is characterized by higher amplitudes at high frequencies. In the case of small world graphs, the predominant frequencies

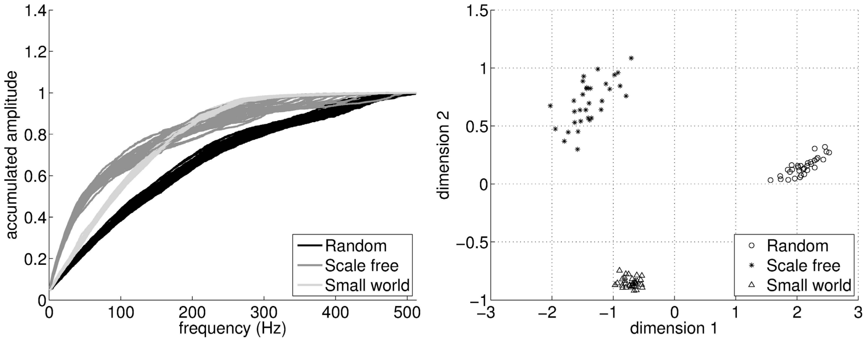


Fig. 7. Characterization of synthetic graphs. Left: cumulative frequency domain traces. Right: MDS results.

are in the middle part of the spectrum. Scale free graphs are characterized by higher amplitudes at lower frequencies. These results suggest that the structure of random graphs is more complex in the sense that it imposes more constraints to the distribution of energy on the graph. As a consequence, energy waves exhibit higher frequency as they propagate. Scale free and small world graphs impose less restrictions on the distribution of energy through the graph, and are associated with lower frequency patterns.

4.3 Characterization of Real-World Data

Our aim in this experiment was to evaluate the validity of our method when applied to real-world data. The 24 graphs studied in this experiment are part of a dataset that has been previously utilized for complex network characterization [14] or network robustness assessment [15]. Our subset of graphs is divided into two categories: a) 9 networks having a homogeneous degree distribution and b) 15 networks having a power law degree distribution. The first category consists of the following graphs: Benguela, Reef Small, Coachella Valley, Shelf, Skipwith, St. Marks Seagrass and Stony food webs and two Macaque visual cortex networks. The second category contains a more heterogeneous set of graphs: four software networks (Abi, Digital, VTK and XMMS), a network of sexual partners in Colorado Springs, a network of injectable drugs users, the airport transportation network in the US in 1997, the Scotch Broom food web, two transcription interaction networks concerning E. Coli and yeast and five different protein interaction networks. In Fig. 8 we show the cumulative frequency domain traces for all the aforementioned graphs. It must be noted that all of them vary widely in size and edge density.

The results of this experiment demonstrate that the relationship between frequency and structure is also held in the context of real-world data. Networks having a homogeneous degree distribution are not characterized by any predominant frequency. Therefore, this type of network produces an almost diagonal

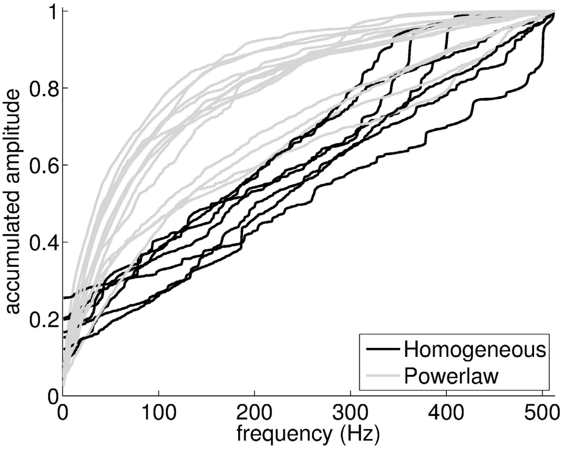


Fig. 8. Characterization of real-world networks

cumulative frequency domain trace. Networks following a power law degree distribution are mainly characterized by lower and medium frequencies. Power law degree distribution arise as a feature of scale free networks, in which there exist nodes with a degree that greatly exceeds the average in the network. This kind of structure is commonly associated with the existence of *hub* nodes that increase the overall robustness of the network, thus improving network connectedness. This observation again supports that the correlation between the lack of constraints to energy propagation on the graph and the predominance of lower frequencies in its characteristic trace.

4.4 Network Dynamics Analysis

In this last experiment we apply our characterization method to the analysis of dynamic network structures, in order to test if such characterization can give an insight into the existence of structural changes with time. We computed the traces for several graphs generated according to the activity-based preferential attachment (APA) model [16]. This model has proved to be the best approximation of the evolution of several real-world cortical networks. The APA model is a generalization of the Barabási and Albert's model, in which new connections are established proportionally to a dynamical process on the entire network, rather than according to a local structural property. Nodes with higher activity have a higher probability of establishing new connections. In the APA model, the activity of a node i is computed from the stationary distribution π of frequency of visits to nodes of a random walk, where $\pi_i = \lim_{t \to \infty} v_i/t$ and v_i the number of times the random walker visits the node i after t time steps.

The plot in Fig. 9 shows the evolution of the APA cumulative frequency domain trace over time for one of the realizations of the model. The plots obtained from other realizations were very similar. Network evolution starts at $t = 0$ with a fully connected network having 5 nodes. At each time step a new node is added, following the APA model. The process is repeated until $t = 1000$. It must be noted that the graphs constructed using the this model are directed. They were converted to undirected ones by simply transforming each directed arc into an undirected edge before computing the graph characteristic trace.

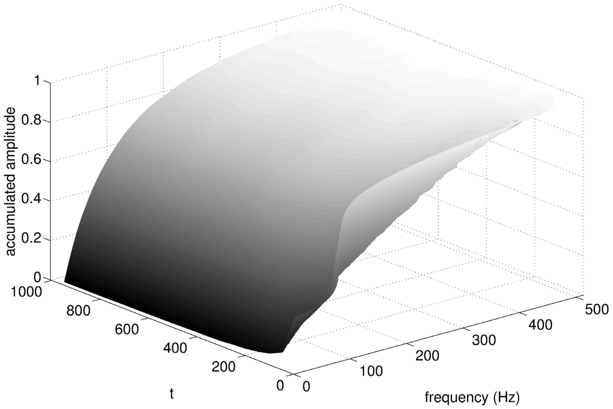


Fig. 9. Cumulative frequency domain trace over time for one dynamic graph

The results show that the structure of a graph modeled by means of activity-based preferential attachment is not subject to significant changes after an early stage in the evolutionary process. This was the expected outcome of the experiment since the APA model is intended to conserve the overall structural properties of the network through time. Moreover, this experiment shows that network growth does not have an impact on its cumulative frequency domain trace as long as global structure is conserved due to the frequency domain normalization process, making the cumulative frequency domain trace capable of comparing the structure of networks having a different amount of nodes.

5 Conclusions and Future Work

Heat flow, based on the heat kernel, has been successfully used to characterize graph structure. The aim of the present paper was to answer the question of whether the Schrödinger operator (the solution to the Schrödinger equation) can be used also to characterize graph structure. After analyzing energy distribution through the graph based on the Schrödinger operator, we introduced a new characterization method based on the analysis in the frequency domain of the quantum equivalent of the heat flow trace that relates frequency to graph structure. Experiments performed both on synthetic and real-world datasets show that the *cumulative frequency domain trace* is a useful tool for graph analysis, that is not sensitive to small changes in graph structure.

However, based on these promising preliminary results, further in depth analysis is required. Firstly, and similarly to heat flow, the cumulative frequency domain trace does not provide us with a quantitative measure to directly compare graph structures. A first step in this direction could be to apply this trace as part of the thermodynamic depth complexity measurement framework, in order to obtain a numerical representation of graph structure [4][13]. Secondly, during our analysis of the Schrödinger operator we detected the presence of symmetric energy distribution patterns on the graph. We could analyze how this symmetry depends on graph structure and whether the results of this analysis are related to previous work on symmetry detection based on quantum walks [7]. Finally, an additional future work idea comes from the results of the experiment on the dynamic dataset. This experiment proved that it would be of great interest to extend our algorithm to the directed graphs domain.

References

1. Aziz, F., Wilson, R.C., Hancock, E.R.: Graph Characterization via Backtrackless Paths. In: Pelillo, M., Hancock, E.R. (eds.) SIMBAD 2011. LNCS, vol. 7005, pp. 149–162. Springer, Heidelberg (2011)
2. Peng, R., Wilson, R.C., Hancock, E.R.: Graph Characterization vi Ihara Coefficients. IEEE Transactions on Neural Networks 22(2), 233–245 (2011)
3. Das, K.C.: Extremal Graph Characterization from the Bounds of the Spectral Radius of Weighted Graphs. Applied Mathematics and Computation 217(18), 7420–7426 (2011)
4. Escolano, F., Hancock, E.R., Lozano, M.A.: Heat Diffusion: Thermodynamic Depth Complexity of Networks. Physical Review E 85(3), 036206(15) (2012)
5. Xiao, B., Hancock, E.R., Wilson, R.C.: Graph Characteristics from the Heat Kernel Trace. Pattern Reognition 42(11), 2589–2606 (2009)
6. Aubry, M., Schlickewei, U., Cremers, D.: The Wave Kernel Signature: A Quantum Mechanical Approach To Shape Analysis. In: IEEE International Conference on Computer Vision (ICCV), Workshop on Dynamic Shape Capture and Analysis (4DMOD) (2011)
7. Rossi, L., Torsello, A., Hancock, E.R.: Approximate Axial Symmetries from Continuous Time Quantum Walks. 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. 144–152. Springer, Heidelberg (2012)
8. Emms, D., Wilson, R.C., Hancock, E.R.: Graph Embedding Using Quantum Commute Times. In: Escolano, F., Vento, M. (eds.) GbRPR. LNCS, vol. 4538, pp. 371–382. Springer, Heidelberg (2007)
9. Farhi, E., Gutmann, S.: Quantum Computation and Decision Trees. Physical Review A 58, 915–928 (1998)
10. Erdös, P., Rényi, A.: On Random Graphs. I. Publicationes Mathematicae 6, 290–297 (1959)
11. Barabási, A.L., Albert, R.: Emergence of Scaling in Random Networks. Science 286(5439), 509–512 (1999)

12. Watts, D.J., Strogatz, S.H.: Collective Dynamics of 'Small-World' Networks. Nature 393(6684), 440–442 (1998)
13. Han, L., Escolano, F., Hancock, E., Wilson, R.: Grapph Characterizations From Von Neumann Entropy. Pattern Recognition Letters 33(15), 1958–1967 (2012)
14. Estrada, E.: Quantifying Network Heterogeneity. Physical Review E 82(6), 066102 (2010)
15. Estrada, E.: Network Robustness to Targeted Attacks. The Interplay of Expansibility and Degree Distribution. The European Physical Journal B - Condensed Matter and Complex Systems 52(4), 563–574 (2006)
16. Antiqueira, L., Rodrigues, F.A., Costa, L.F.: Modeling Connectivity in terms of Network Activity. J. Stat. Mech., L09005 (2009)

Attributed Graph Similarity from the Quantum Jensen-Shannon Divergence

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

[1] Department of Environmental Science, Informatics and Statistics,
Ca' Foscari University of Venice, Italy
{lurossi,torsello}@dsi.unive.it

[2] Department of Computer Science, University of York, YO10 5GH, UK
edwin.hancock@york.ac.uk

Abstract. One of the most fundamental problem that we face in the graph domain is that of establishing the similarity, or alternatively the distance, between graphs. In this paper, we address the problem of measuring the similarity between attributed graphs. In particular, we propose a novel way to measure the similarity through the evolution of a continuous-time quantum walk. Given a pair of graphs, we create a derived structure whose degree of symmetry is maximum when the original graphs are isomorphic, and where a subset of the edges is labeled with the similarity between the respective nodes. With this compositional structure to hand, we compute the density operators of the quantum systems representing the evolution of two suitably defined quantum walks. We define the similarity between the two original graphs as the quantum Jensen-Shannon divergence between these two density operators, and then we show how to build a novel kernel on attributed graphs based on the proposed similarity measure. We perform an extensive experimental evaluation both on synthetic and real-world data, which shows the effectiveness the proposed approach.

Keywords: Graph Similarity, Graph Kernel, Continuous-Time Quantum Walk, Quantum Jensen-Shannon Divergence.

1 Introduction

Graph-based representations have become increasingly popular due to their ability to characterize in a natural way a large number of systems which are best described in terms of their structure. Concrete examples include the use of graphs to represent shapes [1], metabolic networks [2], protein structure [3], and road maps [4]. However, the rich expressiveness and versatility of graphs comes at a cost. In fact, our ability to analyse data abstracted in terms of graphs is severely limited by the restrictions posed by standard pattern recognition techniques, which usually require the graphs to be first embedded into a vectorial space, a procedure which is far from being trivial. The reason for this is that there is no canonical ordering for the nodes in a graph and a correspondence order

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 204–218, 2013.

must be established before analysis can commence. Moreover, even if a correspondence order can be established, graphs do not necessarily map to vectors of fixed length, as the number of nodes and edges can vary.

One of the most fundamental problem that we need to face in the graph domain is that of measuring the similarity, or alternatively the distance, between graphs. Generally, the similarity between two graphs can be defined in terms of the lowest cost sequence of edit operations, for example, the deletion, insertion and substitution of nodes and edges, which are required to transform one graph into the other [5]. Another approach is that of Barrow and Burstall [6], where the similarity of two graphs is characterized using the cardinality of their maximum common subgraphs. Similarly, Bunke and Shearer [7] introduced a metric on unattributed graphs based on the maximum common subgraph, which later Hidović and Pelillo extended to the case of attributed graphs [8]. Unfortunately, both computing the graph edit distance and finding the maximum common subgraphs turn out to be a computationally hard problem.

Closely related to this problem is that of defining a kernel [9] over graphs. Graph kernels are powerful tools that allow the researcher to overcome the restrictions posed by standard pattern recognition techniques by shifting the problem from that of finding an embedding of a graph to that of defining a positive semidefinite kernel, via the well-known kernel trick. In fact, once we define a positive semidefinite kernel $k : X \times X \rightarrow \mathbb{R}$ on a set X, then we know that there exists a map $\phi : X \rightarrow H$ into a Hilbert space H, such that $k(x, y) = \phi(x)^\top \phi(y)$ for all $x, y \in X$. Thus, any algorithm that can be formulated in terms of scalar products of the $\phi(x)$s can be applied to a set of data on which we have defined our kernel. However, due to the rich expressiveness of graphs, the problem of defining effective graph kernels has proven to be extremely difficult.

Many different graph kernels have been proposed in the literature [10,11,12]. Graph kernels are generally instances of the family of R-convolution kernels introduced by Haussler [13]. The fundamental idea is that of defining a kernel between two discrete objects by decomposing them and comparing some simpler substructures. For example, Gärtner et al. [10] propose to count the number of common random walks between two graphs, while Borgwardt and Kriegel [11] measure the similarity based on the shortest paths in the graphs. Shervashidze et al. [12], on the other hand, count the number of graphlets, i.e. subgraphs with k nodes. These kernels can be generally defined both on unattributed and attributed graphs, where in the attributed case one simply enumerates the number of substructures which share the same sequence of labels.

In this paper, we introduce a novel similarity measure between attributed graphs which is based on the evolution of a continuous-time quantum walk [14]. In particular, we are taking advantage of the fact that the interference effects introduced by the quantum walk seem to be enhanced by the presence of symmetrical motifs in the graph [15,16]. Thus, given a pair of graphs, we create a derived structure whose degree of symmetry is maximum when the original graphs are isomorphic. To encode the information on the node attributes, in the new structure we will label the edges connecting one graph to the other with the value of the similarity between the corresponding nodes. With this structure to hand, we will define two continuous-time quantum walks which have orthogonal density operators under the evolution of the walk whenever the two original graphs are isomorphic. Then, to define the similarity measure we make use of the quantum Jensen-Shannon divergence, a measure which has recently been introduced as a means to compute the distance between quantum states [17]. Finally, we use the proposed similarity measure to define a novel kernel for attributed graphs.

The remainder of this paper is organized as follows: Section 2 provides an essential introduction to the basic terminology required to understand the proposed quantum mechanical framework. In Section 3 we introduce our similarity measure and we define a novel attributed graph kernel. Section 4 illustrates the experimental results, and the conclusions are presented in Section 5.

2 Quantum Mechanical Background

Quantum walks are the quantum analogue of classical random walks [14]. In this paper we consider only continuous-time quantum 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 (i.e. a matrix whose columns sum to unity), 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. Hence, the evolution of the quantum walk is reversible, which implies that quantum walks are non-ergodic and do not possess a limiting distribution. Using 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 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

$$\Pr(X^t = u) = \alpha_u(t)\alpha_u^*(t) \tag{2}$$

where $\alpha_u^*(t)$ is the complex conjugate of $\alpha_u(t)$. Moreover $\alpha_u(t)\alpha_u^*(t) \in [0, 1]$, for all $u \in V$, $t \in \mathbb{R}^+$, and in a closed system $\sum_{u \in V} \alpha_u(t)\alpha_u^*(t) = 1$.

Recall that the adjacency matrix of the graph G has elements

$$A_{uv} = \begin{cases} 1 \text{ if } (u, v) \in E \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

The evolution of the walk is governed by Schrödinger equation, where we take the Hamiltonian of the system to be the graph adjacency matrix, which yields

$$\frac{d}{dt}\left|\psi_t\right\rangle = -iA\left|\psi_t\right\rangle \tag{4}$$

Thus, given an initial state $\left|\psi_0\right\rangle$, we can solve Equation 4 to determine the state vector at time t

$$\left|\psi_t\right\rangle = e^{-iAt}\left|\psi_0\right\rangle \tag{5}$$

With the graph adjacency matrix to hand, we can compute its spectral decomposition $A = \Phi\Lambda\Phi^\top$, where Φ is the $n \times n$ matrix $\Phi = (\phi_1|\phi_2|...|\phi_n)$ with the ordered eigenvectors as columns and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_n)$ is the $n \times n$ diagonal matrix with the ordered eigenvalues as elements, such that $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$. Using this spectral decomposition and the fact that $e^{-iAt} = \Phi e^{-i\Lambda t}\Phi^\top$ we can finally re-write Eq. 5 as

$$\left|\psi_t\right\rangle = \Phi e^{-i\Lambda t}\Phi^\top\left|\psi_0\right\rangle \tag{6}$$

2.1 Quantum Jensen-Shannon Divergence

A pure state is defined as a state that can be described by a ket vector $\left|\psi_i\right\rangle$. Consider a quantum system that can be in a number of states $\left|\psi_i\right\rangle$ each with probability p_i. The system is said to be in the ensemble of pure states $\{\left|\psi_i\right\rangle, p_i\}$. The density operator (or density matrix) of such a system is defined as

$$\rho = \sum_i p_i \left|\psi_i\right\rangle\left\langle\psi_i\right| \tag{7}$$

The Von Neumann entropy [18] of a density operator ρ is

$$H_N(\rho) = -Tr(\rho\log\rho) = -\sum_j \lambda_j \log\lambda_j\,, \tag{8}$$

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

$$D_{JS}(\rho,\sigma) = H_N\left(\frac{\rho+\sigma}{2}\right) - \frac{1}{2}H_N(\rho) - \frac{1}{2}H_N(\sigma) \tag{9}$$

This quantity is always well defined, symmetric and negative definite. It can also be shown that $D_{JS}(\rho,\sigma)$ is bounded, i.e.

$$0 \leq D_{JS}(\rho,\sigma) \leq 1 \tag{10}$$

Let $\rho = \sum_i p_i\rho_i$ be a mixture of quantum states ρ_i, with $p_i \in \mathbb{R}^+$ such that $\sum_i p_i = 1$, then we can prove that

$$H_N(\sum_i p_i\rho_i) \leq H_S(p_i) + \sum_i p_i H_N(\rho_i) \tag{11}$$

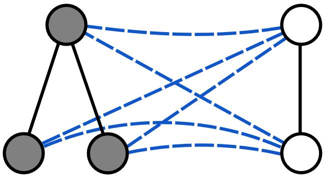


Fig. 1. Given two graphs $G_1(V_1, E_1, \nu_1)$ and $G_2(V_2, E_2, \nu_2)$ we build a new graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = V_1 \cup V_2$, $\mathcal{E} = E_1 \cup E_2$ and we add a new edge (u, v) between each pair of nodes $u \in V_1$ and $v \in V_2$.

where the equality is attained if and only if the states ρ_i have support on orthogonal subspaces, where the support of an operator is the subspace spanned by the eigenvectors of the operator with non-zero eigenvalues. By setting $p_1 = p_2 = 0.5$, we see that

$$D_{JS}(\rho, \sigma) = H_N\left(\frac{\rho + \sigma}{2}\right) - \frac{1}{2}H_N(\rho) - \frac{1}{2}H_N(\sigma) \leq 1 \tag{12}$$

Hence D_{JS} is always less or equal than 1, and the equality is attained only if ρ and σ have support on orthogonal subspaces.

3 A Similarity Measure for Attributed Graphs

Given two graphs $G_1(V_1, E_1, \nu_1)$ and $G_2(V_2, E_2, \nu_2)$, where ν_1 and ν_2 are respectively the functions assigning attributes to the nodes of G_1 and G_2, we build a new graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$ where $\mathcal{V} = V_1 \cup V_2$, $\mathcal{E} = E_1 \cup E_2 \cup E_{12}$, and $(u, v) \in E_{12}$ only if $u \in V_1$ and $v \in V_2$ (see Fig. 1 for an example). Moreover, the edges $(u, v) \in E_{12}$ are labeled with a real value $\omega(\nu_1(u), \nu_2(v))$ representing the similarity between $\nu_1(u)$ and $\nu_2(v)$. With this new structure to hand, we define two continuous-time quantum walks $\left|\psi_t^-\right\rangle = \sum_{u \in V} \psi_{0u}^- \left|u\right\rangle$ and $\left|\psi_t^+\right\rangle = \sum_{u \in V} \psi_{0u}^+ \left|u\right\rangle$ on $\mathcal{G}$ with starting states

$$\psi_{0u}^- = \begin{cases} +\frac{d_u}{C} & \text{if } u \in G_1 \\ -\frac{d_u}{C} & \text{if } u \in G_2 \end{cases} \qquad \psi_{0u}^+ = \begin{cases} +\frac{d_u}{C} & \text{if } u \in G_1 \\ +\frac{d_u}{C} & \text{if } u \in G_2 \end{cases} \tag{13}$$

where d_u is the degree of the node u and C is the normalisation constant such that the probabilities sum to one. Note that the walk will spread at a speed proportional to the edge weights, which means that given an edge $(u, v) \in E_{12}$, the more similar $\nu_1(u)$ and $\nu_2(v)$ are, the faster the walker will propagate along the inter graphs connection (u, v). On the other hand, the intra-graph connection weights, which are not dependent on the nodes similarity, will not affect the propagation speed.

Given this setting, we allow the two quantum walks to evolve until a time T, and we define the average density operators ρ_T and σ_T over this time as

$$\rho_T = \frac{1}{T}\int_0^T \left|\psi_t^-\right\rangle \left\langle\psi_t^-\right| \, \mathrm{d}t \qquad \sigma_T = \frac{1}{T}\int_0^T \left|\psi_t^+\right\rangle \left\langle\psi_t^+\right| \, \mathrm{d}t \tag{14}$$

In other words, we have defined two mixed systems with equal probability of being in any of the pure states defined by the evolution of the quantum walks.

In the next section we will prove that, whenever G_1 and G_2 are isomorphic, the quantum Jensen-Shannon divergence between ρ_T and σ_T will be maximum, i.e., it will be equal to 1. Hence, it seems reasonable to use the value of the quantum Jensen-Shannon divergence as a measure of the similarity between the two graphs. In particular, in the next section we use the QJSD to define a novel kernel for attributed graphs.

3.1 A QJSD Kernel for Attributed Graphs

Given two attributed graphs G_1 and G_2, we define the quantum Jensen-Shannon kernel $k_T(G_1, G_2)$ between them as

$$k_T(G_1, G_2) = D_{JS}(\rho_T, \sigma_T) \tag{15}$$

where ρ_T and σ_T are the density operators defined as in Eq. 14. Note that in this formulation the kernel is parametrised by the time variable T. As it is not clear how we should set this parameter, in this paper we propose to let $T \to \infty$, i.e., we compute $\lim_{T \to +\infty} k_T(G_1, G_2)$. In the following section we will show how to compute analytically this limit.

We now proceed to show some interesting properties of our kernel. First, however, we need to prove the following

Lemma 1. *If G_1 and G_2 are two isomorphic graphs, then ρ_T and σ_T have support on orthogonal subspaces.*

Proof. We need to prove that

$$(\rho_T)^\dagger \sigma_T = \frac{1}{T^2} \int_0^T \rho_{t_1}\, \mathrm{d}t_1 \int_0^T \sigma_{t_2}\, \mathrm{d}t_2 = \mathbf{0} \tag{16}$$

where $\mathbf{0}$ is the matrix of all zeros, $\rho_t = \left|\psi_t^-\right\rangle \left\langle\psi_t^-\right|$ and $\sigma_t = \left|\psi_t^+\right\rangle \left\langle\psi_t^+\right|$. Note that if $\rho_{t_1}^\dagger \sigma_{t_2} = \mathbf{0}$ for every t_1 and t_2, then $\rho^\dagger \sigma = \mathbf{0}$. We now prove that if G_1 is isomorphic to G_2 then $\left\langle \psi_{t_1}^- \middle| \psi_{t_2}^+ \right\rangle = 0$ for every t_1 and t_2.

If $t_1 = t_2 = t$, then

$$\left\langle \psi_0^- \right| (U^t)^\dagger U^t \left|\psi_0^+\right\rangle = 0 \tag{17}$$

since $(U^t)^\dagger U^t$ is the identity matrix and the initial states are orthogonal by construction. On the other hand, if $t_1 \neq t_2$, we have

$$\left\langle \psi_0^- \right| U^{\Delta t} \left|\psi_0^+\right\rangle = 0 \tag{18}$$

where $\Delta_t = t_2 - t_1$.

To conclude the proof we rewrite the previous equation as

$$\begin{aligned} \langle \psi_0^- \,|\, U^{\Delta t} \,|\psi_0^+ \rangle &= \sum_k \psi_{k0}^+ \sum_l \psi_{l0}^+ U_{lk}^{\Delta t} \\ &= \sum_{k_1} \psi_{k_1 0}^+ \sum_l \psi_{l0}^+ U_{lk_1}^{\Delta t} - \sum_{k_2} \psi_{k_2 0}^+ \sum_l \psi_{l0}^+ U_{lk_2}^{\Delta t} \\ &= \sum_l \psi_{l0}^+ \left(\sum_{k_1} \psi_{k_1 0}^+ U_{lk_1}^{\Delta t} - \sum_{k_2} \psi_{k_2 0}^+ U_{lk_2}^{\Delta t} \right) = 0 \end{aligned} \tag{19}$$

where the indices l, k, run over the nodes of $\mathcal{G}$, and k_1 and k_2 run over the nodes G_1 and G_2 respectively.

To see that Eq. 19 holds, note that U is a symmetric matrix and it is invariant to graph symmetries, i.e., if u and v are symmetric then $U_{uu}^{\Delta t} = U_{vv}^{\Delta t}$, and that if G_1 and G_2 are isomorphic, then $k_1 = k_2$ and $\psi_{1:k_1 0}^+ = \psi_{k_1+1:k_2 0}^+$. Recall that $U^t = e^{-iAt}$, where A is the graph adjacency matrix. A symmetry orbit is defined as a group of vertices where v_1 and v_2 belong to the same orbit if there is an automorphism $\sigma \in Aut(G)$ such that $\sigma v_1 = v_2$, where $Aut(G)$ is the set of automorphisms of G. In other words, if u and v belong to a symmetry orbit, there exists an automorphism of the graph with a corresponding permutation matrix P such that

$$A = P^\top A P \tag{20}$$

and

$$P \,|e_u\rangle = |e_v\rangle \tag{21}$$

This in turn implies that the graph adjacency matrix is invariant to symmetries. As we will show, the same holds for the unitary operator of the quantum walk. In fact, given the spectral decomposition of $A = \Phi \Lambda \Phi^\top$, we can see that the following equality holds

$$\Phi \Lambda \Phi^\top = P^\top (\Phi \Lambda \Phi^\top) P \tag{22}$$

and thus

$$\Phi = P^\top \Phi \tag{23}$$

Let us now write the unitary operator in terms of the adjacency matrix eigendecomposition, which yields

$$e^{-iAt} = \Phi e^{-i\Lambda t} \Phi^\top \tag{24}$$

From Equations 23 and 24 it follows that

$$\Phi e^{-i\Lambda t} \Phi^\top = P^\top \Phi e^{-i\Lambda t} \Phi^\top P \tag{25}$$

This in turn implies that if u and v are symmetrical then $U_{uu}^t = U_{vv}^t$, which concludes the proof.

Corollary 1. *Given a pair of graphs G_1 and G_2, the kernel satisfies the following properties: 1) $0 \leq k(G_1, G_2) \leq 1$ and 2) if G_1 and G_2 are isomorphic, then $k(G_1, G_2) = 1$.*

Proof. The first property is trivially proved by noting that, according to Eq. 15, the kernel between G_1 and G_2 is defined as the quantum Jensen-Shannon divergence between two density operators, and then recalling that the value of quantum Jensen-Shannon divergence is bounded to lie between 0 and 1.

The second property follows again from Eq. 15 and Theorem 1. It is sufficient to note that the quantum Jensen-Shannon divergence reaches its maximum value if and only if the density operators have support on orthogonal spaces.

Unfortunately we cannot prove that our kernel is positive semidefinite, but both empirical evidence and the fact that the Jensen-Shannon Divergence is negative semidefinite on pure quantum states [21] while our graph kernel is maximal on orthogonal states suggest that the kernel constraints are never violated in practice.

3.2 Kernel Computation

We conclude this section with a few remarks on the computational complexity of our kernel. In particular, we show that the solutions to Eq. 14 can be computed analytically. Recall that $|\psi_t\rangle = e^{-iAt} |\psi_0\rangle$, then we rewrite Eq. 14 as

$$\rho_T = \frac{1}{T} \int_0^T e^{-iAt} |\psi_0\rangle \langle\psi_0| e^{iAt} \, \mathrm{d}t \tag{26}$$

Since $e^{-iAt} = \Phi e^{-i\Lambda t} \Phi^\top$, we can rewrite the previous equation in terms of the spectral decomposition of the adjacency matrix,

$$\rho_T = \frac{1}{T} \int_0^T \Phi e^{-i\Lambda t} \Phi^\top |\psi_0\rangle \langle\psi_0| \Phi e^{i\Lambda t} \Phi^\top \, \mathrm{d}t \tag{27}$$

The (r, c) element of ρ_T can be computed as

$$\begin{aligned} \rho_T(r,c) = \frac{1}{T} \int_0^T & \left(\sum_k \sum_l \phi_{rk} e^{-i\lambda_k t} \phi_{lk} \psi_{\overline{0l}} \right) \\ & \cdot \left(\sum_m \sum_n \psi^\dagger_{0m} \phi_{mn} e^{i\lambda_n t} \phi_{cn} \right) \mathrm{d}t \end{aligned} \tag{28}$$

Let $\bar{\psi}_k = \sum_l \phi_{lk} \psi_{0l}$ and $\bar{\psi}_n = \sum_m \phi_{mn} \psi^\dagger_{0n}$, then

$$\rho_T(r,c) = \frac{1}{T} \int_0^T \left(\sum_k \phi_{rk} e^{-i\lambda_k t} \bar{\psi}_k \sum_n \phi_{cn} e^{i\lambda_n t} \bar{\psi}_n \right) \mathrm{d}t \tag{29}$$

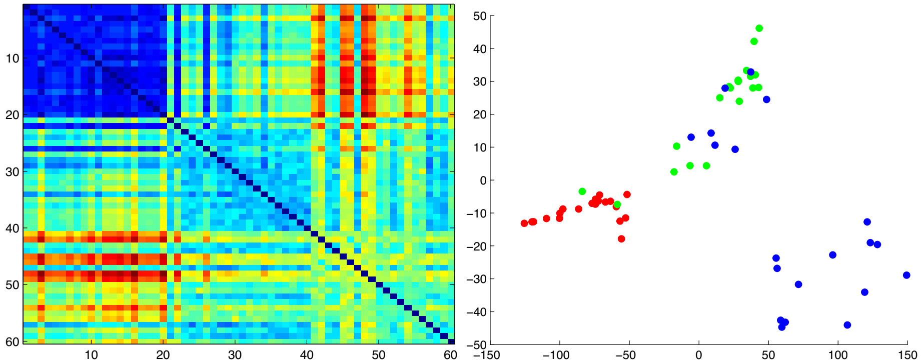


Fig. 2. Edit distance matrix and Multidimensional Scaling of the graph distances for the synthetic dataset

which can be finally rewritten as

$$\rho_T(r,c) = \sum_k \sum_n \phi_{rk}\phi_{cn}\bar{\psi}_k\bar{\psi}_n \frac{1}{T}\int_0^T e^{i(\lambda_n-\lambda_k)t}\,\mathrm{d}t \tag{30}$$

If we let $T \to \infty$, Eq. 30 further simplifies to

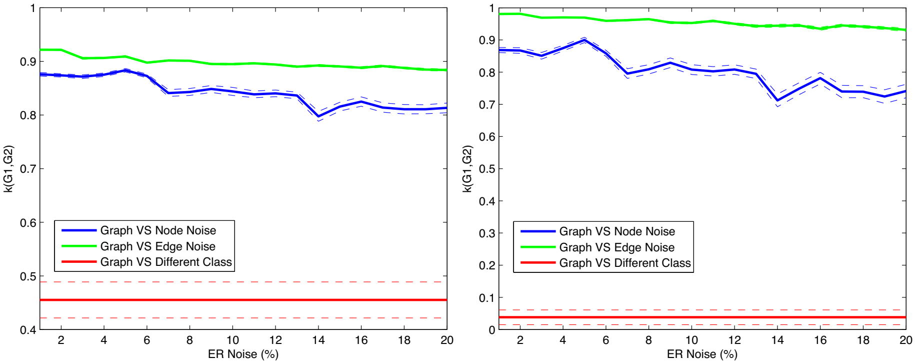
$$\rho_T(r,c) = \sum_{\lambda\in\tilde{\Lambda}} \sum_k \sum_n \phi_{rk}\phi_{cn}\bar{\psi}_k\bar{\psi}_n \tag{31}$$

where $\tilde{\Lambda}$ is the set of distinct eigenvalues of A, while k and n are indices which run over the dimensions of the eigenspace associated with $\lambda \in \tilde{\Lambda}$. As a consequence, we see that the complexity of computing the QJSD kernel is upper bounded by that of computing the eigendecomposition of $\mathcal{G}$, i.e. $O(|\mathcal{V}|^3)$.

4 Experimental Results

In this section, we evaluate the performance of the proposed kernel and we compare it with a number of well-known alternative graph kernels, namely the classic random walk kernel [10], the shortest-path kernel [11] and the 3-nodes graphlet kernel [12], both in their unattributed and attributed versions. Note that since the attributed versions of these kernels are defined only on graphs with categorically labeled nodes, in our experiments we will need to bin the node attributes before computing the kernels.

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) Without Attributes

(b) With Attributes

Fig. 3. The effects of Erdös-Rényi structural noise applied to the nodes and edges of the graph on the kernel value. Using the proposed similarity measure, the noisy versions of the graph belonging to the first class are clearly distinguishable from the instances of the second class. As expected, taking the attributes into account (right) makes the distinction even clearer (note the difference in the scale).

4.1 Synthetic Data

We start by evaluating the proposed kernel on a set of synthetically generated graphs. To this end, we have randomly generated 3 different weighted graph prototypes with size 16, 18 and 20 respectively. For each prototype we started with an empty graph and then we iteratively added the required number of nodes each labeled with a random mean and variance. Then we added the edges and their associated observation probabilities up to a given edge density. Given the prototypes, we sampled 20 observations from each class being careful to discard graphs that were disconnected. Details about the generative model used to sample the graphs can be found in [19]. Figure 2 shows the edit distance matrix of the dataset and the Multidimensional Scaling [20] of the graph distances.

With the synthetic graphs to hand, we initially investigate how the value of the kernel between two graphs varies as we apply Erdös-Rényi noise to the graph structure. In this case the similarity between two nodes u and v is defined as $\omega(u,v) = e^{-\lambda(\nu_1(u)-\nu_2(v))^2}$, where $\nu_1(u)$ and $\nu_2(v)$ are the real-valued attributes associated with u and v respectively. Figure 3 shows the result of this experiment. Here we randomly pick a graph G belonging to class 1, and we compute a number of increasingly noisy versions of it. The noise is applied either to the edges only, i.e. adding or deleting edges, or to the nodes as well, i.e. adding or deleting nodes and edges. We then compute the average value of the kernel between G and its corrupted versions, and we plot it against the average similarity between G and the graphs of class 2. Figure 3 shows that, even at considerably high levels of noise, G is clearly distinguishable from the instances of the second class. As expected, taking the attributes into account renders the distinction even clearer (note the change in the y-scale). However, when augmented with the attributes information, our simi-

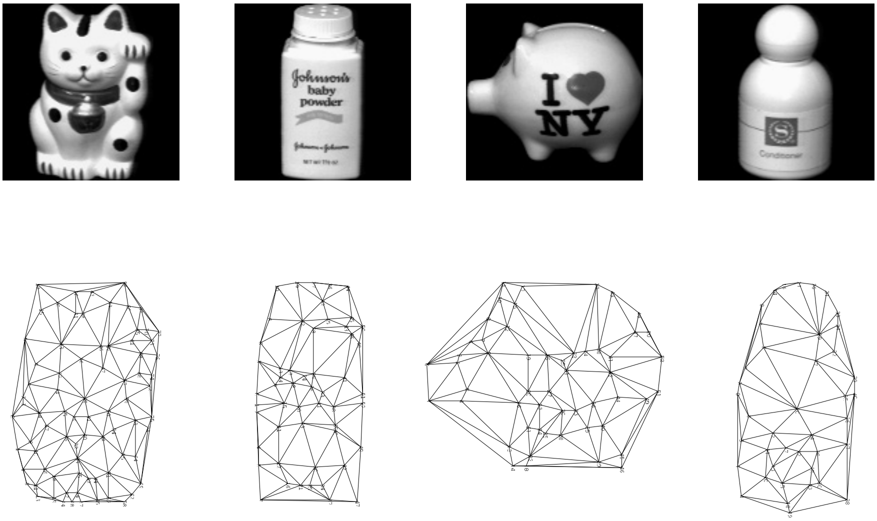


Fig. 4. The four selected objects from the COIL [22] dataset and a sample of their associated Delaunay graphs. Each node of the graphs is labeled with the (x, y) coordinates of the corresponding feature point.

larity measure seems to be slightly more sensitive to structural noise, in particular when the noise is affecting the nodes of the graph.

As a second experiment, we test the accuracy of our kernel in a classification task. The results are shown in Table 1. As we can see, our kernel outperforms or is competitive with the alternatives, and yields a close to 100% average accuracy. Note also that, as expected, taking the similarity between the node attributes into account results in a marked increase in the kernel performance. Quite surprisingly, however, we found that the random walk kernel on the categorically labeled graphs yields a lower performance than its unattributed version.

4.2 Delaunay Graphs

We then tested the efficacy of the proposed kernel on the COIL [22] dataset, which consists of images of different objects, with 72 views of each object obtained from equally spaced viewing directions over 360°. For each image, a graph is obtained by computing the Delaunay triangulation of the corner points extracted by the Harris corner detection algorithm. Moreover, each node is labeled with the (x, y) coordinates of the corresponding feature point. The similarity between two nodes is $\omega(u, v) = e^{-\lambda||\nu_1(u)-\nu_2(v)||_2^2}$, where $||\nu_1(u) - \nu_(v)||_2$ is the Euclidean distance between the two feature points u and v. Here we choose 4 different objects, each with 21 different 5° rotated views. Figure 4 shows the four selected objects together with their associated graphs, while Figure 5 shows the edit distance matrix and the MDS of the graph distances.

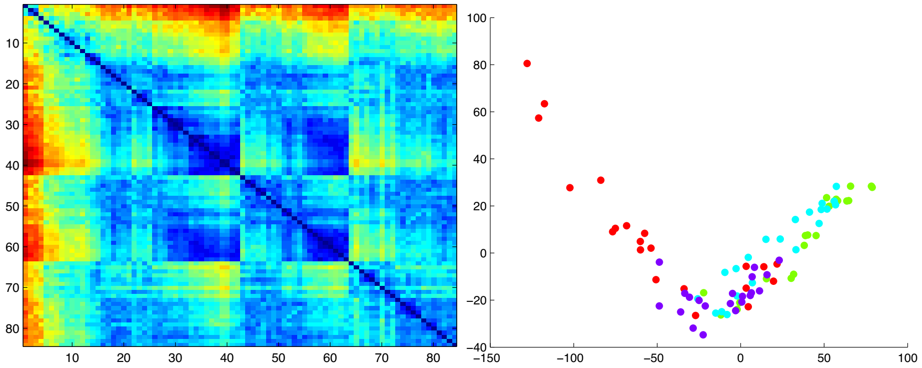


Fig. 5. Edit distance matrix and Multidimensional Scaling of the graph distances for the COIL dataset

We first investigate how integrating the information on the nodes attributes influences the expressive power of our kernel. Figure 6 shows the MDS embedding on the graph distances computed from the unattributed kernel (left) and the attributed one (right). Although the embedding shows that a considerable overlap remains between the different classes, taking the node attributes similarities into account adds a further dimension which can help to discriminate better among the 4 selected objects.

This is indeed reflected in the results of the classification task shown in Table 1. In the attributed case, in fact, the average accuracy of the QJSD kernel is increased by more than 10%, and it outperforms that of all the remaining kernels. Note, however, that if the node labels are dropped, the performance of the QJSD kernel is among the lowest, which once again underlines the importance of incorporating the attributes similarities in the compositional structure.

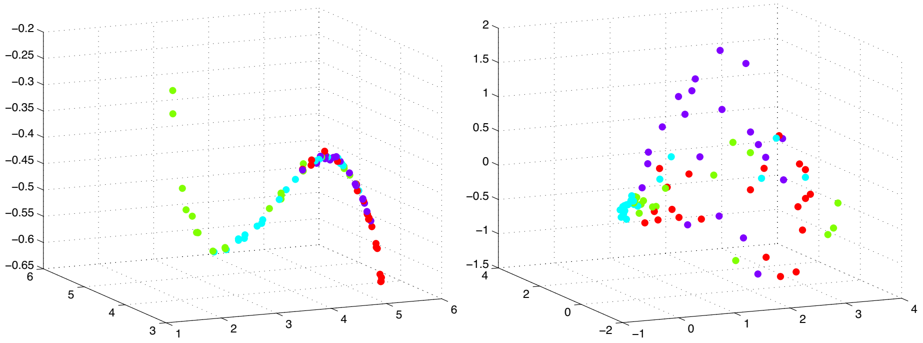


Fig. 6. Multidimensional Scaling of the graph distances computed from the kernel matrix of the COIL dataset. Left, completely structural approach; right, including the information on the nodes attributes.

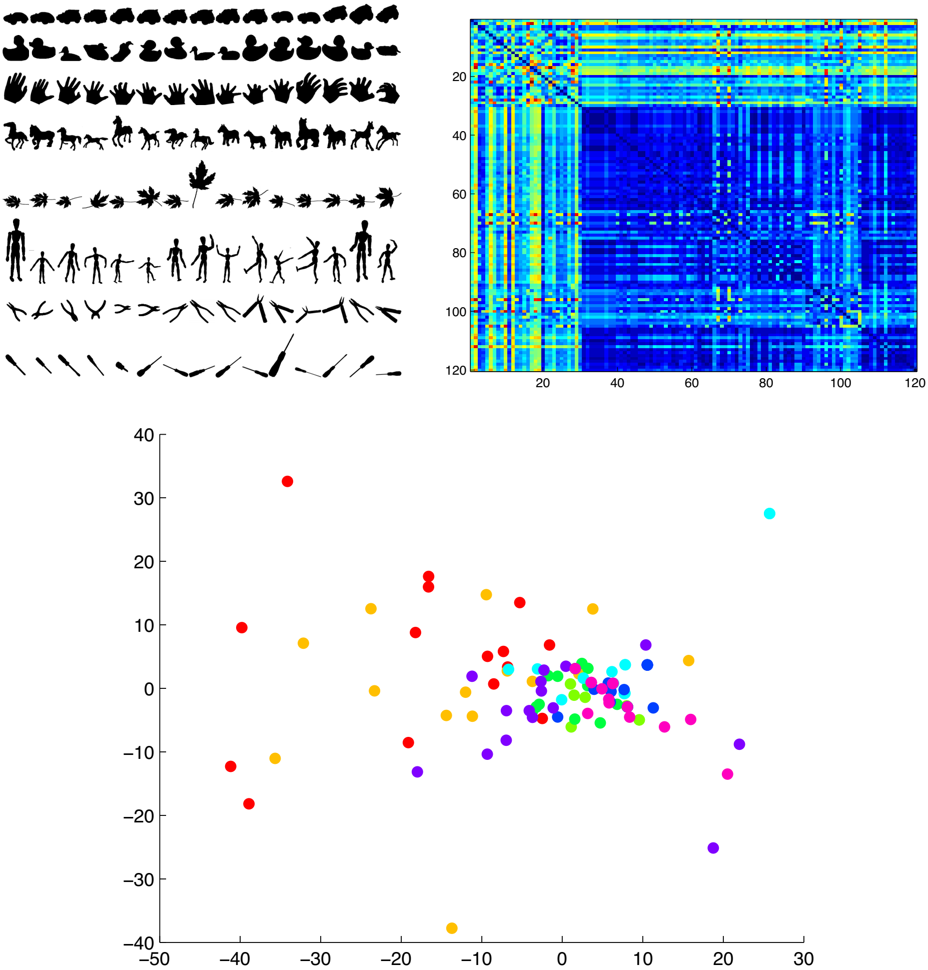


Fig. 7. Top row: Left, shape database; right, edit distance matrix. Bottom row: Multidimensional Scaling of the edit distances. As we can see, the class structure is not very clear and there is a considerable overlap between different classes.

4.3 Shock Graphs

Finally, we experimented using shock graphs, a skeletal-based representation of the differential structure of the boundary of a 2D shape. We extracted graphs from a database composed of 120 shapes divided into 8 classes of 15 shapes each. Each graph has a node attribute that reflects the size of the boundary feature generating the corresponding skeletal segment. Figure 7 shows the shape database, the edit distances matrix between the shock graphs and the corresponding MDS. As we can see, the class structure is not very clear, and there is a considerable overlap between different classes. This is reflected in the average accuracy of the kernels, which is the

Table 1. Classification accuracy (± standard error) on attributed graph datasets. QJSD is the proposed kernel, SP is the shortest-path kernel of Borgwardt and Kriegel [11], RW is the random walk kernel of Gartner et al. [10], while G_3 denotes the graphlet kernel computed using all graphlets of size 3 described in Shervashidze et al. [12]. The subscript w identifies the kernels which make use of the attributes information. The best performing kernel for each dataset is highlighted in bold.

Kernel	Synth	Shock	COIL
QJSD$_w$	95.87 ± 0.14	$\mathbf{66.65 \pm 0.22}$	$\mathbf{95.56 \pm 0.20}$
QJSD	84.57 ± 0.25	53.97 ± 0.19	84.05 ± 0.22
SP$_w$	$\mathbf{96.36 \pm 0.12}$	65.05 ± 0.25	94.40 ± 0.14
SP	91.13 ± 0.15	52.62 ± 0.32	85.25 ± 0.21
RW$_w$	92.97 ± 0.18	53.26 ± 0.29	90.78 ± 0.26
RW	80.23 ± 0.30	26.11 ± 0.32	78.60 ± 0.25
G3$_w$	88.75 ± 0.25	41.18 ± 0.27	89.25 ± 0.21
G3	85.60 ± 0.25	38.85 ± 0.32	84.20 ± 0.22

lowest among the three datasets, as Table 1 shows. However, the proposed kernel still outperforms or is competitive with the others.

5 Conclusions and Future Work

In this paper, we have introduced a novel similarity measure for attributed graphs based on the time evolution of a continuous-time quantum walk. More precisely, given a pair of graphs we computed the quantum Jensen-Shannon divergence between the evolution of two quantum walks on a suitably defined union of the original graphs. With the quantum Jensen-Shannon divergence to hand, we then established our similarity measure. Finally, we introduced a novel kernel on attributed graphs based on the proposed similarity measure. We performed an extensive experimental evaluation both on synthetic and real-world datasets, and we demonstrated the effectiveness of the proposed approach.

However, in this paper we limited our definition of the kernel to the case where the time parameter T is taken to the limit, i.e., $T \to \infty$. Future work will focus on studying the role of the time parameter more in depth, and it will try to develop a heuristic to establish the optimal time T in terms of classification accuracy.

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

References

1. Siddiqi, K., Shokoufandeh, A., Dickinson, S., Zucker, S.: Shock graphs and shape matching. International Journal of Computer Vision 35, 13–32 (1999)
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. Kalapala, V., Sanwalani, V., Moore, C.: The structure of the united states road network. University of New Mexico (2003) (preprint)
5. Shapiro, L., Haralick, R.: Structural descriptions and inexact matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, 504–519 (1981)
6. Barrow, H.G., Burstall, R.M.: Subgraph isomorphism, matching relational structures and maximal cliques. Inf. Process. Lett. 4, 83–84 (1976)
7. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph. Pattern Recognition Letters 19, 255–259 (1998)
8. Hidović, D., Pelillo, M.: Metrics for attributed graphs based on the maximal similarity common subgraph. International Journal of Pattern Recognition and Artificial Intelligence 18(3), 299–313 (2004)
9. Smola, A., Schölkopf, B.: Learning with kernels. Citeseer (1998)
10. Gärtner, T., Flach, P., 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)
11. Borgwardt, K., Kriegel, H.: Shortest-path kernels on graphs. In: Fifth IEEE International Conference on Data Mining, p. 8. IEEE (2005)
12. 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. Society for Artificial Intelligence and Statistics (2009)
13. Haussler, D.: Convolution kernels on discrete structures. Technical report, UC Santa Cruz (1999)
14. Kempe, J.: Quantum random walks: an introductory overview. Contemporary Physics 44, 307–327 (2003)
15. Emms, D., Wilson, R.C., Hancock, E.R.: Graph embedding using quantum commute times. In: Escolano, F., Vento, M. (eds.) GbRPR. LNCS, vol. 4538, pp. 371–382. Springer, Heidelberg (2007)
16. Rossi, L., Torsello, A., Hancock, E.R.: Approximate axial symmetries from continuous time quantum walks. In: Gimel'farb, G., Hancock, E.R., Imiya, A., Kuijper, A., Kudo, M., Omachi, S., Windeatt, T., Yamada, K. (eds.) SSPR&SPR 2012. LNCS, vol. 7626, pp. 144–152. Springer, Heidelberg (2012)
17. Lamberti, P., Majtey, A., Borras, A., Casas, M., Plastino, A.: Metric character of the quantum Jensen-Shannon divergence. Physical Review A 77, 052311 (2008)
18. Nielsen, M., Chuang, I.: Quantum computation and quantum information. Cambridge university press (2010)
19. Torsello, A., Rossi, L.: Supervised learning of graph structure. In: Pelillo, M., Hancock, E.R. (eds.) SIMBAD 2011. LNCS, vol. 7005, pp. 117–132. Springer, Heidelberg (2011)
20. Wish, M., Carroll, J.D.: 14 Multidimensional scaling and its applications. Handbook of Statistics 2, 317–345 (1982)
21. Briët, J., Harremoës, P.: Properties of classical and quantum jensen-shannon divergence. Physical Review A 79, 052311 (2009)
22. Nayar, S., Nene, S., Murase, H.: Columbia object image library (coil 100). Technical report, Tech. Report No. CUCS-006-96. Department of Comp. Science, Columbia University (1996)

Entropy and Heterogeneity Measures for Directed Graphs

Cheng Ye[1], Richard C. Wilson[1], César H. Comin[2], Luciano da F. Costa[2], and Edwin R. Hancock[1,⋆]

[1] Department of Computer Science, University of York,
York, YO10 5GH, UK
{cy666,richard.wilson,edwin.hancock}@york.ac.uk

[2] Institute of Physics at São Carlos, University of São Paulo,
P.O. Box 369, São Carlos, São Paulo, 13560-970, Brazil
{appdnails,ldfcosta}@gmail.com

Abstract. In this paper, we aim to develop novel methods for measuring the structural complexity for directed graphs. Although there are many existing alternative measures for quantifying the structural properties of undirected graphs, there are relatively few corresponding measures for directed graphs. To fill this gap in the literature, we explore a number of alternative techniques that are applicable to directed graphs. We commence by using Chung's generalisation of the Laplacian of a directed graph to extend the computation of von Neumann entropy from undirected to directed graphs. We provide a simplified form of the entropy which can be expressed in terms of simple vertex in-degree and out-degree statistics. Moreover, we find approximate forms of the von Neumann entropy that apply to both weakly and strongly directed graphs, and that can be used to characterize network structure. Next we explore how to extend Estrada's heterogeneity index from undirected to directed graphs. Our measure is motivated by the simplified von Neumann entropy, and involves measuring the heterogeneity of differences in in-degrees and out-degrees. Finally, we perform an analysis which reveals a novel linear relationship between heterogeneity and resistance distance (commute time) statistics for undirected graphs. This means that the larger the difference between the average commute time and shortest return path length between pairs of vertices, the greater the heterogeneity index. Based on this observation together with the definition of commute time on a directed graph, we define an analogous heterogeneity measure for directed graphs. We illustrate the usefulness of the measures defined in this paper for datasets describing Erdos-Renyi, 'small-world', 'scale-free' graphs, Protein-Protein Interaction (PPI) networks and evolving networks.

Keywords: directed graph, structural complexity, von Neumann entropy, heterogeneity index.

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

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 219–234, 2013.

1 Introduction

Recently there has been considerable interest in analyzing the properties of complex networks since they play a significant role in modelling large-scale systems in biology, physics and the social sciences. In fact, complex networks provide convenient models for complex systems. However, to render such models tractable, it is essential to have to hand methods for characterizing their salient properties. As Costa and Rodrigues [7] stated, complex networks are graphs whose connectivity properties deviate from those of regular graphs, which can be defined as a process of being 'simple', and the complexity of a network can be understood as the complement of simplicity. Structural complexity is therefore perhaps the most important property of a complex network. In order to analyze the features of a complex network it is imperative that computationally efficient measures are to hand that can be used to represent and quantify the structural complexity.

In this context graph theoretic methods are often used since they provide effective tools for characterizing network structure together with the intrinsic complexity. This approach has lead to the design of several practical methods for characterizing the global and local structure of undirected networks. However, there is relatively little work aimed at characterizing directed network structure. One of the reasons for this is that the graph theory underpinning directed networks is less developed than that for undirected networks.

The aim in this paper is to explore whether a number of different characterizations developed for undirected graphs can be extended to the domain of directed graphs, using some recent results from spectral graph theory.

1.1 Related Literature

Recently, Amancio et al. [1] have shown that labyrinths can be modelled as complex networks and studied in terms of the concept of absorption time, defined as the time it takes for a random walk from an internal node to an output node, to classify networks' metrics. Moreover, Estrada [10] has proposed an index that can be used to quantify the heterogeneity characteristics of undirected graphs. This index depends on vertex degree statistics and graph size. The lower bound of this quantity is zero, which occurs for a regular graph (i.e. all the vertices have the same degree). The upper bound is equal to one, which is obtained for a star graph (i.e. there exists a central vertex and all other vertices connect and only connect to it).

Working in the domain of structural pattern recognition, Xiao et al. [19] have explored how the heat kernel trace can be used as a means to characterize the structural complexity of graphs. To do this, they first consider the zeta function associated with the Laplacian eigenvalues and use the derivative of zeta function at origin as a characterization for distinguishing different types of graphs. Ren et al. [15] have developed a novel method to characterize unweighted graphs by using the polynomial coefficients determined by the Ihara zeta function. To do this, they construct a pattern vector of Ihara coefficients, and successfully use this to cluster unweighted graphs. Furthermore, they extend their work by applying

Ihara coefficients from unweighted graphs to edge-weighted graphs, which is achieved by establishing the Perron-Frobenius operator with the assistance of a reduced Bartholdi zeta function.

Escolano et al. [8] have used the concept of thermodynamic depth to measure the complexity of networks. They first define the polytopal complexity of a graph and then introduce a phase-transition principle which links this complexity to the heat flow, and thus obtain a complexity measure referred as flow complexity. Recently, Han et al. [12] have developed simplified expressions of von Neumann entropy on undirected graphs. To do this, they replace the Shannon entropy by its quadratic counterpart, investigate how to simplify and approximate the calculation of von Neumann entropy. They also explore the relationship among the heterogeneity index, commute time and the von Neumann entropy, and introduce a graph complexity measure based on thermodynamic depth.

The above provides a brief survey of recent work on the structural complexity of undirected graphs. However, in the real world, directed graphs are also common as many networks can be modelled with them. For example, the World Wide Web is a directed network in which vertices represent web pages while edges are the hyperlinks between pages.

Turning our attention to directed graphs, Riis [16] has extended the concept of entropy to directed graphs, using the definitions of guessing number and shortest index code. He shows that the entropy is the same as the guessing number and can be bounded by the graph size and shortest index code size. Berwanger et al. [4] have proposed a new parameter for the complexity of infinite directed graphs by measuring to what extent the cycles in graphs are intertwined. This index is defined according to the definitions of tree width, directed tree width and hypertree width and a similar 'robber-and-cops' game. Recently Escolano et al. [9] have extended the concept of heat diffusion thermodynamic depth for undirected networks to directed networks and thus obtain a measure to quantify the complexity of structural patterns encoded by directed graphs.

1.2 Paper Outline

One natural way of capturing the structure of directed networks is to use statistics that capture the balance of in-degree and out-degree at vertices. In this paper we commence from Passerini and Severini's work [13], which interprets the normalized Laplacian as a density matrix for an undirected graph, and this in turn allows the graph to be characterized in terms of the von Neumann entropy associated with the density matrix. We extend this work to directed graphs, using Chung's [6] definition of the normalized Laplacian on a directed graph. According to this definition, the directed normalized Laplacian is Hermitian, so Passerini and Severini's interpretation still holds for the domain of directed graphs. The von Neumann entropy is essentially the Shannon entropy associated with the normalized Laplacian eigenvalues. If we approximate the Shannon entropy by its quadratic counterpart, then the von Neumann entropy can be simplified. The resulting expression depends on the in-degree and out-degree of pairs of vertices connected by edges.

To simplify this resulting expression a step further, we consider graphs that are either weakly or strongly directed, i.e. those in which there are large or small proportions of bidirectional edges, and develop corresponding approximations of the von Neumann entropy.

Finally, we explore how Estrada's heterogeneity index can be extended from undirected to directed graphs. Our study of von Neumann entropy suggests a statistic determined by the in-degree and out-degree for nodes connected by a directed edge. We show that the resulting heterogeneity index is linked to the difference between the elements of the normalized adjacency matrix (as a local measure of connectivity) and the average commute time between nodes (or resistance distance) as a more global measure of connectivity structure.

The outline of this paper is as follows. In Sect.2, we develop the simplified forms of von Neumann entropy of directed graphs, and in Sect.3, we introduce the heterogeneity index and commute time on directed graphs and then investigate their correlation. In Sect.4, we analyze our theoretical result by undertaking experiment on network datasets and finally we conclude this paper with an evaluation of our contribution and suggestions for future work.

2 Von Neumann Entropy of Directed Graphs

In this section, we propose novel methods for characterizing the complexity of directed graphs. The first method is based on extending the definition of von Neumann entropy from undirected to directed graphs. To do this we commence from Chung's definition of the Laplacian for directed graphs. This leads to an expression for the von Neumann entropy in terms of the in-degree and out-degree statistics of vertices. We then provide approximations for the von Neumann entropy for both strongly directed graphs where there are few bidirectional edges and weakly directed graphs where there are few edges that are not bidirectional.

2.1 Laplacian of Directed Graphs

Suppose $G(V, E)$ is a directed graph with vertex set V and edge set $E \subseteq V \times V$, then the structure of this graph can be represented by a $|V| \times |V|$ adjacency matrix A as follows (where $|V|$ is the number of vertices in the graph)

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The in-degree and out-degree of vertex i are

$$d_i^{in} = \sum_{j=1}^{|V|} A_{ji}, \quad d_i^{out} = \sum_{j=1}^{|V|} A_{ij}. \tag{2}$$

With these ingredients, the transition matrix P for the directed graph G is defined as

$$P_{ij} = \begin{cases} \frac{A_{ij}}{d_i^{out}} & \text{if } (i,j) \in E \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

According to the Perron-Frobenius Theorem, for a strongly connected directed graph, the transition matrix P has a unique left eigenvector ϕ with $\phi(i) > 0$, $\forall i$ which satisfies $\phi P = \rho \phi$ where ρ denotes the eigenvalue of P. The theorem also implies that if P is aperiodic, the eigenvalues of P have absolute values smaller than the leading eigenvalue $\rho = 1$. Thus any random walk on a directed graph will converge to a unique stationary distribution if the graph satisfies the properties of strong connection and aperiodicity. We normalize ϕ s.t. $\sum_{i=1}^{|V|} \phi(i) = 1$, this normalized vector corresponds to the unique stationary distribution. Therefore, the probability of a random walk is at vertex i is the sum of all incoming probabilities of vertices j satisfying $(j,i) \in E$, i.e. $\phi(i) = \sum_{j,(j,i)\in E} \phi(j) P_{ji}$, then we can obtain the following approximate equation

$$\frac{\phi(i)}{\phi(j)} \approx \frac{d_i^{in}}{d_j^{in}}. \tag{4}$$

As stated in Chung [6], if we let $\Phi = diag(\phi(1), \phi(2), ...)$, then the normalized Laplacian matrix of a directed graph can be defined as

$$\tilde{L} = I - \frac{\Phi^{1/2} P \Phi^{-1/2} + \Phi^{-1/2} P^T \Phi^{1/2}}{2}. \tag{5}$$

Clearly, the normalized matrix is Hermitian matrix, i.e. $\tilde{L} = \tilde{L}^T$ where $\tilde{L}^T$ denotes the conjugated transpose of $\tilde{L}$.

2.2 Von Neumann Entropy of Undirected Graphs

Having defined the prerequisites, we now show how the concept of von Neumann entropy can be extended from undirected to directed graphs. Passerini and Severini [13] have argued that the normalized Laplacian can be interpreted as the density matrix of an undirected graph, and hence the associated von Neumann entropy of the graph is the Shannon entropy associated with the normalized Laplacian eigenvalues, i.e.

$$H_{VN}^U = -\sum_{i=1}^{|V|} \frac{\tilde{\lambda}_i}{|V|} \ln \frac{\tilde{\lambda}_i}{|V|} \tag{6}$$

where $\tilde{\lambda}_i$, $i = 1, ..., |V|$ are the eigenvalues of the normalized Laplacian matrix.

Commencing from their definition, Han et al. [12] have shown that for an undirected graph $G(V,E)$, the Shannon entropy $H_S^U = -\sum_{i=1}^{|V|} \frac{\tilde{\lambda}_i}{|V|} \ln \frac{\tilde{\lambda}_i}{|V|}$ can be approximated by the quadratic entropy $H_Q^U = \sum_{i=1}^{|V|} \frac{\tilde{\lambda}_i}{|V|}(1 - \frac{\tilde{\lambda}_i}{|V|})$. As a result the von Neumann entropy of undirected graphs can be approximated by

$$H^U_{VN} = \frac{Tr[\tilde{L}]}{|V|} - \frac{Tr[\tilde{L}^2]}{|V|^2}. \tag{7}$$

For undirected graphs, the traces appearing in the above expression can be approximated by degree statistics, with the result that

$$H^U_{VN} = 1 - \frac{1}{|V|} - \frac{1}{|V|^2} \sum_{(i,j)\in E} \frac{1}{d_i d_j}. \tag{8}$$

2.3 Von Neumann Entropy of Directed Graphs

To extend the analysis of Han et al. [12] to directed graphs, we commence from (7) and repeat the computation of traces for the case of a directed graph. This is not a straightforward task, and requires that we distinguish between the in-degree and out-degree of vertices. To commence, we turn to Chung's expression for the normalized Laplacian of directed graphs and write

$$\begin{aligned} Tr[\tilde{L}] &= Tr[I - \frac{\Phi^{1/2}P\Phi^{-1/2} + \Phi^{-1/2}P^T\Phi^{1/2}}{2}] \\ &= Tr[I] - \frac{1}{2}Tr[\Phi^{1/2}P\Phi^{-1/2}] - \frac{1}{2}Tr[\Phi^{-1/2}P^T\Phi^{1/2}]. \end{aligned} \tag{9}$$

Since the trace is invariant under cyclic permutations, i.e. $Tr[ABC] = Tr[BCA] = Tr[CAB]$, we have

$$\begin{aligned} Tr[\tilde{L}] &= Tr[I] - \frac{1}{2}Tr[P\Phi^{-1/2}\Phi^{1/2}] - \frac{1}{2}Tr[P^T\Phi^{1/2}\Phi^{-1/2}] \\ &= Tr[I] - \frac{1}{2}Tr[P] - \frac{1}{2}Tr[P^T]. \end{aligned} \tag{10}$$

The diagonal elements of the transition matrix P are all zeros, hence we obtain

$$Tr[\tilde{L}] = Tr[I] = |V|, \tag{11}$$

which is exactly the same as in the case of undirected graphs.

Next we turn our attention to $Tr[\tilde{L}^2]$,

$$\begin{aligned} Tr[\tilde{L}^2] &= Tr[I^2 - (\Phi^{1/2}P\Phi^{-1/2} + \Phi^{-1/2}P^T\Phi^{1/2}) + \\ &\quad \frac{1}{4}(\Phi^{1/2}P\Phi^{-1/2}\Phi^{1/2}P\Phi^{-1/2} + \Phi^{1/2}P\Phi^{-1/2}\Phi^{-1/2}P^T\Phi^{1/2} + \\ &\quad \Phi^{-1/2}P^T\Phi^{1/2}\Phi^{1/2}P\Phi^{-1/2} + \Phi^{-1/2}P^T\Phi^{1/2}\Phi^{-1/2}P^T\Phi^{1/2})] \\ &= Tr[I^2] - Tr[P] - Tr[P^T] + \frac{1}{4}(Tr[P^2] + Tr[P\Phi^{-1}P^T\Phi] + Tr[P^T\Phi P\Phi^{-1}] + Tr[P^{T^2}]) \\ &= |V| + \frac{1}{2}(Tr[P^2] + Tr[P\Phi^{-1}P^T\Phi]), \end{aligned} \tag{12}$$

which is different to the result obtained in the case of undirected graphs.

To continue the development we first divide the edge set E into two disjoint subsets E_1 and E_2, where $E_1 = \{(i,j)|(i,j) \in E \text{ and } (j,i) \notin E\}$, $E_2 = \{(i,j)|(i,j) \in E \text{ and } (j,i) \in E\}$ that satisfy the conditions $E_1 \bigcup E_2 = E$, $E_1 \bigcap E_2 = \emptyset$. Then according to the definition of the transition matrix, we find

$$Tr[P^2] = \sum_{i=1}^{|V|}\sum_{j=1}^{|V|} P_{ij}P_{ji} = \sum_{(i,j)\in E_2} \frac{1}{d_i^{out}d_j^{out}}. \tag{13}$$

Using the fact that $\Phi = diag(\phi(1), (2), ...)$ we have

$$Tr[P\Phi^{-1}P^T\Phi] = \sum_{i=1}^{|V|}\sum_{j=1}^{|V|} P_{ij}^2 \frac{\phi(i)}{\phi(j)} = \sum_{(i,j)\in E} \frac{\phi(i)}{\phi(j) d_i^{out^2}}. \tag{14}$$

Using (4), i.e. $\frac{\phi(i)}{\phi(j)} \approx \frac{d_i^{in}}{d_j^{in}}$, we can approximate the von Neumann entropy of a directed graph in terms of the in-degree and out-degree of the vertices as follows

$$H_{VN}^D = 1 - \frac{1}{|V|} - \frac{1}{2|V|^2}\left\{ \sum_{(i,j)\in E}\left(\frac{1}{d_i^{out}d_j^{out}} + \frac{d_i^{in}}{d_j^{in}d_i^{out^2}}\right) - \sum_{(i,j)\in E_1} \frac{1}{d_i^{out}d_j^{out}}\right\}, \tag{15}$$

or equivalently,

$$H_{VN}^D = 1 - \frac{1}{|V|} - \frac{1}{2|V|^2}\left\{ \sum_{(i,j)\in E} \frac{d_i^{in}}{d_j^{in}d_i^{out^2}} + \sum_{(i,j)\in E_2} \frac{1}{d_i^{out}d_j^{out}}\right\}. \tag{16}$$

We can simplify this expression a step further according to the relative sizes of the sets E_1 and E_2.

For weakly directed graphs, $|E_1| \ll |E_2|$, i.e. few of the edges are not bidirectional, and we can ignore the summation over E_1 in (15). Re-writing the remaining terms in curly braces in terms of a common denominator and then dividing numerator and denominator by $d_i^{out}d_j^{out}$ we obtain

$$H_{VN}^{WD} = 1 - \frac{1}{|V|} - \frac{1}{2|V|^2} \sum_{(i,j)\in E} \frac{\frac{d_i^{in}}{d_i^{out}} + \frac{d_j^{in}}{d_j^{out}}}{d_i^{out}d_j^{in}}. \tag{17}$$

The first term $1 - \frac{1}{|V|}$ tends to unity as the graph size becomes large and the remaining term is normalized by $2|V|^2$. In its second term above, the numerator is given in terms of the sum of the ratios of in-degree and out-degree at the two vertices. Since the directed edges cannot commence at a sink (a node of zero out-degree), the ratios do not become infinite. Replacing $d_i^{out^2}$ in the denominator by $d_i^{in}d_i^{out}$, we obtain the following expression that approximates the von Neumann entropy for weakly directed graphs

$$H_{VN}^{WD} = 1 - \frac{1}{|V|} - \frac{1}{2|V|^2} \sum_{(i,j)\in E} \left\{ \frac{1}{d_i^{out} d_j^{in}} + \frac{1}{d_i^{in} d_j^{out}} \right\}. \tag{18}$$

On the other hand, for strongly directed graphs, there are few bidirectional edges, i.e. $|E_2| \ll |E_1|$, and we can ignore the summation over E_2 in (16), giving the approximate entropy for strongly directed graphs

$$H_{VN}^{SD} = 1 - \frac{1}{|V|} - \frac{1}{2|V|^2} \sum_{(i,j)\in E} \left\{ \frac{1}{d_i^{out} d_j^{in}} \right\}. \tag{19}$$

Both the weakly and strongly directed forms of the von Neumann entropy (H_{VN}^{WD} and H_{VN}^{SD}) contain two terms. The first is the graph size while the second one depends on the in-degree and out-degree statistics of each pair of vertices linked by an edge. Moreover, the computational complexity of these expressions is quadratic in the graph size.

There are a number of points to note concerning the development above. First, the normalized Laplacian matrix of directed graphs denoted by $\tilde{L}$ in (5) satisfies Passerini and Severini's conditions [13] for the density matrix. Moreover, we have shown that $\tilde{L}$ is Hermitian matrix, so its eigenvalues are all real. Hence theoretically, the density matrix interpretation of Passerini and Severini [13] can be extended to directed graphs. Second, when the out-degree and in-degree are the same at a vertex, then the von Neumann entropy for directed and undirected graphs are identical.

3 Heterogeneity Index and Commute Time

In this section, we present an index which quantifies the heterogeneous properties of directed graphs. We introduce the definitions of hitting time and commute time and describe how to compute them, then explore that on undirected graphs, there exists a relationship between heterogeneity index and commute time, and show that the similar relationship also applies to the directed graphs.

3.1 Heterogeneity Index of Directed Graphs

Following Estrada [10], in order to compute a heterogeneity index for directed graphs, we first require a local index to measure the irregularity associated with a single edge $(i,j) \in E$. Estrada [10] uses the following quantity to measure the variation in degree

$$\sigma_{ij}^{U} = [f(d_i) - f(d_j)]^2 \tag{20}$$

where $f(d)$ is a function of the vertex degree. To extend this measure to directed graphs, we measure the difference in out-degrees and in-degrees and write

$$\sigma_{ij}^{D} = [f(d_i^{out}) - f(d_j^{in})]^2. \tag{21}$$

This local heterogeneity measure takes on a value zero when the out-degree of the starting vertex is the same as the in-degree of the end vertex. On the other hand, the index should become larger when the difference of both degrees increases, thus we can select $f(d) = d^{-1/2}$. The local heterogeneity index associated with the irregularity of the edge $(i, j) \in E$ in a directed graph is given by

$$\sigma_{ij}^{D} = \left(\frac{1}{\sqrt{d_i^{out}}} - \frac{1}{\sqrt{d_j^{in}}} \right)^2 . \tag{22}$$

To compute the global heterogeneity index of a directed graph we sum the local measure over all the edges in the graph to obtain

$$\rho^D(G) = \sum_{(i,j)\in E} \left\{ \frac{1}{\sqrt{d_i^{out}}} - \frac{1}{\sqrt{d_j^{in}}} \right\}^2 = \sum_{(i,j)\in E} \left\{ \frac{1}{d_i^{out}} + \frac{1}{d_j^{in}} \right\} - 2 \sum_{(i,j)\in E} \frac{1}{\sqrt{d_i^{out} d_j^{in}}} . \tag{23}$$

The heterogeneity index should take on a minimal value when the graph is regular, i.e. all the vertices have the same in-degree and out-degree. It is maximal when the graph is a star graph, i.e. there exists a central vertex such that all the other vertices connect and only connect to it. We calculate the lower and upper bounds of $\rho^D(G)$ according to these constraints. For a regular directed graph, suppose all the vertices have the same in-degree and out-degree d_0, then

$$\rho^D(G) = \sum_{(i,j)\in E} \left\{ \frac{1}{d_0} + \frac{1}{d_0} \right\} - 2 \sum_{(i,j)\in E} \frac{1}{d_0} = 0.$$

On the other hand, for a star graph, suppose that the central vertex has out-degree (in-degree) $|V| - 1$ and all the other vertices have in-degree (out-degree) 1. Then,

$$\rho^D(G) = \sum_{i=1}^{|V|} \left(\frac{1}{|V| - 1} + 1 \right) - 2 \sum_{i=1}^{|V|} \frac{1}{\sqrt{|V| - 1}} = \frac{|V|(|V| - 2\sqrt{|V| - 1})}{|V| - 1} \approx |V| - 2\sqrt{|V| - 1}.$$

We hence have the following lower and upper bounds for the heterogeneity index

$$0 \le \rho^D(G) = \sum_{(i,j)\in E} \left\{ \frac{1}{d_i^{out}} + \frac{1}{d_j^{in}} - \frac{2}{\sqrt{d_i^{out} d_j^{in}}} \right\} \le |V| - 2\sqrt{|V| - 1}. \tag{24}$$

Therefore we can define the normalized heterogeneity index of directed graphs as

$$\tilde{\rho}^D(G) = \frac{1}{|V| - 2\sqrt{|V| - 1}} \sum_{(i,j)\in E} \left\{ \frac{1}{d_i^{out}} + \frac{1}{d_j^{in}} - \frac{2}{\sqrt{d_i^{out} d_j^{in}}} \right\} \tag{25}$$

This index is zero for regular directed graphs, one for star graphs, i.e. $0 \le \tilde{\rho}^D(G) \le 1$.

3.2 Commute Time of Directed Graphs

To take our development one step further, we establish a relationship between the heterogeneity index and the commute time (or resistance distance) between nodes in a graph. To this end we commence by introducing the definitions of hitting time and commute time on directed graphs. The hitting time Q_{ij}^D is the expected number of steps for a random walk to reach vertex j for the first time, starting from vertex i. The commute time C_{ij}^D is the sum of Q_{ij}^D and Q_{ji}^D, i.e. $C_{ij}^D = Q_{ij}^D + Q_{ji}^D$, is the expected number of steps of a random walk starting at vertex i, visits j for the first time and then returns to vertex i.

Our expressions for both the hitting time and commute time are from Boley et al. [5]. We first introduce the definition of fundamental matrix Z which has elements

$$Z_{ij} = \sum_{t=0}^{\infty}(P_{ij}^t - \phi(j)), \quad 1 \leq i, j \leq |V| \tag{26}$$

or in matrix form,

$$Z = \sum_{t=0}^{\infty}(P^t - \mathbf{1}\phi) \tag{27}$$

where P is the transition matrix, $\mathbf{1} = (1, ..., 1)^T$ and ϕ is the stationary distribution.

The formulae for hitting time and commute time are

$$Q_{ij}^D = \frac{Z_{jj} - Z_{ij}}{\phi(j)}, \quad C_{ij}^D = Q_{ij}^D + Q_{ji}^D = \frac{Z_{jj} - Z_{ij}}{\phi(j)} + \frac{Z_{ii} - Z_{ji}}{\phi(i)}. \tag{28}$$

3.3 Relationship between Heterogeneity Index and Commute Time

According to Estrada [10], the normalized heterogeneity index of undirected graph has the following form

$$\tilde{\rho}^U(G) = \frac{1}{|V| - 2\sqrt{|V| - 1}} \sum_{(i,j)\in E} \left\{ \frac{1}{d_i} + \frac{1}{d_j} - \frac{2}{\sqrt{d_i d_j}} \right\}. \tag{29}$$

Recently, von Luxburg et al. [17] have shown that if the graph size is large enough, then the hitting time and commute time can be approximated by the resistance distance which takes on a simple form in terms of the vertex degree. In particular, $C_{ij}^U \approx vol\left(\frac{1}{d_i} + \frac{1}{d_j}\right)$ where vol is the volume of graph defined by $vol = \sum_{i=1}^{|V|} d_i$. As a result the first term appearing in the expression for Estrada's heterogeneity index can be expressed in terms of commute time.

To take this development one step further, we note that the normalized adjacency matrix for an undirected graph is given by $\tilde{A} = D^{-1/2}AD^{-1/2}$ where D is the diagonal matrix of vertex degrees. The normalized adjacency matrix has elements $\tilde{A}_{ij} = \frac{1}{\sqrt{d_i d_j}}$, if $(i, j) \in E$. As a result, in the heterogeneity index formula, if we make the substitutions $\frac{1}{d_i} + \frac{1}{d_j} = \frac{C^U_{ij}}{vol}$ and $\frac{1}{\sqrt{d_i d_j}} = \tilde{A}_{ij}$ we obtain the approximation

$$\tilde{\rho}^U(G) \approx \frac{1}{|V| - 2\sqrt{|V|-1}} \sum_{(i,j)\in E} \left\{ \frac{C^U_{ij}}{vol} - 2\tilde{A}_{ij} \right\}. \tag{30}$$

To extend this result to directed graphs, we note that

$$\sum_{(i,j)\in E} \left\{ \frac{1}{d^{out}_i} + \frac{1}{d^{in}_j} \right\} \approx \sum_{(i,j)\in E} \frac{C^D_{ij}}{vol} \tag{31}$$

where $vol = \sum_{i=1}^{|V|} d^{out}_i = \sum_{i=1}^{|V|} d^{in}_i$. If we denote by D_{out}, D_{in} the diagonal matrices of vertex out-degrees and in-degrees respectively, then the normalized adjacency matrix for a directed graph is $\tilde{A}^D = D_{out}^{-1/2} A D_{in}^{-1/2}$ with elements $\tilde{A^D}_{ij} = \frac{1}{\sqrt{d^{out}_i d^{in}_j}}$, if $(i,j) \in E$.

Hence, we obtain the following relationship between the heterogeneity index and commute time on directed graphs as

$$\tilde{\rho}^D(G) \approx \frac{1}{|V| - 2\sqrt{|V|-1}} \sum_{(i,j)\in E} \left\{ \frac{C^D_{ij}}{vol} - 2\tilde{A^D}_{ij} \right\}. \tag{32}$$

Thus we have shown that this relationship between heterogeneity index and commute time applies not only to undirected graphs but also to directed graphs.

Hence for both directed and undirected graphs, if the heterogeneity index is chosen in an appropriate way then there are two observations that can be drawn from this analysis. First, the heterogeneity index is proportional to the average commute time over pairs of nodes connected by an edge. Second, the heterogeneity index is greatest when the difference between the commute time and the twice the normalized adjacency matrix element is greatest. Hence, the heterogeneity index will be smallest for regular graphs and greatest for trees or star graphs.

4 Experiments and Evaluations

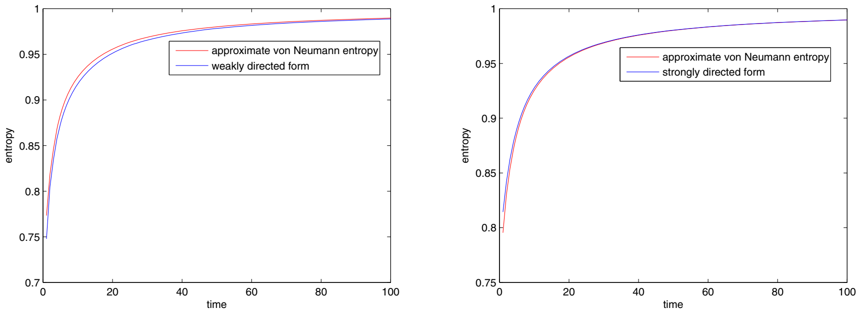
We have suggested several novel methods to measure the structural complexity of directed graphs. In this section, we aim to evaluate these methods on network data and give empirical analysis of their properties. First we examine both the weakly and strongly directed forms of von Neumann entropy, and compare their performance. Next, we explore whether our theoretically derived relationship between the heterogeneity index and commute time holds for both undirected and directed graphs.

4.1 The Datasets

Before undertaking our experiments, we first give a brief overview of the datasets used. The first dataset contains 150 undirected graphs in which the graph size varies from 50 to 100 nodes. Of this sample, 50 graphs are generated using the Erdos-Renyi model, which is considered as the most classical random graph model. An additional 50 graphs are generated according to the 'small-world' model, which was introduced by Watts and Strogatz [18]. The remaining 50 graphs are generated using the 'scale-free' model, which was developed by Barabasi and Albert [3]. The second dataset contains Protein-Protein Interaction (PPI) networks extracted from Franceschini et al. [11]. These graphs represent the interaction relationships between histidine kinase in different species of bacteria. The third dataset consists of 10 evolving directed networks. Each network commences from a fully connected network of size 5, and evolves gradually with new connections being established proportionally to the current dynamical activity of each vertex (preferential attachment). This dataset is generated using the model developed by Antiqueira et al. [2].

4.2 Entropy for Weakly and Strongly Directed Graphs

Equations (18) and (19) give the simplified forms of the von Neumann entropy for weakly and strongly directed graphs. We calculate them according to these two equations respectively and compare their behaviours with a reference entropy, i.e. the approximate von Neumann entropy generated using (15) (or equivalently, (16)), on the weakly and strongly directed networks in the third dataset.



(a) (b)

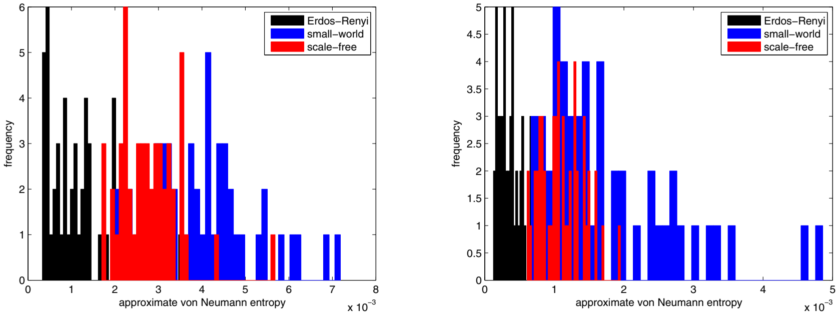
Fig. 1. Entropy for weakly & strongly directed graphs

We see in both Fig.1(a) and Fig.1(b), as the network evolves, both the simplified form and the reference entropy increase approximately monotonically until a plateaux value of unity is reached. Moreover, it is worth noting that the difference between these two quantities is negligible, thus we conclude that for weakly (strongly) directed graphs, the approximate von Neumann entropy and the simplified weakly (strongly) directed form we suggested are approximately equivalent.

We then explore whether the von Neumann entropy can be used to distinguish different types of graph. To this end we create directed versions of the Erdos-Renyi, ’small-world’ and ’scale-free’ graphs by deleting at random elements from the adjacency matrix. This has the effect of creating directed edges. In this analysis we consider the quantity

$$J = \left| H_{VN}^{D} - (1 - \frac{1}{|V|}) \right| = \frac{1}{2|V|^2} \left\{ \sum_{(i,j)\in E_2} \frac{1}{d_i^{out} d_j^{out}} + \sum_{(i,j)\in E} \frac{d_i^{in}}{d_j^{in} d_i^{out2}} \right\}$$

which removes some of the size dependence of the entropy.



(a)

(b)

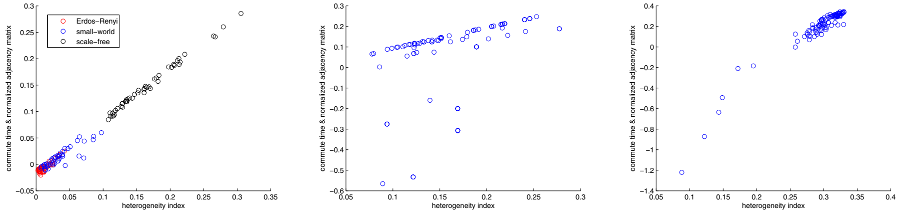
Fig. 2. Directed/Undirected graph characterization using von Neumann Entropy

In the left-hand of Fig.2, the plot shows superimposed histograms of J for each of the three types of directed graph. The main feature to note is that the Erdos-Renyi graphs are well separated from the ’small-world’ and ’scale-free’ graphs. Moreover, the ’scale-free’ and ’small-world’ networks although overlapped are reasonably well separated. The right-hand panel of Fig.2 repeats this analysis for the undirected versions of the three types of graph, using the original form of the von Neumann entropy suggested by Han et al. [12]. Here there is significantly more overlap, and the different types of network can not be easily separated, especially for the ’small-world’ and ’scale-free’ networks.

4.3 Heterogeneity Index and Commute Time

We have shown theoretically that the heterogeneity index has a linear dependance on the the commute time for both undirected and directed graphs. In this subsection we aim to confirm these results empirically. In Fig.3 we plot the heterogeneity index versus commute time for different types of graphs. Here the commute time of undirected graphs is calculated precisely using the graph spectral formula used by Qiu and Hancock [14]. Figure 3(a) shows the result for the Erdos-Renyi, 'small-world' and 'scale-free' graphs (shown in different colours). All three types of graphs follow a linear trend (i.e. they satisfy our theoretical prediction), but populate different parts of the 'heterogeneity-commute time' space. The second plot is for the protein-protein interaction networks. Although there are some outliers, most of the data falls in a linear regression curve. In fact, these outliers represent the graphs with particularly small graph size (e.g. 6 or 8), which is too small compared with others, thus these graphs do not perform the similar relation as other graphs do. Then we turn our attention to Fig.3(c), which is the plot of heterogeneity index versus average commute time for the directed graphs in the evolving sequence. The commute time here is computed according to (28). For the tightly clustered points in the upper right-hand corner of the plot, there is again a clear linear relationship, which confirms our theoretical prediction in (32).

Finally we explore the performance of directed graph characterization using the heterogeneity index. The histogram of the directed graph heterogeneity index is shown in Fig.4. In the histogram the 'scale-free' graphs are perfectly



(a) (b) (c)

Fig. 3. Relationship between Heterogeneity Index and Commute Time on undirected/directed graphs

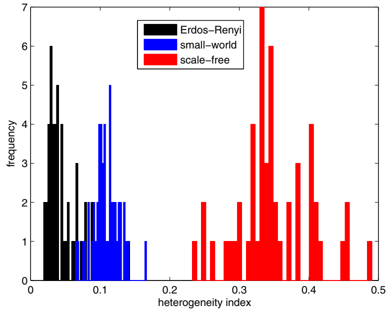


Fig. 4. Directed graph characterization using Heterogeneity Index

separated from the Erdos-Renyi and 'small-world' graphs. The result is not unexpected since for 'scale-free' graphs, the difference in the vertex in-degrees and out-degrees is particularly large, and the heterogeneity index of such graphs is greater than that for other types of graphs.

5 Conclusion

In this paper, motivated by the aim of developing novel and effective methods for quantifying the structural complexity of directed graphs, first we have developed approximations of the von Neumann entropy for both strongly and weakly directed graphs. They both depend on the vertex in-degree and out-degree statistics. Our approximations are based on using Chung's definition of normalized Laplacian matrix of directed graphs and simplifying the calculation via replacing the Shannon entropy by the quadratic entropy. Next, following the idea of developing the heterogeneity index for undirected graphs proposed by Estrada [10], we construct a similar measure which quantifies the heterogeneous properties of directed graphs. Moreover, concerning the commute time (or resistance distance), we have found that on an undirected graph, the heterogeneity index has a particular relation with it. Extending this correlation to directed graphs, we have discovered that they also exhibit a similar behaviour, which shows that the heterogeneity index can be approximated by the commute time and the normalized adjacency matrix. Then, in order to evaluate these methods and analyze their properties, we have undertaken some experiments on both undirected and directed network data and the experimental outcomes have demonstrated the effectiveness of our methods. In the future, our work can be extended by introducing more approaches to improving the measures we proposed in this paper for representing the structural complexity for directed graphs, and developing more novel indices which can reflect a directed graph's structure based on the entropy and heterogeneity index.

Acknowledgements. The authors are grateful to FAPESP (12/50986-7) and the University of York for financial support.

References

1. Amancio, D.R., Oliveira Jr., O.N., Costa, L.da F.: On the Concepts of Complex Networks to Quantify the Difficulty in Finding the Way Out of Labyrinths. Physica A 390, 4673–4683 (2011)
2. Antiqueira, L., Rodrigues, F.A., Costa, L.da F.: Modeling Connectivity in Terms of Network Activity. Journal of Statistical Mechanics: Theory and Experiment 0905.4706 (2009)
3. Barabasi, A.L., Albert, R.: Emergence of Scaling in Random Networks. Science 286, 509–512 (1999)
4. Berwanger, D., Gradel, E., Kaiser, L., Rabinovich, R.: Entanglement and the Complexity of Directed Graphs. Theoretical Computer Science 463, 2–25 (2012)

5. Boley, D., Ranjan, G., Zhang, Z.: Commute Times for a Directed Graph Using an Asymmetric Laplacian. Linear Algebra and Its Applications 435, 224–242 (2011)
6. Chung, F.: Laplacians and the Cheeger Inequailty for Directed Graphs. Annals of Combinatorics 9, 1–19 (2005)
7. Costa, L.da F., Rodrigues, F.A.: Seeking for Simplicity in Complex Networks. Europhysics Letters 85, 48001 (2009)
8. Escolano, F., Hancock, E.R., Lozano, M.A.: Heat Diffusion: Thermodynamic Depth Complexity of Networks. Physical Review E 85, 036206 (2012)
9. 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)
10. Estrada, E.: Quantifying Network Heterogeneity. Physical Review E 82, 066102 (2010)
11. Franceschini, A., Szklarczyk, D., Frankild, S., Kuhn, M., Simonovic, M., Roth, A., Lin, J., Minguez, P., Bork, P., von Mering, C., Jensen, L.J.: STRING v9.1: protein-protein interaction networks, with increased coverage and integration. Nucleic Acids Res. 41, D808–D815 (2013)
12. Han, L., Escolano, F., Hancock, E.R., Wilson, R.C.: Graph Characterizations from Von Neumann Entropy. Pattern Recognition Letters 33, 1958–1967 (2012)
13. Passerini, F., Severini, S.: The Von Neumann Entropy of Networks. International Journal of Agent Technologies and Systems, 58–67 (2008)
14. Qiu, H., Hancock, E.R.: Clustering and Embedding Using Commute Times. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 1873–1890 (2007)
15. Ren, P., Wilson, R.C., Hancock, E.R.: Graph Characterization via Ihara Coefficients. IEEE Transactions on Neural Networks 22, 233–245 (2011)
16. Riis, S.: Graph Entropy, Network Coding and Guessing Games. The Computing Research Repository 0711.4175 (2007)
17. von Luxburg, U., Radl, A., Hein, M.: Hitting and Commute Times in Large Graphs are often Misleading. Data Structures and Algorithms 1003.1266 (2010)
18. Watts, D.J., Strogatz, S.H.: Collective Dynamics of 'Small-World' Networks. Nature 393, 440–442 (1998)
19. Xiao, B., Hancock, E.R., Wilson, R.C.: Graph Characteristics from the Heat Kernel Trace. Pattern Recognition 42, 2589–2606 (2009)

Fast Learning of Gamma Mixture Models with k-MLE

Olivier Schwander[1] and Frank Nielsen[2]

[1] École Polytechnique, Palaiseau, France
[2] Sony Computer Science Laboratories Inc, Tokyo, Japan

Abstract. We introduce a novel algorithm to learn mixtures of Gamma distributions. This is an extension of the k-Maximum Likelihood Estimator algorithm for mixtures of exponential families. Although Gamma distributions are exponential families, we cannot rely directly on the exponential families tools due to the lack of closed-form formula and the cost of numerical approximation: our method uses Gamma distributions with a fixed rate parameter and a special step to choose this parameter is added in the algorithm. Since it converges locally and is computationally faster than an Expectation-Maximization method for Gamma mixture models, our method can be used beneficially as a drop-in replacement in any application using this kind of statistical models.

1 Introduction and Prior Work

Statistical mixtures are among the most used tools in many applications which require to model experimental data with probability distributions. Such a mixture $m(x)$ is a weighted sum of components which are themselves probability distributions (usually the same kind of distribution is shared by all the components):

$$m(x) = \sum_{i=1}^{k} \omega_i p(x; \theta_i) \tag{1}$$

The big challenge here is to learn the parameter vectors ω and θ and the number of components k (we limit us to the case of finite mixtures but some algorithms may consider mixtures with an infinite number of components [1]). One of the most famous algorithms to learn the parameters ω and θ is the Expectation-Maximization (EM) algorithm [2].

We address here the problem of learning mixtures of Gamma distributions (see Fig. 1). Although not as common as Gaussian mixture models, Gamma mixtures are of interest in many applications as various as bioinformatics [3], communication networks modeling [4] or health services analysis [5] and a lot of work has been devoted to these mixtures.

Our new algorithm is an extension of the$k$-Maximum Likelihood Estimator (k-MLE) algorithm by Nielsen [6]. It relies on the same principle which was already used for mixtures of generalized Gaussians [7]. Our contribution is to provide

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 235–249, 2013.

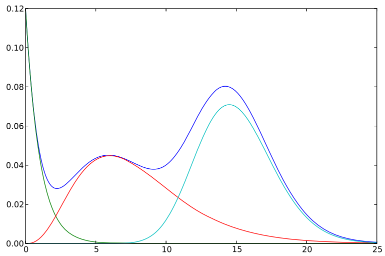


Fig. 1. A mixture of Gamma distributions with 3 components: $\omega_1 = 0.12, \alpha_1 = 1, \beta_1 = 1$; $\omega_2 = 0.4, \alpha_2 = 4, \beta_2 = 2$; $\omega_3 = 0.48, \alpha_3 = 30, \beta_3 = 0.5$

a new algorithm for Gamma mixtures which is faster than methods based on Expectation-Maximization.

Since the studied method relies on the exponential families framework, the necessary background about exponential families is recalled and we show that Gamma distributions are members of the exponential families. After a description of two algorithms designed to learn mixtures of exponential families, Bregman Soft Clustering, which relies on EM and k-MLE, we explain why they are not well suited for the particular case of Gamma mixtures. In the following section we present our *extension* of k-MLE which allows to efficiently learn mixtures of Gamma distributions. In the last section we evaluate the effectiveness of our proposed algorithm both in terms of computational cost and in terms of quality of the models.

2 Exponential Families and Their Parametrizations

2.1 Definition

Exponential families are a widespread class of distributions and many commonly used distributions belong to this class (with the notable exception of the uniform distribution): for example Gaussian, Beta, Gamma, Rayleigh, Von Mises are all members of this class ([8] provides a vast list of exponential families with their decomposition). An exponential family is a set of probability mass or probability density functions which admits the following canonical decomposition:

$$p(x;\theta) = \exp(\langle t(x), \theta\rangle - F(\theta) + k(x)) \tag{2}$$

with

- $t(x)$ the sufficient statistic,
- θ the natural parameters,
- $\langle\cdot,\cdot\rangle$ the inner product,
- F the log-normalizer,
- $k(x)$ the carrier measure.

The log-normalizer characterizes the exponential family and is derived from $\int p(x;\theta)dx = 1$ as:

$$F(\theta) = \log \int_x \exp(\langle t(x), \theta\rangle + k(x))\, dx \tag{3}$$

Many common distributions like the Beta, Gaussian or Dirichlet distributions are exponential families once a 1-to-1 mapping from the usual parameterization λ to the natural parameter θ is expressed [8]:

$$p(x;\lambda) = p(x;\theta(\lambda)). \tag{4}$$

Since this log-normalizer F is a strictly convex and differentiable function, it admits a dual representation, the convex conjugate F^*, by the Legendre-Fenchel transform:

$$F^\star(\eta) = \sup_\theta \{\langle \theta, \eta\rangle - F(\theta)\} \tag{5}$$

We get the maximum for $\theta = (\nabla F)^{-1}(\eta)$ and $F^\star$ can be computed with:

$$F^\star(\eta) = \langle \eta, (\nabla F)^{-1}(\eta)\rangle - F((\nabla F)^{-1}(\eta)) \tag{6}$$

Thus we deduce that the gradient of F and of its dual $F^\star$ are inversely reciprocal:

$$\nabla F = (\nabla F^\star)^{-1} \tag{7}$$

The duality between F and its Legendre transform $F^\star$ leads to a new parametrization for the exponential families, which is the dual of the natural parameters: the expectation parameters $\eta = \nabla F(\theta)$. The parameters η are called expectation parameters since $\eta = E\left[t(x)\right]$ [8].

In the general case, the dual $F^\star$ may be not known in closed-form and thus may require numerical approximation (which is time consuming and proned to various practical problems like the choice of the initialization for an iterative procedure).

2.2 Bregman Divergences

Bregman divergences are a family of divergences parametrized by the set of strictly convex and differentiable functions and is written as:

$$B_F(p\|q) = F(p) \;-\; F(q) \;-\; \langle p \;-\; q,\; \nabla F(q)\rangle \tag{8}$$

The function F is called the *generator* of the Bregman divergence.

The family of Bregman divergences generalizes many usual divergences, for example:

- the squared Euclidean distance, for $F(x) = x^2$,
- the Kullback-Leibler (KL) divergence, with the Shannon negative entropy $F(x) = \sum_{i=1}^{d} x_i \log x_i$ (also called Shannon information).

2.3 Bijection between Exponential Families and Bregman Divergences

Banerjee *et al.* [9] showed that Bregman divergences are in bijection with the exponential families through the generator F. For each exponential family with a log-normalizer F there is one and only one Bregman divergence whose generator is $F^\star$, the Legendre dual of F. We can rewrite the exponential family in terms of the corresponding Bregman divergence:

$$\begin{aligned} p(x;\theta) &= \exp(\langle t(x), \theta\rangle - F(\theta) + k(x)) && (9)\\ &= \exp(-B_{F^\star}(t(x)\|\eta) + F^\star(t(x)) + k(x)) && (10) \end{aligned}$$

where η is the expectation parameter of the family ($\eta = \nabla F(\theta)$).

This bijection allows in particular to compute the Kullback-Leibler divergence between two members of the same exponential family:

$$\begin{aligned} \mathrm{KL}\left(p(x,\theta_1); p(x,\theta_2)\right) &= \int_x p(x;\theta_1) \log \frac{p(x;\theta_1)}{p(x;\theta_2)}\, dx && (11)\\ &= B_F(\theta_2\|\theta_1) && (12) \end{aligned}$$

where F is the log-normalizer of the exponential family and the generator of the associated Bregman divergence.

Thus, computing the Kullback-Leibler divergence between two members of the same exponential family is equivalent to computing a Bregman divergence between their natural parameters (with swapped order).

2.4 Gamma Family Is an Exponential Family

The general case of the Gamma distribution is

$$p(x;\alpha,\beta) = \frac{\beta^\alpha x^{\alpha-1} \exp(-\beta x)}{\Gamma(\alpha)} \quad (13)$$

with $\alpha, \beta > 0$ and x is a positive real number.

The parameter α is called the **shape** parameter and β is called the **rate** parameter (or **inverse scale** parameter). It is common to find another parametrization which replace the rate parameter by the **scale** parameter $\theta = \frac{1}{\beta}$.

This distribution is an exponential family with the following parametrization:

Natural Parameters. $(\theta_1, \theta_2) = (-\beta, \alpha - 1)$

Sufficient Statistics. $t(x) = (x, \log x)$

Log Normalizer. $F(\theta_1, \theta_2) = (-(\theta_2 + 1)\log(-\theta_1) + \log \Gamma(\theta_2 + 1))$

Gradient Log Normalizer. $\nabla F(\theta_1, \theta_2) = \left(\frac{\theta_2+1}{-\theta_1}, -\log(-\theta_1) + \psi(\theta_2 + 1)\right)$

Dual Log Normalizer. $F^\star(\eta_1, \eta_2) = \left\langle (\nabla F)^{-1}(\eta_1, \eta_2), (\eta_1, \eta_2)\right\rangle - F\left((\nabla F)^{-1}(\eta_1, \eta_2)\right)$

Although the log-normalizer F and its gradient ∇F are known in closed-form, it is not the case for its dual $F^\star$ and for the gradient of the dual $\nabla F^\star = (\nabla F)^{-1}$. It thus requires numerical approximation, which is computationally costly.

3 Learning Mixtures of Exponential Families

3.1 Bregman Soft Clustering

The Bregman Soft Clustering for mixtures of exponential families has been introduced in [9]. It is a meta-algorithm which takes the considered family as an input of the algorithm and which does not require specific adaptation for each family, contrary to most of the previously proposed methods. As a variant of EM, it still relies on the usual two steps:

Expectation Step. The usual Expectation-Maximization algorithm gives us the following formulation for the posterior probabilities:

$$p(i|x_t, \eta) = \frac{\omega_i p(x_t; \eta_i)}{\sum_{j=1}^k \omega_j p(x_t; \eta_j)} \tag{14}$$

Using the bijection between exponential families, we can replace the probability density function of the exponential family by its expression using the associated Bregman divergence:

$$\begin{aligned} p(i|x_t, \eta) &= \frac{\omega_i \exp\left(-B_{F^\star}(t(x_i)\|\eta_i)\right) \exp k(x_t)}{\sum_{j=1}^k \omega_j \exp\left(-B_{F^\star}(t(x_t)\|\eta_j)\right) \exp k(x_t)} \quad (15) \\ &= \frac{\omega_i \exp\left(-B_{F^\star}(t(x_t)\|\eta_i)\right)}{\sum_{j=1}^k \omega_j \exp\left(-B_{F^\star}(t(x_t)\|\eta_j)\right)} \quad (16) \end{aligned}$$

Since $B_{F^\star}(p\|q) = F^\star(p) - F^\star(q) - \langle p-q, \nabla F^\star(q)\rangle$ we can expand the expression of the Bregman divergence in the previous expression:

$$p(i|x_t,\eta) = \frac{\omega_i \exp\left(-F^\star(t(x_t)) - F^\star(\eta_i) - \langle t(x_t) - \eta_i, \nabla F^\star(\eta_i)\rangle\right)}{\sum_{j=1}^k \omega_j \exp\left(-F^\star(t(x_t)) - F^\star(\eta_j) - \langle t(x_t) - \eta_j, \nabla F^\star(\eta_j)\rangle\right)} \tag{17}$$

$$= \frac{\omega_i \exp\left(F^\star(\eta_i) + \langle t(x_t) - \eta_i, \nabla F^\star(\eta_i)\rangle\right)}{\sum_{j=1}^k \omega_j \exp\left(F^\star(\eta_j) + \langle t(x_t) - \eta_j, \nabla F^\star(\eta_j)\rangle\right)} \tag{18}$$

Maximization Step. The maximization step is done with the maximum likelihood estimator for exponential families [9]. It can be computed as the average of the sufficient statistics on the observations:

$$\hat{\eta} = E\left[t(x)\right] = \frac{1}{n}\sum t(x_i) \tag{19}$$

Notice that we get an estimate which lives in the space of the expectation parameters. If one wants the associated natural parameter $\hat{\theta} = \nabla F^\star(\hat{\eta})$, the $\nabla F^\star$ function will be needed, either in closed-form or with a numerical approximation (which will be computationally costly). Note that the MLE is guaranteed to exist if and only if $\hat{\eta}$ falls inside the interior of the convex hull of the $t(x_i)$'s.

3.2 *k*-Maximum Likelihood Estimator

Assume we have a set $\mathcal{X} = \{x_1, \ldots, x_n\}$ of n observations which have been sampled from a finite mixture model with k components. The joint probability distribution of theses samples with the missing components z_i (indicating from which component each observation x_i comes from) is:

$$p(x_1, z_1, \ldots, x_n, z_n) = \prod_i p(z_i|\omega)p(x_i|z_i,\theta) \tag{20}$$

Since the variables z_i are not observed in practice, we marginalize these variable and we get:

$$p(x_1, \ldots, x_n|\omega,\theta) = \prod_i \sum_j p(z_i = j|\omega)p(x_i|z_i = j,\theta) \tag{21}$$

The straightforward way to optimize this distribution would be to test the k^n labels but this is not tractable in practice. Instead, Expectation-Maximization optimizes the following quantity, the expected log-likelihood:

$$\bar{l}(x_1, \ldots, x_n) = \frac{1}{n}\log p(x_1, \ldots, x_n) \tag{22}$$

$$= \frac{1}{n}\sum_i \log \sum_j p(z_i = j|\omega)p(x_i|z_i = j,\theta) \tag{23}$$

Contrary to this approach, the k-Maximum Likelihood Estimator maximizes the average complete log-likelihood:

$$\begin{aligned}
\bar{l}'(x_1, z_1, \ldots, x_n, z_n) &= \frac{1}{n} \log p(x_1, z_1, \ldots, x_n, z_n) &(24)\\
&= \frac{1}{n} \sum_i \log \prod_j \left((\omega_j p_F(x_i, \theta_j))^{\delta(z_i)} \right) &(25)\\
&= \frac{1}{n} \sum_i \sum_j \delta_j(z_i) \left(\log p_F(x_i, \theta_j) + \log \omega_j \right), &(26)
\end{aligned}$$

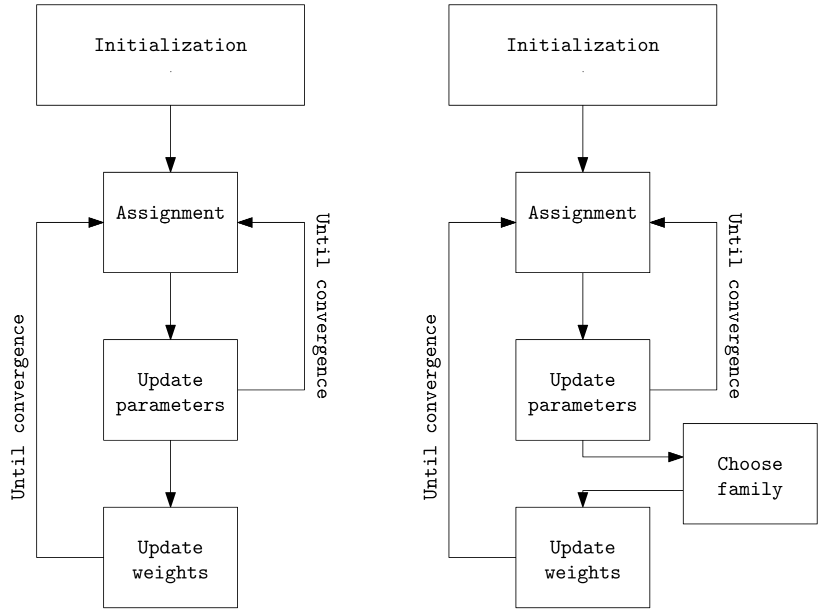
where $\delta_j(z_i) = 1$ if and only if z_i emanates from the j-th component.

Since p_F is an exponential family, we have:

$$\log p_F(x_i, \theta_j) = -B_{F^*}(t(x), \eta_j) + \underbrace{F^\star(t(x)) + k(x)}_{\text{does not depend on } \theta} \tag{27}$$

The terms which do not depend on θ are of no interest for the maximization problem and can be removed: We can then rewrite Eq. (26) to get the equivalent problem:

$$\arg\min \sum_i \sum_j \delta(z_i) \left(B_{F^*}(t(x), \eta_j) - \log \omega_j \right) \tag{28}$$



(a) Original k-MLE (b) Extended k-MLE

Fig. 2. Block diagram for the original k-MLE algorithm and its extension

As stated in [6] this problem can be solved for a fixed set of weights ω_i using the Bregman k-means algorithm with the Bregman divergence B_{F^*} (actually, any heuristic for k-means is convenient).

The weights can now be optimized by taking $\omega_i = \frac{|\mathcal{C}_i|}{n}$ (where $|C_i|$ is the number of observations put in the cluster $\mathcal{C}_i$ by the solution of the previous clustering problem). This step amounts to maximize the cross-entropy of the mixture [6].

The full algorithm can be summarized as follows (see Fig. 2(a) for a block diagram):

1. **Initialization** (choose seeds θ_i randomly or by using k-MLE ++[6]);
2. **Assignment** $z_i = \arg\max_j \log(\omega_j p_F(x_i|\theta_j))$;
3. **Update** of the η parameters $\eta_i = \frac{1}{n_j}\sum_{x\in\mathcal{C}_j} t(x_i)$;
   Goto step 2 until local convergence;
4. **Update** of the parameters ω_j;
   Goto step 2 until local convergence of the complete likelihood.

4 *k*-MLE for Gamma

4.1 Gamma with Fixed Rate Parameter

The algorithms described in the two previous sections needs frequent conversions between natural parameters θ and expectation parameters η. The bijection between the two parameter spaces uses the functions ∇F and $\nabla F^\star$. $\nabla F^\star$ is not known in closed-form for the Gamma distribution. Moreover, the evaluation of the Bregman divergence $B_{F^\star}$ is also needed, but the function $F^\star$ is also missing in closed-form. k-MLE may still be applicable to Gamma mixtures but the numerical approximations needed would dramatically reduce the speed of the algorithm, which is one of its main interests [10].

To avoid the computational difficulties for the functions which are not known in closed form, we introduce the Gamma distribution with a fixed rate parameter. The parameter β is not any more a member of the source parametrization and is instead a parameter of the exponential family $\{p_\beta(x;\alpha)\}$:

$$p_\beta(x;\alpha) = \frac{\beta^\alpha x^{\alpha-1}\exp(-\beta x)}{\Gamma(\alpha)} \tag{29}$$

This is still an exponential family with the following parametrization (a comprehensive list of formulas is given in Table 1):

Natural Parameters. $\theta = \alpha - 1$

Log Normalizer. $F(\theta) = -(\theta+1)\log(\beta) + \log\Gamma(\theta+1)$

Gradient Log Normalizer. $\nabla F(\theta) = -\log(\beta) + \psi(\theta+1)$

Dual Log Normalizer. $F^\star(\eta) = \langle \nabla F^\star(\eta), \eta \rangle - F\left(\nabla F^\star(\eta)\right)$

Gradient of the Dual Log Normalizer. $\nabla F^\star(\eta) = (\nabla F)^{-1}(\eta)$

The ∇F function can be inverted in closed-form with respect to the inverse digamma function ψ^{-1}, yielding:

$$(\nabla F)^{-1}(\eta) = \psi^{-1}(\eta + \log \beta) - 1 = \nabla F^\star(\eta) \tag{30}$$

We can now compute the $F^\star$ function by directly applying the Legendre transform to the log-normalizer F:

$$\begin{aligned} F^\star(\eta) &= \langle \nabla F^\star(\eta), \eta \rangle - F\left(\nabla F^\star(\eta)\right) && (31)\\ &= \eta\left(\psi^{-1}(\eta + \log \beta) - 1\right) + \psi^{-1}(\eta + \log \beta) \log \beta \\ &\quad - \log \Gamma\left(\psi^{-1}(\eta + \log \beta)\right) && (32) \end{aligned}$$

Strictly speaking, this is still not a closed-form but, contrary to the functions we get for the full Gamma distribution, the two missing functions Γ and ψ^{-1} can be computed efficiently: algorithms for the Γ function are well known [11] and ψ^{-1} is numerically well behaved and can be computed efficiently computed with a dichotomic search[1].

4.2 Maximum Likelihood Estimator

Results from exponential families [9] give an estimator for the expectation parameters of the fixed rate family:

$$\hat{\eta} = \frac{1}{n} \sum t(x_i) = \frac{1}{n} \sum \log(x_i) = - \log \hat{\alpha} + \psi(\beta) \tag{33}$$

Since the family is univariate (*i.e.*, one parameter α), the MLE always exist.

By derivation of the likelihood function, we get an estimator for the rate parameter β [4]:

$$\hat{\beta} = \frac{n \hat{\alpha}}{\sum x_i} \tag{34}$$

4.3 Learning Mixtures

The original k-MLE algorithm builds mixture models where all the components belong to the same exponential family. Although generic Gamma distributions

[1] See `http://hips.seas.harvard.edu/files/invpsi.m` for a working Matlab(R) implementation which can be easily translated in any language.

Table 1. Gamma distribution with fixed rate as an exponential family

PDF	$p_\beta(x;\alpha) = \frac{\beta^\alpha x^{\alpha-1}\exp(-\beta x)}{\Gamma(\alpha)}$
$\Lambda \to \Theta$	$\theta = \alpha - 1$
$\Theta \to \Lambda$	$\alpha = \theta + 1$
$\Lambda \to H$	$\eta = -\log\beta + \psi(\alpha)$
$H \to \Lambda$	$\alpha = \psi^{-1}(\eta + \log\beta)$
$\Theta \to H$	$\eta = \nabla F(\theta)$
$H \to \Theta$	$\theta = \nabla F^\star(\eta)$
Log normalizer	$F(\theta) = -(\theta+1)\log\beta + \log\Gamma(\theta+1)$
Gradient log normalizer	$\nabla F(\theta) = -\log\beta + \psi(\theta+1)$
Dual log normalizer	$F^\star(\eta) = \eta(\psi^{-1}(\eta+\log\beta)-1) + \psi^{-1}(\eta+\log\beta)\log\beta + \log\Gamma(\psi^{-1}(\eta+\log\beta))$
Gradient dual log normalizer	$\nabla F^\star(\eta) = \psi^{-1}(\eta+\log\beta) - 1$
Sufficient statistic	$t(x) = \log x$
Carrier measure	$k(x) = -\beta x$

are exponential families, Gamma distributions with fixed rate are not in the **same** exponential family if the rate parameter is not the same across components. In order to build a mixture with a different β parameter for each component, we will follow the approach introduced in [7] (for generalized Gaussian) which adds a supplementary step to the k-MLE procedure (see Fig. 2(b)): before updating the weights, the family of each component is chosen using a maximum likelihood estimator. In the Gamma case, it amounts to choosing the rate parameter of each component, using the MLE given in Eq. (34).

The new k-MLE algorithm for Gamma mixtures (k-MLE-Gamma) can be summarized as follows:

1. **Initialization** (random or using k-MLE ++[6]);
2. **Assignment** $z_i = \arg\max_j \log(\omega_j p_{F_j}(x_i|\theta_j))$;
3. **Update** of the η parameters $\eta_i = \frac{1}{n_j}\sum_{x\in\mathcal{C}_j}\log(x_i)$;
   Goto step 2 until stability (local convergence of the k-means);
4. **Update** of the parameters ω_j and β_j (for all j);
   Goto step 2 until local convergence of the complete likelihood.

Notice that this algorithm can be interpreted as a hard EM-type algorithm with two Maximization (M) steps.

4.4 Convergence to a Local Maximum

As the one proposed for generalized Gaussian, this algorithm converges to a local maximum of the complete log-likelihood. We want to minimize the same cost function as the original k-MLE algorithm, the complete log-likelihood of the mixture, with the slight difference that the log-normalizer is not shared among components but now depends on the values β_j and is now written F_j instead of F:

$$\begin{aligned}\bar{l}(x_1, z_1, ..., x_n, z_n|w, \theta) =& \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\delta_j(z_i)(\log p_{F_j}(x_i|\theta_j) + \log \omega_j) & (35)\\ =& \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\delta_j(z_i)\Big(- B_{F_j{}^*}(t(x_i), \eta_j) & \\ & + F_j{}^*(t(x_i)) + k_j(x_i) + \log \omega_j\Big) & (36)\end{aligned}$$

Let $\mathcal{C}_j$ be the set of the indices of the observations sampled from the j-th component. Maximizing the log-likelihood $\bar{l}$ is equivalent to minimizing the cost function $-\bar{l}$:

$$\bar{l}' = -\bar{l} = \frac{1}{n}\sum_{j=1}^{k}\sum_{i\in\mathcal{C}_j} U_j\left(x_i, \eta_j\right) \tag{37}$$

where

$$\begin{aligned}U_j(x_i, \eta_j) &= - \big(\log p_{F_j}(x_i|\theta_j) + \log \omega_j\big) & (38)\\ &= B_{F_j{}^*}(t(x_i) : \eta_j) - F_j{}^*(t(x_i)) & (39)\\ &\quad - k_j(x_i) - \log \omega_j &\end{aligned}$$

is the cost for the observation i to have been sampled from the component j. Notice this cost depends on j since each component has a different generator F_j and a different auxiliary carrier measure k_j.

This minimization problem can be solved with the Lloyd k-means algorithm [12] using the cost function U (which is not a distance nor a divergence and can even be negative). A proof of the convergence of the Lloyd algorithm for this cost function is given in [6,7].

After the execution of the Lloyd algorithm, the log-likelihood has been optimized for fixed ω_j and β_j. The final step is to update these two parameters using the proportion of samples in each cluster for the weights and the estimator for β (from Eq. (34)).

5 Expectation-Maximization for Gamma Mixtures

Almhana *et al.* [4] proposed a specific variant of Expectation-Maximization for Gamma mixtures. The E step is unchanged compared to the classical EM algorithm, the only changes are in the M step: a specific update step is used for the

α and β parameters. We will use this algorithm as a reference in the experiments presented in Section 6.

Maximization Step. Given the current estimate for the parameters ω, α and β, the new values can be computed with:

$$\omega_i^{(k+1)} = \frac{1}{n} \sum_{t=1}^{n} p(i|x_t, \theta^{(k)}) \tag{40}$$

$$\beta_i^{(k+1)} = \frac{\alpha_i^{(k)} \sum_{t=1}^{n} p(i|x_t, \theta^{(k)})}{\sum_{t=1}^{n} x_t p(i|x_t, \theta^{(k)})} \tag{41}$$

$$\alpha_i^{(k+1)} = \alpha_i^{(k)} + \frac{1}{k}\, G \tag{42}$$

with

$$G = \frac{1}{n} \sum_{t=1}^{n} \left(\log x_t + \log \beta_i^{(k)} - \psi(\alpha_i^{(k)}) \right) p(i|x_t, \theta^{(k)}) \tag{43}$$

6 Experiments

An implementation (in the C language) of the EM algorithm for Gamma mixtures and of the k-MLE for Gamma mixtures is available at `http://www.lix.polytechnique.fr/~schwander/libmef/`. In addition to the algorithms studied in the article, some other mixture models related algorithms are available (in particular: Bregman Soft Clustering, Bregman Hard Clustering, others variant of k-MLE). The following experiments use this implementation to evaluate our proposed algorithm.

6.1 On Synthetic Data

The first experiment evaluates the convergence of k-MLE and the convergence of EM on a synthetic example: 15000 observations are sampled from a given three-component Gamma mixture and the two evaluated methods are used to estimate Gamma mixture models with three components. We draw in Fig. 3 the log-likelihood of each mixture at each iteration of the two algorithms. Although the goal of k-MLE is to maximize the complete log-likelihood (Eq. (24)) and not the log-likelihood (Eq. (22) we see that both algorithms converge to a (local) maximum of the log-likelihood. Moreover k-MLE provides better results and converges way faster than EM.

6.2 On a Real Dataset

The second experiment describes experimental results on a real dataset which collects distances between atoms inside RNA molecules in order to predict the

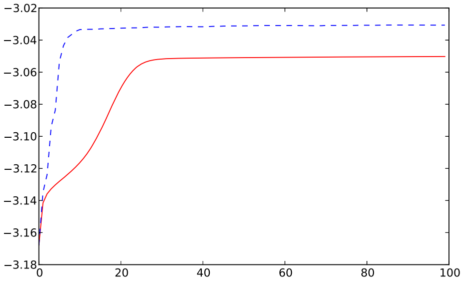
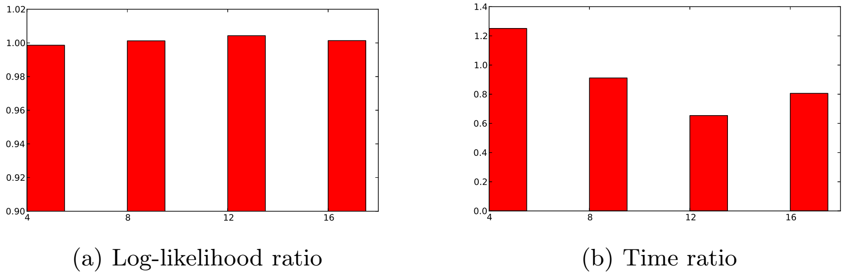


Fig. 3. Log-likelihood with respect to the number of components for k-MLE (dashed curve) and EM (plain curve). Higher curve (k-MLE model) means better model.

3D structure of these molecules. Gaussian mixture models were successfully used to model the density of these distances [13] [14] but since the observations are *intrinsically* positive a mixture model with a positive support (remember that Gaussian distribution is defined on $\mathbb{R}$ whereas the Gamma distribution is defined on $\mathbb{R}_+$) would be more statistically meaningful.



(a) Log-likelihood ratio

(b) Time ratio

Fig. 4. Log-likelihood and computation time ratios for k-MLE (right red bars) and EM (left blue bars) with respect to the number of components in the mixture. EM is our reference for comparison and thus has the score 1.

Fig. 4 presents results on this dataset, in terms of log-likelihood and computation time with respect to the number of components in the mixture (4, 8, 12 and 16 components). Since absolute value for likelihood and time are difficult to compare meaningfully, we plot the mean ratio between the values we got with k-MLE and the one got with EM (which is our reference for comparison and represented by 1 on the graphics). We observe that k-MLE for Gamma mixtures performs similarly (or even better) to EM for Gamma mixtures for the quality of the models and outperforms EM for the computation time (between 10% and 40%). The only case where k-MLE is worse than EM is for 4 components: k-MLE seems to be less robust when the number of components is not enough to model accurately the observations.

7 Conclusion

We presented a new algorithm for mixtures of Gamma distributions which is both fast and accurate. Accuracy is important since it means that the quality of the produced models (and thus the performances in the considered applications) will not decrease: our new algorithm could thus be considered as a drop-in replacement for other Gamma mixtures algorithms. The faster speed not only means that the computation time will decrease in applications where Gamma mixtures are already used but also that these mixtures will become of new interest in areas where the use of the Gamma distribution was theoretically interesting but not feasible in practice due to the high computation time. Moreover, this new extension of the k-Maximum Likelihood estimator shows the power and the genericity of the method which allows interesting perspectives for new and unexplored kinds of mixtures.

Acknowledgments. We thank the anonymous referees for their insightful comments and their careful proofreading.

References

1. Neal, R.M.: Markov chain sampling methods for Dirichlet process mixture models. Journal of Computational and Graphical Statistics 9(2), 249–265 (2000)
2. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 1–38 (1977)
3. Mayrose, I., Friedman, N., Pupko, T.: A Gamma mixture model better accounts for among site rate heterogeneity. Bioinformatics 21(suppl. 2) (2005)
4. Almhana, J., Liu, Z., Choulakian, V., McGorman, R.: A recursive algorithm for gamma mixture models. In: IEEE International Conference on Communications, ICC 2006, vol. 1, pp. 197–202 (June 2006)
5. Venturini, S., Dominici, F., Parmigiani, G.: Gamma shape mixtures for heavy-tailed distributions. The Annals of Applied Statistics 2(2), 756–776 (2008); Zentralblatt MATH identifier: 05591297; Mathematical Reviews number (MathSciNet): MR2524355
6. Nielsen, F.: k-MLE: A fast algorithm for learning statistical mixture models. CoRR (2012)
7. Schwander, O., Schutz, A.J., Nielsen, F., Berthoumieu, Y.: k-MLE for mixtures of generalized Gaussians. In: 2012 21st International Conference on Pattern Recognition (ICPR), pp. 2825–2828 (November 2012)
8. Nielsen, F., Garcia, V.: Statistical exponential families: A digest with flash cards. CoRR 09114863 (2009)
9. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with Bregman divergences. The Journal of Machine Learning Research 6, 1705–1749 (2005)
10. Nielsen, F.: k-MLE: A fast algorithm for learning statistical mixture models. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 869–872 (March 2012)

11. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical recipes: The art of scientific computing, 3rd edn. Cambridge University Press (2007)
12. Lloyd, S.P.: Least squares quantization in PCM. IEEE Transactions on Information Theory 28(2), 129–137 (1982)
13. Sim, A.Y., Schwander, O., Levitt, M., Bernauer, J.: Evaluating mixture models for building rna knowledge-based potentials. Journal of Bioinformatics and Computational Biology 10(02) (2012)
14. Bernauer, J., Huang, X., Sim, A.Y., Levitt, M.: Fully differentiable coarse-grained and all-atom knowledge-based potentials for rna structure evaluation. RNA 17(6), 1066–1075 (2011)

Exploiting Geometry in Counting Grids

Alessandro Perina[2], Manuele Bicego[1], Umberto Castellani[1],
and Vittorio Murino[1,3]

[1] Department of Computer Science, University of Verona, Italy
[2] Microsoft Research Redmond, Washington, USA
[3] Istituto Italiano di Tecnologia (IIT), Genova, Italy

Abstract. In this paper we exploit the use of known information about the geometry structure of a recently proposed generative model, namely Counting Grid (CG) [1] to improve the performance of classification accuracy. Once the generative model is trained, the geometric structure of the model introduces a natural spatial relations among the estimated latent variables. Such relation is generally ignored when standard maximum likelihood approach (or classical hybrid generative-discriminative approach) is employed for classification purpose. In this work, we propose to take into account the geometric relations of the generative model by proposing an ad hoc similarity measure for CG. In particular, the values relative to each point of the grid is spread around its neighborhood by using information coming from the CG training phase. The proposed approach is succesfully applied in two applicative scenarios: expression microarray classification and MRI brain classification. Experiments show a drastic improvement over standard schemes when our approach is employed.

Keywords: generative models, kernels, microarray, MRI.

1 Introduction

In pattern recognition some counting strategies are often introduced, especially when source data is not naturally lying on a vectorial space. A very popular example is the *Bag of Words* approach, where objects are represented as disorganized bags of basic components such as the words of a dictionary. This approach has been succesfully employed in very different applicative domains like computer vision for 2D image or 3D shape retrieval, in bioinformatics for microarray classification, or in medical domain for brain disease detection [2–8]. However, the Bag of Words (BoW) method has some disadvantage since in many situations it looses a lot of important information. For instance, BoW approach does not take into account words relations or co-occurences. To this aim, LDA or pLSA models have been succesfully proposed by showing how inter-relations among words, i.e., *topics* are crucial to improve object encoding [9, 10]. Recently, a new generative model has been proposed, namely Counting Grid (CG) [1] which goes beyond topic-based approach. Indeed, CG exploits not only words co-occurences but also topological relations among words. In particular, with CG an ordering procedure between BoWs is introduced by allowing BoWs to lie in

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 250–264, 2013.

an n-dimensional grid structure. Such approach has already shown its benefits on document retrieval, 2D scene classification, and microarray expression classification[1, 11]. In all these applications, the classification stage has been computed by standard maximum likelihood scheme, or by employing discriminative classifiers like Support Vector Machine (SVM) with generative kernels, nevertheless without taking into account the peculiar geometry of the model.

In this paper we propose to further exploit the advantage of CGs by studing an ad hoc (dis)similarity measure. We start from the observation that in the CG scenario, the classical classification scheme is based on the grid posterior of a given sample, which is treated as a vector and used for comparison. In such a way, spatial relation between values is lost. Nevertheless, due to the nature of the CG, in the training phase a BoW, or a *count*, is distributed on a local region around a specific point in the grid which is defined by an hidden variable. This leads to a spatial relation among grid points which can be used to improve the classification stage. The idea is to spread the posterior evaluated on a single grid-point around its neighborhood. In this fashion, when two samples are compared, an implicit cross-count evaluation is introduced by avoiding a fully grid alignment constraint. Experiments show that our new (dis-)simmilarity approach leads to a drastic improvement in comparison with standard methods.

The rest of the paper is organized as following. In Section 2 the background on Counting Grids is introduced. Section 3 describes the proposed (dis-)similarity measure for the proposed generative model. Section 4 reports experiments on two applicative domains, namely expression microarray classification and MRI brain disease classification. Finally, conclusions and future work are discussed in Section 5.

2 Background: Counting Grid Model

Data samples are often represented as an unordered bags of features, where each t-th observation is characterized by a vector called *count* vector $\{c_z^t\}$ which contains the number of occurrences of each feature z [12, 9]. For instance, a text document can be described by the number of words occurrences it contains (or an image with the number of occurrences of different visual features it contains). This choice is often motivated by the difficulty or computational efficiency of modeling the known structure of the data.

The counting grid model, recently introduced in [1], is a generative model that extends such representations. The models starts from a common choice in counting data modelling, which implies that the bag of features of a given sample is generated by a latent variable; in the counting grid model, nevertheless it is assumed that a spatial relation between latent variables exists, and can be learnt and used to improve the understanding of the models or to provide rich descriptors for classification. More explicitly, we can unformally say that the generative process of a given bag of features is based on a latent variable but also on some of its spatial neighbours. Formally, the basic counting grid $\pi_{\mathbf{i},z}$ is a

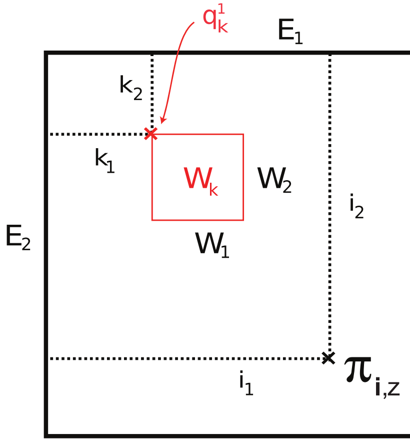


Fig. 1. An example of a counting grid geometry

set of normalized counts of features indexed by z on the 2-dimensional[1] discrete grid indexed by $\mathbf{i} = (i, j)$ where $i \in [1 \ldots E_1]$, $j \in [1 \ldots E_2]$ and $\mathbf{E} = [E_1, E_2]$ describes the extent of the counting grid. Since π is a grid of distributions, $\sum_z \pi_{\mathbf{i},z} = 1$ everywhere on the grid.

A given bag of features, represented by counts $\{c_z\}$ is assumed to follow a count distribution found in a patch of the counting grid. In particular, using a window of dimensions $\mathbf{W} = [W_1, W_2]$, each bag can be generated by first selecting a position $\mathbf{k}$ on the grid and then by placing the window in the grid such that $\mathbf{k}$ is its upper left corner. Then, all counts in this patch are averaged to form the histogram $h_{\mathbf{k},z} = \frac{1}{W_1 \cdot W_2} \sum_{\mathbf{i} \in W_{\mathbf{k}}} \pi_{\mathbf{i},z}$, and finally a set of features in the bag is generated. In other words, the position of the window $\mathbf{k}$ in the grid is a latent variable given which the probability of the bag of features $\{c_z\}$ is

$$p(\{c_z\}|\mathbf{k}) = \prod_z (h_{\mathbf{k},z})^{c_z} = \frac{1}{W_1 \cdot W_2} \prod_z (\sum_{\mathbf{i} \in W_{\mathbf{k}}} \pi_{\mathbf{i},z})^{c_z}$$

where with $W_{\mathbf{k}}$ we indicate the particular window placed at location $\mathbf{k}$ (see Figure 1).

We will refer to E and W respectively as the counting grid and the window size. We will also often refer to the ratio of the CG area and the window area $\kappa = \frac{E_1 \cdot E_2}{W_1 \cdot W_2}$, as the capacity of the model, which can be seen – using a topic models parallelism – as an equivalent number of topics (this is how many nonoverlapping windows can be fit onto the grid). Computing and maximizing the log likelihood of the data turns to be an intractable problem; therefore it is necessary to employ an iterative EM algorithm. The E step aligns all bags of features to grid windows, to match the bags' histograms, inferring the posterior probability $q^t_{\mathbf{k}}$, the probability that the sample t is generated from the position

[1] N-dimensional in general, here we focus on 2 dimensions.

$\mathbf{k}$, i.e., where each bag maps on the grid. This posterior can be computed as $q^t_{\mathbf{k}} \propto \exp \sum_z c^t_z \cdot \log h_{\mathbf{k},z}$. In the M-step the model parameter, i.e. the counting grid π, is re-estimated. To avoid severe local minima it is important to consider the Counting Grid as a torus, and perform all windowing operation accordingly. For details on the learning algorithm and on its efficiency see [1].

3 (Dis-)Similarity Measure for CG

Once the training phase is performed, the CG $\pi_{\mathbf{i},z}$ is available and can be used for classification purposes. Given a sample A, represented by counts $\{c^A_z\}$, its posterior $q^A_{\mathbf{k}}$ is computed. In general, the matrix $q^A_{\mathbf{k}}$ can be used in a maximum likelihood scheme or it can be fed in a discriminative classifier such as a Support Vector Machine, after its vectorization, representing a straightforward hybrid generative-discriminative classification approach. When using standard vector-based kernels (like linear kernel), the implicit assumption is that counts are well aligned, so that each count in one sample is only compared to corresponding count in another sample. Here, we exploit cross-count distances by observing that each point in the grid depends by its neighborhood which is defined by $\mathbf{W}$. Indeed, we propose to spread the values $q^A_{\mathbf{k}}$ around a neighborhood region defined by $W_{\mathbf{k}}$. Actually, by construction, the value in a given location $\mathbf{k}$ is computed by using all CG parameters belonging to the subwindow $\mathbf{W}$.

More in details, given two samples A and B, our similarity measure – which we call *Spreading Similarity Measure* is defined by:

$$SSM_S(A,B) = SM(q^A_{\mathbf{k}} * S_{\mathbf{W}}, q^B_{\mathbf{k}} * S_{\mathbf{W}}), \tag{1}$$

where $S_{\mathbf{W}}(\mathbf{x})$ is a box function, of dimension defined by the spreading window $\mathbf{W}$, defined as:

$$S_{\mathbf{W}}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \mathbf{W} \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

and $SM(\cdot,\cdot)$ is any (dis)-similarity measure. In our experiments we evaluate standard inner product[13], histogram intersection [14], and Jensen-Shannon distance [15]. Reasonably, we chose to set the size of the spreading windows as the size of the Counting Grid Window. In the experimental part we make some experiments while varying the dimension of the spreading window, showing that, as expected, our choice is almost always the best choice.

Figure 2 shows the effect of our new (dis)similarity measure. Two posteriors are displayed, each with a peak in a particular zone of the grid. When using a punctual kernel (such as the histogram intersection kernel), which needs aligned grids, we can observe that even if the two peaks are close in the grid the intersection is almost null, and therefore the similarity is null as well (see Figure 2(top)). Conversely, in Figure 2(center) and 2(bottom) the grid intersection, and therefore the similarity, is significative and it increases with the size of the convolution window.

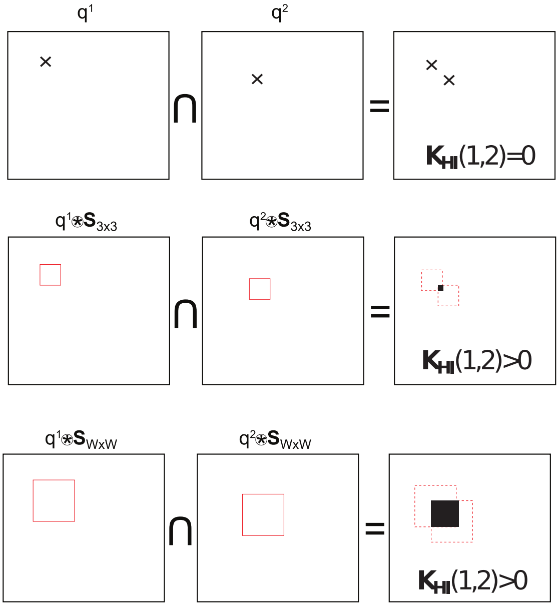


Fig. 2. The spreding effect of using our approach in comparing q_1 and q_2. The histogram intersection ($K_{HI}(\cdot,\cdot)$) is considered as measure $SM(\cdot,\cdot)$. When standard K_{HI} is considered no intersection is observed (top), while using the spreding strategy the similarity between q_1 and q_2 is significative (center), and it increases with the size of $\mathbf{W}$.

As a further note: it is straightforward to show that if $SM(\cdot)$ is a kernel, also our $SSM_S(\cdot,\cdot)$ is a kernel. This may be of great practical importance, since permits to develop a hybrid generative-discriminative scheme where SVM can be used as discriminative classifiers.

4 Experimental Evaluation

In this section the experimental evaluation is presented. In particular, the proposed framework is evaluated within two biomedical applications: cancer classification via the analysis of expression microarray and schizophrenia detection through brain classification using MRI scans.

4.1 Microarray Classfication

In this application, the goal is to analyze gene expression microarray data in order to distinguish between healthy people and people affected by cancer. The starting point is a microarray gene expression matrix, where the element at position (i, j) represents the expression level of the $i - th$ gene in the $j - th$ subject/sample. Methods based on counting values (as CG and topic models) have been recently and successfully applied in this context (see, *e.g.*, [16, 17, 11]). This is possible if we establish an analogy between a word-document pair and a gene-sample pair; it seems reasonable to interpret samples as documents and genes as words. In this way, the gene expression levels in a sample may interpreted as the word counts in a document. Consequently, we can simply take a gene expression matrix and (of course, after a preprocessing step, for example, to remove possibly negative numbers [16]) interpret it as a count matrix **C** from which a CG or a LDA model can be estimated.

The experiments presented in this paper have been performed using two microarray datasets: the ovarian [18] and the colon [19] datasets, whose characteristics are summarized in table 1.

Table 1. Summary of the employed microarray datasets

Dataset Name	n. of genes	n. of samples	n. of classes	citation
1. Ovarian cancer	1513	53	2	[18]
2. Colon cancer	2000	62	2	[19]

4.2 Brain Classification

In this application the main goal is to distinguish between healthy and schizophrenic people through the classification of MRI brain scans.

Data Set. The study population used in this work consists of 42 patients (21 male, 21 female) who were being treated for schizophrenia and 40 controls (19 male, 21 female) with no DSM-IV axis I disorders and had no psychiatric disorders among first-degree relatives. Diagnoses for schizophrenia were corroborated by the clinical consensus of two psychiatrists. T1 weighted structural MRI scans were acquired with a 1.5 Tesla machine and to minimize biases and head motion, restraining foam pads were used. The original image size is 384x512x144; these images are then rotated and realigned to a resolution of 256x256x192. After this alignment, they were segmented into specific brain regions called Regions of Interest (ROIs) manually by experts following a specific protocol for each ROI [20]. In this work, we use three ROIs from the two hemispheres of the brain summing upto a total of six different brain regions: Dorsolateral prefrontal cortex (*ldlpfc* and *rdlpfc*), Entorhinal Cortex (*lec* and *rec*), and Thalamus (*lthal* and *rthal*) which are found to be impaired in schizophrenic patients.

Preprocessing. After the alignment and ROI tracing, DARTEL [21] tools within SPM software [22] was used to pre-process the data. Initially, images are segmented into grey and white matter in *Native* and *DARTEL imported* spaces. The DARTEL imported images have lower resolution than the original images but are used to spatially align to standard MNI atlas. In the second step, DARTEL template generation is applied which creates an average template from the input data while simultaneously aligning white and grey matter. In this step, the flowfields of the registration are also computed which will be used to segment the MNI space normalized images into ROIs. In the final step, the DARTEL template is used to spatially normalize all images into standard MNI space. In this way, smoothed (12 mm Gaussian), and Jacobian scaled grey matter images are constructed which is general practice in neuroimaging applications.

Feature Extraction. The images at the end of the preprocessing pipeline are the intensity probability maps which are then used to construct the features for our classification experiments. Since we already have ROI segmented source images, using the flow fields computed in the second step of preprocessing we create the intensity maps for every subject and ROI instead of extracting a single set of features for the whole brain. Since the ROIs have different bounding boxes, the sizes of these images are not the same for all subjects. By applying thresholding at 0.2 level, we compute histograms of probability maps for every subject and ROI. Number of bins in each histogram is chosen to be 40 which showed the best performance in our experiments. As a result, we have a data set of six different ROIs, 82 subjects with a counting vector of size 40 which we apply our classification pipeline.

4.3 Experimental Details

The experimental evaluation is aimed at validating the proposed approach. In particular, we start by assessing the baseline CG results, without any spreading operation, using the ovarian dataset. Then we evaluate the impact of the proposed approach. Third, we investigate the impact of the dimension of the spreading window. Finally, we show some more results on the colon microarray experiment and on the Brain MRI classification task.

For all the experiments the following protocol has been adopted:

- Since, as a base level, we are mostly interested in the quality of unsupervised learning of the distributions over the samples, the whole dataset has been used to train a CG (of course labels are ignored in this phase), in a transductive way [13, 4]. As explained in the methodological section, here we employed bidimensional squared Counting Grid models (in principle, also higher dimensional/not squared grids can be used, see [1]). Two parameters should be set when learning the Couting Grid: the dimension of the Grid $\mathbf{E}$ and the dimension of the Window $\mathbf{W}$. Here we performed a large scope analysis, reporting results for many different configurations, with $\mathbf{E}$ ranging from $[10, 10]$ to $[90, 90]$, and $\mathbf{W}$ ranging from $[4, 4]$ to $[19, 19]$[2]. An interesting parameter which can be used to summarize the dimension of a Counting Grid is the capacity κ, which, as explained in the methodological section, represents the ratio between the dimension of the grid and the dimension of the window, and can be seen as the number of topics in the standard topic models.

[2] Of course only valid configurations were retained – e.g. $\mathbf{E} = [10, 10]$, $\mathbf{W} = [15, 15]$ is not a valid configuration.

- In order to avoid to get stuck in local optima during the learning procedure (given the initialization, E-M converges to the nearest local optima), we repeated the training 10 times, starting from random initialization, retaining the model with the highest training likelihood.
- Given the model, an hybrid generative-discriminative approach is used to perform classification. In particular, for every pair of samples A, B, represented by counts $\{c_z^A\}$, $\{c_z^B\}$, we computed its posterior $q_{\mathbf{k}}^A$, $q_{\mathbf{k}}^B$ given the learned counting grid, comparing them with a kernel, employed to perform a discriminative classification via Support Vector Machines. In all experiments the parameter C of the SVM was set, after some preliminary evaluations, to 10000.
- In all experiments, classification accuracy has been computed using Leave-One-Out Cross validation, as typically done with these small size problems.
- In all the experiments we also computed and reported the performances of the Latent Dirichlet Allocation (LDA - [23]), the most famous topic model, whose usefulness has been already shown in these contexts [17, 16, 24]. LDA can straightforwardly be considered as a counting grid where the Window Size is equal to 1, since there are no interactions between latent variables (i.e. topics). For classification, the same hybrid generative-discriminative approach explained before is used. In this case, given a pair of samples A, B, the posterior Dirichlet parameters have been computed through the learned LDA model and compared via a kernel, to be used in a SVM classification scenario. Given the parallelism between the concept of the capacity of the Counting Grids and the number of topics, we performed an experiment with LDA for every capacity value experimented for our approach.

Similarity Measures and Kernels Concerning the similarity measures / kernels to be adopted in our hybrid generative-discriminative scheme, different options can been used. Given the modularity of our proposed scheme, we can straightforwardly apply the same kernel $S(\cdot, \cdot)$ with and without performing the spreading via the convolution. This will permit us to directly investigate the impact of the spreading operation. In particolar, we experimented three different options:

1. *Linear Kernel.* This is the standard inner product between the representations of the two objects, namely

$$K^{\text{ LI}}(q_{\mathbf{k}}^A, q_{\mathbf{k}}^B) = q_{\mathbf{k}}^A \cdot q_{\mathbf{k}}^B \tag{3}$$

2. *Jensen Shannon Kernel.* This represents a standard and well known Information Theoretic Kernel, namely a kernel based on probability measures. These kernels have been shown very effective in classification problems involving text, images, and other types of data [25–27]. Very recently, moreover, they have been found to be very suitable in hybrid generative discriminative scenarios [28]. Given two posterior probabilities $q_{\mathbf{k}}^A$ and $q_{\mathbf{k}}^B$, representing two objects, the Jensen-Shannon kernel is defined as

$$K^{\text{ JS}}(q_{\mathbf{k}}^A, q_{\mathbf{k}}^B) = \ln(2) - JS(q_{\mathbf{k}}^A, q_{\mathbf{k}}^B), \tag{4}$$

with $JS(q^A_{\mathbf{k}}, q^B_{\mathbf{k}})$ being the Jensen-Shannon divergence

$$JS(q^A_{\mathbf{k}}, q^B_{\mathbf{k}}) = H\left(\frac{q^A_{\mathbf{k}} + q^B_{\mathbf{k}}}{2}\right) - \frac{H(q^A_{\mathbf{k}}) + H(q^B_{\mathbf{k}})}{2}, \tag{5}$$

where $H(p)$ is the usual Shannon entropy.

3. *Histogram Intersection Kernel.* This Kernel, initially designed to compare histograms, can be safely used also in case of multinomials (as the Counting Grid posteriors), which are simply normalized Histograms. Given two object representations $q^A_{\mathbf{k}}$ and $q^B_{\mathbf{k}}$, the kernel is defined as [29]

$$K^{\text{ HI}}(q^A_{\mathbf{k}}, q^B_{\mathbf{k}}) = \sum_k \min(q^A_k, q^B_k) \tag{6}$$

A further note: by looking at the formulation of our proposed dissimilarity measure, some similarities with the diffusion distance [30] can be found. Actually, in both cases, the value of every particular point is spread/diffused in its neighborhood. It seems therefore interesting to compare our approach with this distance[3], applied on the original model posteriors. More in detail, the distance between two representations $q^A_{\mathbf{k}}$, $q^B_{\mathbf{k}}$ is defined as a temperature field $T(\mathbf{k}, t)$ with $T(\mathbf{k}, 0) = q^A_{\mathbf{k}} - q^B_{\mathbf{k}}$. Using the heat diffusion equation

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial \mathbf{k}^2}$$

which has a unique solution

$$T(\mathbf{k}, t) = T(\mathbf{k}, 0) * \phi(\mathbf{k}, t)$$

where

$$\phi(\mathbf{k}, t) = \frac{1}{(2\phi)^{1/2} t} \exp -\frac{\mathbf{k}^2}{2t^2},$$

we can compute the distance D as:

$$D(q^A_{\mathbf{k}}, q^B_{\mathbf{k}}) = \int_0^r \eta(|T(\mathbf{k}, t)|) dt$$

where $\eta(\cdot, \cdot)$ is a norm which measures how $T(\mathbf{k}, t)$ differs from 0. Given this distance, we can obtain a kernel following the extended gaussian kernels recipe [31]:

$$K^{\text{ DD}}(q^A_{\mathbf{k}}, q^B_{\mathbf{k}}) = e^{-\rho D(q^A_{\mathbf{k}}, q^B_{\mathbf{k}})} \tag{7}$$

In our experiments, following the suggestion given in [32], the scale parameter ρ has been set to the average diffusion distance between all pairs of objects in the training set.

[3] The code has been taken from the author's home page: `http://www.ist.temple.edu/~hbling/code_data.htm`

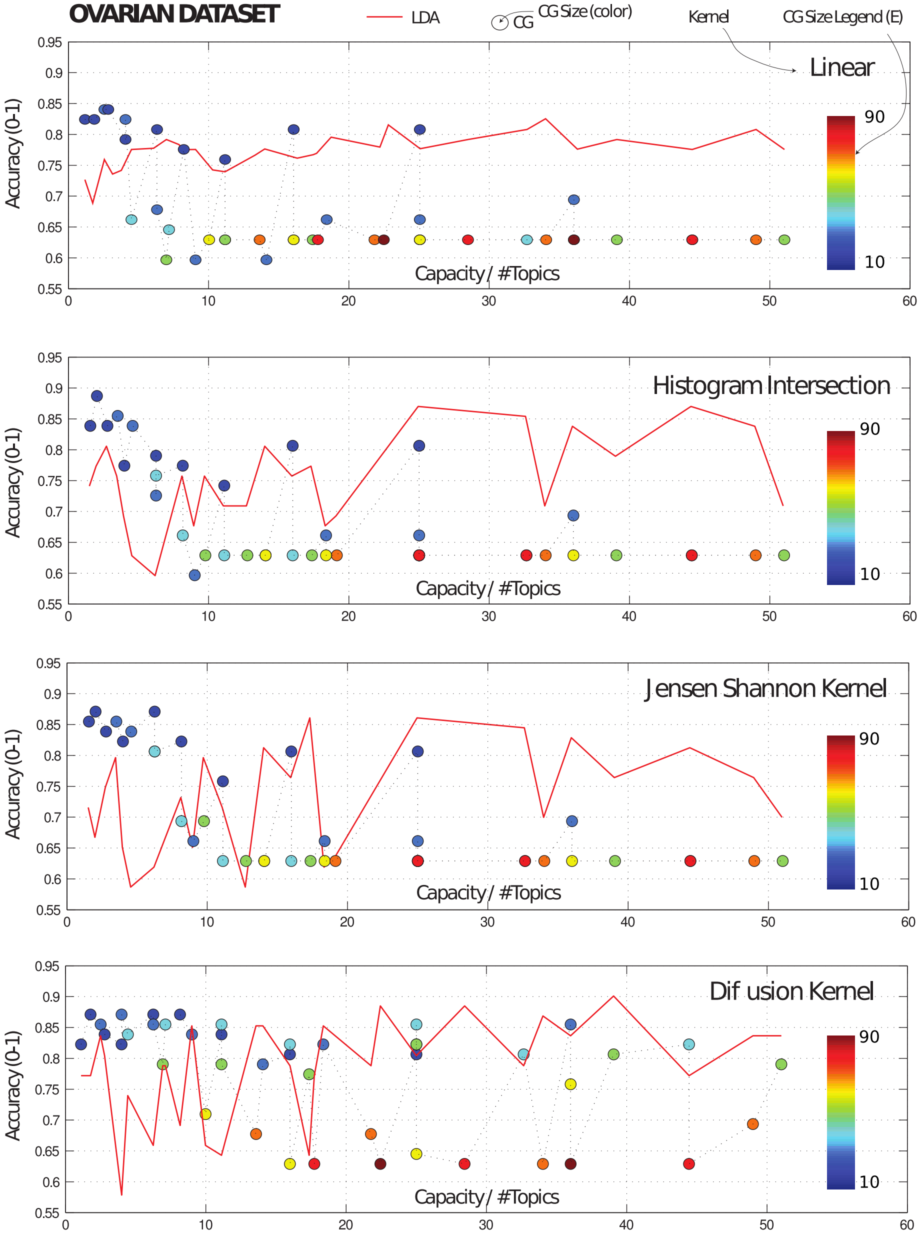


Fig. 3. Baseline results

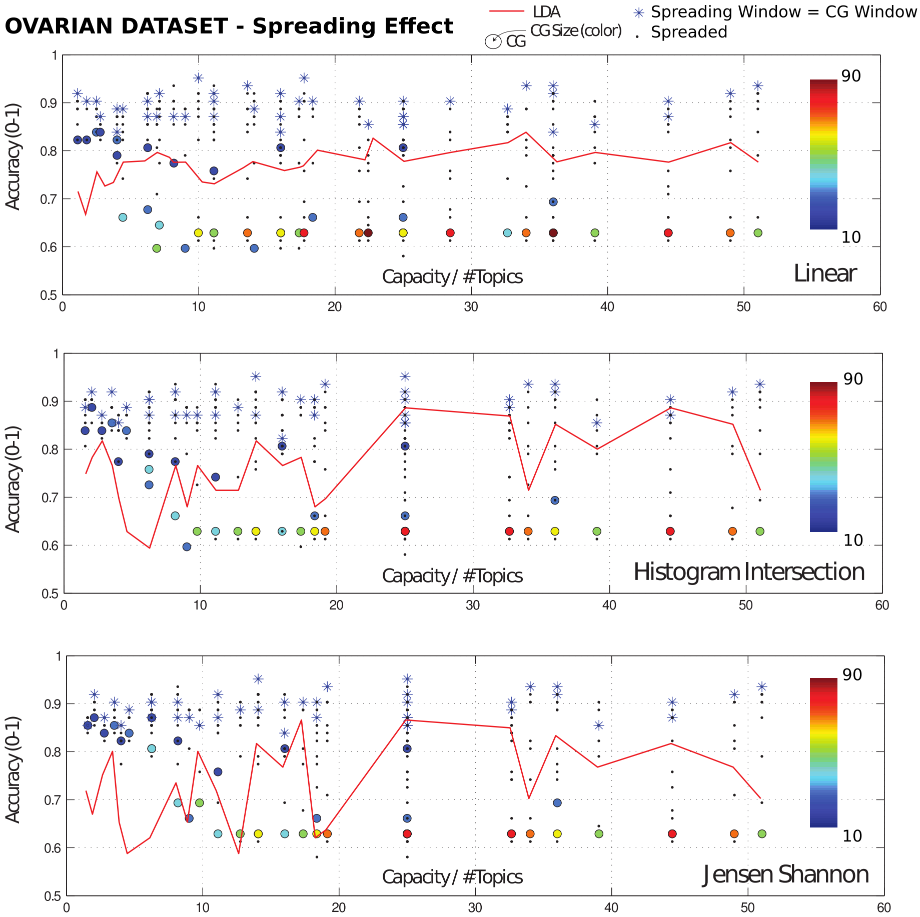


Fig. 4. Results obtained with the proposed spreading operation

4.4 Results

Results are presented in figures 3, 4, 5 and 6. More in detail, in Figure 3 the performances of the original Counting Grids scheme, without any spreading operation, are presented for the different kernels. In particolar, on the x-axis we have the different model size (different capacities), whereas in the y-axis we reported the accuracy. The solid line represents the performances of the LDA. The dimension E of the counting grid is represented by the color. From this figure we can infer that Counting grids are better than the LDA model only for small capacities, whereas for larger capacities the simpler LDA model is preferrable. Moreover it can be noted that the diffusion distance-based kernel represents the best choice (especially for LDA), confirming the intuition that diffusing the values of the posterior represents a good idea. This is more evident by looking at Figure 4, where we plot also the results with the proposed approach (marked with

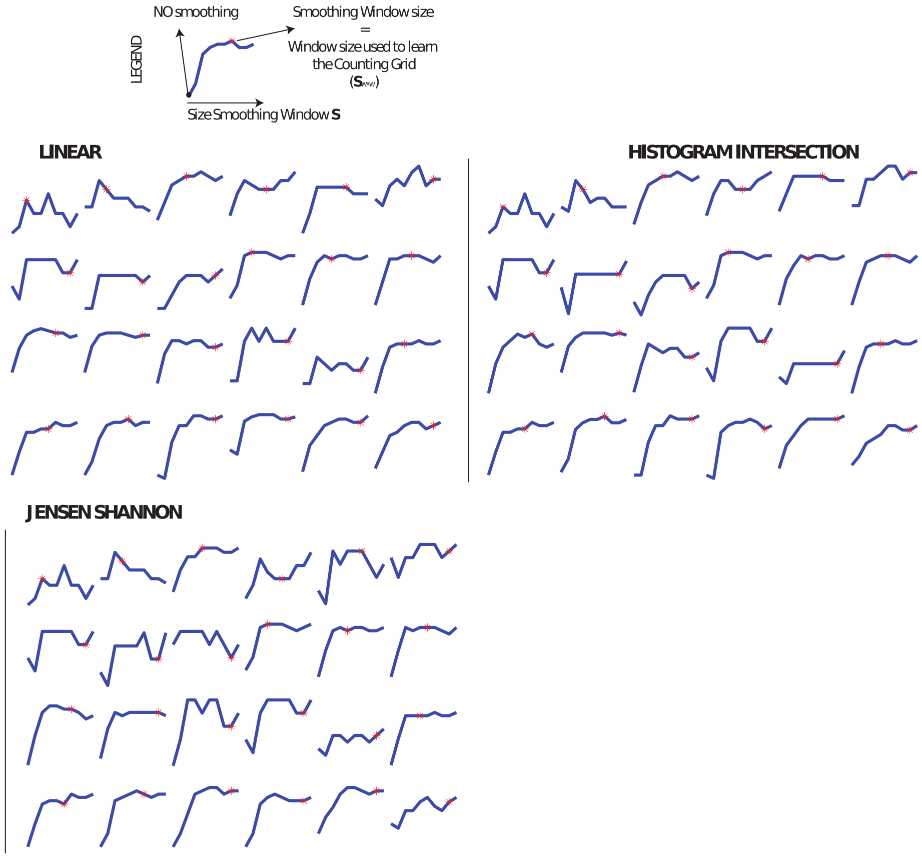


Fig. 5. Analysis of the impact of the dimension of the spreading window

an asterisk), for three of the four kernels – we excluded the diffusion distance-based kernel since already possessing the property of spreading the values. In this case, results with the Counting Grids always outperform the corresponding LDA, making the choice of the capacity less crucial. In that figure, moreover, we also plotted the different accuracies obtained by varying the dimension (from 2 to 10) of the spreading window (marked with a dot). From this figure, it is evident that selecting as the size of the spreading window the size of the counting grid window almost always represents the best choice, as expected. This can be confirmed with the analysis plotted in Figure 5, where for some configurations of the Counting Grid the accuracy for different values of the spreading window is plotted. Also in this case, the asterisk indicates the CG window size, which is almost everywhere among the best values.

Finally, with the same visualization scheme of figure 4, in figure 6 we plot results for the MRI Brain dataset and for the colon cancer microarray dataset. Also in these cases it is evident the gain obtained by the spreading approach.

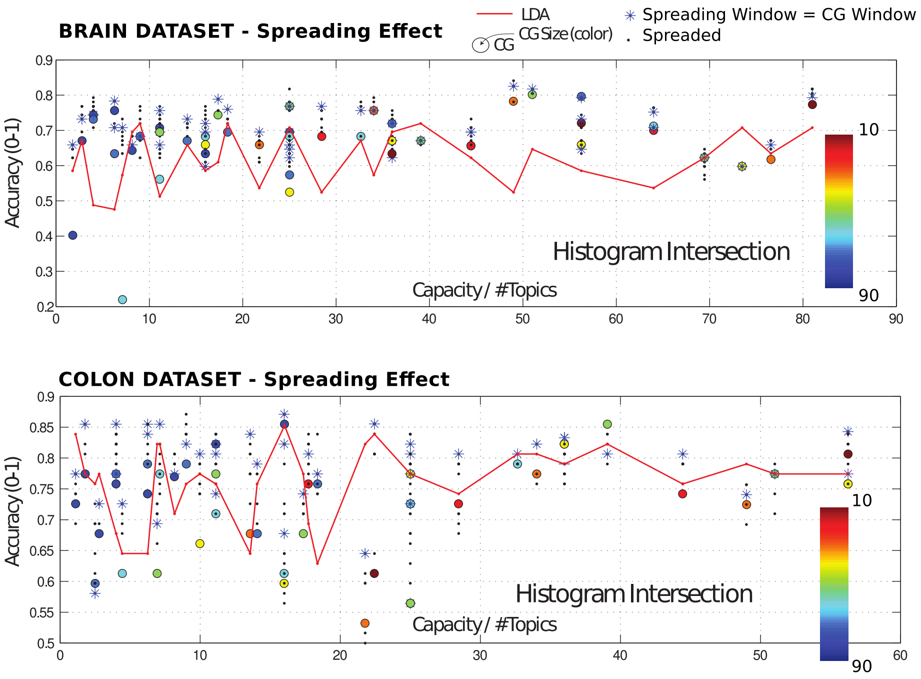


Fig. 6. Results on other datasets

5 Conclusions

In this paper a new approach to compare data represented by counts is introduced. Starting from the recently proposed CGs, we show how the classification perfomance can increase by carefully taking into account of information coming from the generative learning procedure. The proposed Spreading Similarity Mesure leads to a drastic improvement in comparison with standard approaches as shown on different applicative scenarios. In particular, our SSM approach outperfoms diffusion distance which is known to well dealing with cross-count contraints.

Acknowledgements. Authors would like to thank Dr. A. Ulas for the help in the preprocessing of the Brain MRI dataset.

References

1. Jojic, N., Perina, A.: Multidimensional counting grids: Inferring word order from disordered bags of words. In: Uncertainty in Artificial Intelligence (2011)
2. Rogers, S., Girolami, M., Campbell, C., Breitling, R.: The latent process decomposition of cdna microarray datasets. IEEE/ACM Transactions on Computational Biology and Bioinformatics (2005)

3. Bicego, M., Lovato, P., Oliboni, B., Perina, A.: Expression microarray classification using topic models. In: SAC, pp. 1516–1520 (2010)
4. Perina, A., Lovato, P., Cristani, M., Bicego, M.: A comparison on score spaces for expression microarray data classification. In: Loog, M., Wessels, L., Reinders, M.J.T., de Ridder, D. (eds.) PRIB 2011. LNCS, vol. 7036, pp. 202–213. Springer, Heidelberg (2011)
5. Cruska, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV Workshop on Statistical Learning in Computer Vision, pp. 1–22 (2004)
6. Toldo, R., Castellani, U., Fusiello, A.: The bag of words approach for retrieval and categorization of 3D objects. The Visual Computer 26(10), 1257–1268 (2010)
7. Brelstaff, G., Bicego, M., Culeddu, N., Chessa, M.: Bag of peaks: interpretation of nmr spectrometry. Bioinformatics 25, 258–264 (2009)
8. Castellani, U., Rossato, E., Murino, V., Bellani, M., Rambaldelli, G., Perlini, C., Tomelleri, L., Tansella, M., Brambilla, P.: Classification of schizophrenia using feature-based morphometry. Journal of Neural Transmission 119, 395–404 (2012)
9. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)
10. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Mach. Learn. 42(1-2), 177–196 (2001)
11. Lovato, P., Bicego, M., Cristani, M., Jojic, N., Perina, A.: Feature selection using counting grids: application to microarray data. 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. 629–637. Springer, Heidelberg (2012)
12. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
13. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
14. Swain, M.J., Ballard, D.H.: Color indexing. International Journal of Computer Vision 7(1), 11–32 (1991)
15. Lin, J.: Divergence measures based on the shannon entropy. IEEE Transactions on Information Theory 37(1), 145–151 (1991)
16. Bicego, M., Lovato, P., Perina, A., Fasoli, M., Delledonne, M., Pezzotti, M., Polverari, A., Murino, V.: Investigating topic models' capabilities in expression microarray data classification. IEEE/ACM Trans. Comput. Biology Bioinform. 9(6), 1831–1836 (2012)
17. Perina, A., Lovato, P., Murino, V., Bicego, M.: Biologically-aware latent Dirichlet allocation (balda) for the classification of expression microarray. In: Dijkstra, T.M.H., Tsivtsivadze, E., Marchiori, E., Heskes, T. (eds.) PRIB 2010. LNCS, vol. 6282, pp. 230–241. Springer, Heidelberg (2010)
18. Dhanasekaran, S., Barrette, T., Ghosh, D., Shah, R., Varambally, S., Kurachi, K., Pienta, K., Rubin, M., Chinnaiya, A.: Delineation of prognostic biomarkers in prostate cancer. Nature 412(6849), 822–826 (2001)
19. Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., Levine, A.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Natl. Acad. Sci. 96, 6745–6750 (1999)
20. Baiano, M., Perlini, C., Rambaldelli, G., Cerini, R., Dusi, N., Bellani, M., Spezzapria, G., Versace, A., Balestrieri, M., Mucelli, R.P., Tansella, M., Brambilla, P.: Decreased entorhinal cortex volumes in schizophrenia. Schizophrenia Research 102(1-3), 171–180 (2008)

21. Ashburner, J.: A fast diffeomorphic image registration algorithm. Neuroimage 38(1), 95–113 (2007)
22. Friston, K., Ashburner, J., Kiebel, S., Nichols, T., Penny, W. (eds.): Statistical Parametric Mapping: The Analysis of Functional Brain Images. Academic Press (2007)
23. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
24. Castellani, U., Perina, A., Murino, V., Bellani, M., Rambaldelli, G., Tansella, M., Brambilla, P.: Brain morphometry by probabilistic latent semantic analysis. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) MICCAI 2010, Part II. LNCS, vol. 6362, pp. 177–184. Springer, Heidelberg (2010)
25. Martins, A., Smith, N., Xing, E., Aguiar, P., Figueiredo, M.: Nonextensive information theoretic kernels on measures. Journal of Machine Learning Research 10, 935–975 (2009)
26. Cuturi, M., Fukumizu, K., Vert, J.P.: Semigroup kernels on measures. Journal of Machine Learning Research 6, 1169–1198 (2005)
27. Jebara, T., Kondor, R., Howard, A.: Probability product kernels. Journal of Machine Learning Research 5, 819–844 (2004)
28. Bicego, M., Ulas, A., Castellani, U., Perina, A., Murino, V., Martins, A., Aguiar, P., Figueiredo, M.: Combining information theoretic kernels with generative embeddings for classification. Neurocomputing 101, 161–169 (2013)
29. Odone, F., Barla, A., Verri, A.: Building kernels from binary strings for image matching. IEEE Transactions on Image Processing 14(2), 169–180 (2005)
30. Ling, H., Okada, K.: Diffusion distance for histogram comparison. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2006, vol. 1, pp. 246–253 (2006)
31. Chapelle, O., Haner, P., Vapnik, V.: Support vector machines for histogram-based image classifcation. IEEE Transactions on Neural Networks 10(5), 1055–1064 (1999)
32. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. International Journal of Computer Vision 73(2), 213–238 (2007)

On the Dissimilarity Representation and Prototype Selection for Signature-Based Bio-cryptographic Systems

George S. Eskander, Robert Sabourin, and Eric Granger

Laboratoire d'Imagerie, de Vision et d'Intelligence Artificielle,
Ecole de Technologie Supérieure, Université du Québec, 1100 Rue Notre-Dame Ouest,
Room A-3600, Montréal, QC, H3C 1K3, Canada
geskander@livia.etsmtl.ca, {robert.sabourin,eric.granger}@etsmtl.ca

Abstract. Robust bio-cryptographic schemes employ encoding methods where a short message is extracted from biometric samples to encode cryptographic keys. This approach implies design limitations: 1) the encoding message should be concise and discriminative, and 2) a dissimilarity threshold must provide a good compromise between false rejection and acceptance rates. In this paper, the dissimilarity representation approach is employed to tackle these limitations, with the offline signature images are employed as biometrics. The signature images are represented as vectors in a high dimensional feature space, and is projected on an intermediate space, where pairwise feature distances are computed. Boosting feature selection is employed to provide a compact space where intra-personal distances are minimized and the inter-personal distances are maximized. Finally, the resulting representation is projected on the dissimilarity space to select the most discriminative prototypes for encoding, and to optimize the dissimilarity threshold. Simulation results on the Brazilian signature DB show the viability of the proposed approach. Employing the dissimilarity representation approach increases the encoding message discriminative power (the area under the ROC curve grows by about 47%). Prototype selection with threshold optimization increases the decoding accuracy (the Average Error Rate AER grows by about 34%).

Keywords: Dissimilarity-representation, Prototype selection, Bio-Cryptography, Offline signatures.

1 Introduction

Bio-cryptographic systems are introduced to replace the traditional usage of simple user passwords by biometric traits like fingerprint, iris, face, signatures, etc., to secure the cryptographic keys within security schemes like encryption and digital signatures [1]. Different than the simple passwords, biometrics provide a more trusted authentication tool. However, their fuzzy nature harden the classification decision. Similarities between inter-personal traits result in false

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 265–280, 2013.

acceptance and dissimilarities between intra-personal traits result in false rejections.

Robust bio-cryptographic systems operate in the key-binding mode where classical crypto-keys are coupled with the biometric message. For key binding, some encoding schemes like Fuzzy Commitment [2] and Fuzzy Vault (FV) [3] are the most commonly employed. In the enrollment phase, a prototype biometric message encodes the secret key. In the authentication phase, a message is extracted from the query sample to decode the key. The idea behind these schemes is to consider the query biometric message as a noisy version of the encoded message. If the query sample is genuine, the dissimilarity between the encoding and decoding messages is limited, so this noise can be eliminated by the decoder. On the other hand, if the query sample belongs to another person, or if it is a forged sample, the dissimilarity between the two messages is too high to cancel. Accordingly, the secret key will be unlocked only to users who apply similar enough query samples.

Some error correction codes like R-S codes [4] are employed to realize the key binding approach. Practical decoding complexity of such codes need that employed biometric messages should be concise. Also, error correction capacity of such codes can be controlled by adjusting a dissimilarity threshold. The decoder succeeds to unlock the secret, only if the dissimilarity between the prototype and the query message is beyond the threshold. Accordingly, this threshold should be properly adjusted based on the expected dissimilarity ranges. So that, the code can cancel the intra-personal dissimilarities and fails to cancel the inter-personal dissimilarities.

For physiological biometrics like fingerprint and iris, small number of simple features extracted in the spacial domain can be employed to constitute informative encoding messages. This is simply because the intrinsic stability and discriminative nature of such biometrics. On the other hand, for behavioral biometrics like offline signature images, the intra-personal variability and inter-personal similarity are intrinsic properties. Moreover, it is easy to produce forged signature images. Accordingly, discrimination between genuine and forged signatures needs high dimensional feature representation and complicated classifiers [5]. It is a challenging task to produce a concise and informative messages from the signature images, and to use simple classifiers like the bio-cryptographic decoders to differentiate between genuine and forged signatures.

In this paper, design of reliable decoders for offline signature-based bio-cryptography is tackled by employing the concept of dissimilarity-representation [6]. This concept is originally introduced to build classical classifiers, by replacing the feature representation of objects by their dissimilarity to a fixed set of prototypes. Performance of these classifiers relies on the accuracy of the employed dissimilarity measure and how carefully the prototypes are chosen [8]. In literature, dissimilarity measures often composed of graphs, strings, or normalized versions of the raw measurements. However, the dissimilarity approach may also be used on top of a feature representation, where object proximity is represented by computing the distance between ordinary feature representations in a vectorial space [7].

As most of work on classical offline signature verification is feature-based, where many techniques of feature extraction are already proposed [5], we base our method on top of a feature representation. In such case, the encoding messages are composed of a set of features. The dissimilarity between the prototype and query messages is measured by the distance between the feature vectors that constitute these messages. The rational behind the proposed method is that the overall dissimilarity between two messages is an accumulation of individual dissimilarities between every pair of corresponding elements of the message. So, to increase the separation between the intra-personal and inter-personal dissimilarity ranges, we select features that decrease the intra-personal distances and that increase the inter-personal distances.

The enrolling signature images are first represented as vectors in a high dimensional feature space. This representation is projected on an intermediate space, which we call a "feature-dissimilarity" space, where pairwise feature distances are computed. Boosting feature selection is employed in this intermediate space, producing a compact space with the intra-personal distances are minimized and the inter-personal distances are maximized. Finally, the resulting representation is projected on the dissimilarity space to select the most discriminative prototypes for encoding, along with optimizing the dissimilarity threshold.

For proof of concept simulations, the Brazilian signature DB (including genuine and samples with different levels of forgeries) is employed [9]. The impact of proposed dissimilarity representation approach is investigated by analyzing the separation between the intra-personal and the inter-personal dissimilarity distributions. The benefit of prototype selection with optimizing the dissimilarity threshold is tested by its impact on the overall recognition accuracy.

The rest of this paper is organized as follows. The next section provides some background on the dissimilarity representations as applied to bio-cryptographic offline signature based systems. The proposed dissimilarity representation and prototype selection approach for designing signature-based bio-cryptographic systems is illustrated in section 3. The experimental methodology is illustrated in section 4. The experimental results are presented and discussed in section 5.

2 Background

Signature Verification systems (SV) are employed to authenticate individuals based on their handwritten signatures. Classical SV systems output a simple acceptance/rejection decision for a query signature sample. On the other hand, signature-based bio-cryptographic systems release a secret cryptographic key only for a user who applies a genuine signature sample. There are two modes of operation for signature-based systems: online and offline. For online systems, users use special devices like special pens and tablets to acquire their signature dynamics such as velocity, pressure, etc. On the other hand, offline signature-based systems use scanned signature images for the recognition task. Only static information can be acquired from the signature images, producing less informative signals, and hence, a harder pattern recognition task.

Most of work done in the signature verification area applied feature-based pattern recognition approaches, where feature representations are constituted from signature signals. The classifiers are then designed in the feature space. Performance of such systems are basically limited by the quality of employed feature representations.

Handwritten signature images imply high variability between different user samples, and also high similarity between signatures of different users. Accordingly, the feature-based approach succeeds to produce offline SV verification systems, only when high dimensional feature representations and complex classifiers are employed. For a comprehensive review on the different approaches see [5].

For bio-cryptographic systems design, there are some restrictions on the size of the employed feature representations, and on the classification complexity. Accordingly, direct application of the feature-based approach produces inaccurate systems. In literature, few bio-cryptographic implementations are done based on the handwritten signatures. The online signatures produced bio-cryptographic systems with acceptable performance [14], as discriminative features like velocity, pressure, etc, are employed. On the other hand, it is shown that static features extracted from the offline signature images are unstable and they are not discriminant enough to design a bio-cryptographic system [15].

Different than the feature-based approach, the concept of dissimilarity-based classification has been proposed by Elzbieta Pekalska and Robert P.W. Duin., [6]. The rational behind this concept is that modeling the proximity between objects may be more discriminative than modeling the objects themselves. This is because objects belong to a specific class have a shared degree of commonality that could be captured by a dissimilarity value.

We propose that the dissimilarity-based approach can be employed to design reliable key-binding bio-cryptographic systems. In such systems, error correction-based decoders are used. If the dissimilarity between the decoding and the encoding signals is less than a specific threshold, the decoder succeeds to decouple the encoded bio-ctyptographic key. So, functionality of these decoders can be considered as two-class simple thresholding classifiers that operate in the dissimilarity space.

In literature, the concept of dissimilarity representation is not directly employed to design bio-cryptographic systems. However, some authors proposed methodologies to absorb the dissimilarities between encoding and decoding biometric signals, so that they are within the error correction capacity of the decoder. For instance, Fingerprint-based fuzzy vaults are designed by using some minutia points extracted in the spatial space to constitute the encoding message [16]. The dissimilarity between encoding and decoding messages is decreased by aligning the query and the template fingerprints prior to the decoding process. For our proposed method, instead of aligning the dissimilar messages, we design them in a way that produces similar intra-personal messages and dissimilar inter-personal encoding messages. A preliminary realization of the proposed method is appeared in [17], where a Fuzzy Vault (FV) system based on the offline signature images is proposed. Boosting feature selection (BFS) is employed to select informative representation, so that intra-personal dissimilarities are minimized and inter-personal dissimilarities are maximized. Although produced discriminative representations, this method did not cancel some of the intrinsic fuzziness of the signature signals.

In this paper, we extend the method in [17], so that some of the residual fuzziness of the signature representations is canceled. Inspired by fingerprint alignment technique proposed by Nandakumar et al., [16], we model the representation dissimilarities, and use this information to absorb the residual message fuzziness before sending it to the bio-cryptographic decoders. Moreover, as quality of representation relies mainly on the quality of employed reference signatures (few work is done on selecting a reference subset for classical signature verification systems, e.g., [10].), we extend this idea to the bio-cryptography domain. The designed messages are projected to the dissimilarity space, where each dimension is the message distance to a prototype message. In this space, the most discriminative prototypes are selected, along with optimizing the dissimilarity threshold.

3 Proposed Dissimilarity Representation and Prototype Selection Method

Assume an encoding biometric message: $E^p = \{f_i^p\}_{i=1}^t$, where p is the signature prototype used for message extraction, f_i^p is a feature extracted from p to constitute a message element, and t is the message length. In the enrollment phase, E^p is extracted and used to encode a secret cryptographic key K. In the authentication time, a decoding query message $E^Q = \{f_i^Q\}_{i=1}^t$ is extracted, where Q is the query signature sample applied to decode the locked key K[1]. Assume the dissimilarity between the two messages is D^{Qp}. For error correction decoders like the R-S decoders [4], the decoder succeeds to cancel the dissimilarity between Q and p, if the dissimilarity (error) D^{Qp} is less than its error correction capacity Θ. Hence, decoder functionality DF can be formulated as follows:

$$DF = \begin{cases} 1 & if \ \ D^{Qp} \ \leq \ \Theta \\ 0 & if \ \ D^{Qp} \ > \ \Theta \end{cases} \tag{1}$$

where Θ is the error correction capacity of the decoder (dissimilarity threshold). Hence, to achieve perfect decoding accuracy, the following condition should be satisfied:

[1] Details of how the crypto-key is encoded/decoded by means of a biometric message is out of the scope of this paper. For more details on this aspect see [3], and [2].

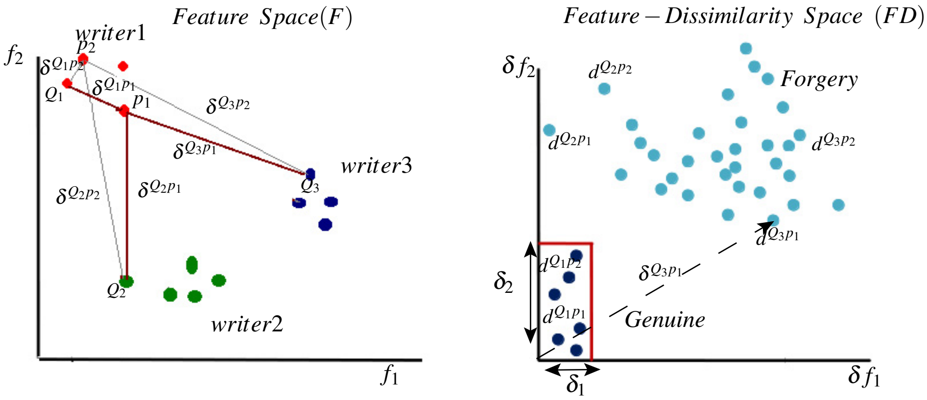


Fig. 1. Illustration of feature selection in the original feature space (left) and in the feature-dissimilarity space (right)

$$D^{Qp} \begin{cases} \leq \Theta & if\ Q\ is\ a\ genuine\ sample \\ > \Theta & if\ Q\ is\ a\ forgery\ sample \end{cases} \tag{2}$$

Satisfying the above condition relies on the following design issues:

1. selection of the message elements $\{f_i\}_{i=1}^{t}$.
2. the dissimilarity measure employed to produce the dissimilarity score D^{Qp}.
3. selection of the signature prototype p for encoding.
4. the correction capacity of the decoder (dissimilarity threshold) Θ.

In this paper we propose a methodology to optimize these design issues, so high decoding accuracy is achieved. The proposed method consists of two main stages: 1) design of the encoding messages and the dissimilarity measure, and 2) prototype selection and dissimilarity threshold optimization.

3.1 Design of the Encoding Messages and the Dissimilarity Measure

For a message of length t, consider Euclidean distance $\delta^{Q_jp_r}$ between the query message Q_j and the prototype p_r:

$$\delta^{Q_jp_r} = \sqrt{\sum_{i=1}^{t} (\delta f_i^{Q_jp_r})^2} \tag{3}$$

where $\delta f_i^{Q_jp_r} = \|f_i^{Q_j} - f_i^{p_r}\|$.

Hence, the overall dissimilarity between messages is an accumulation of the individual dissimilarities between every two corresponding elements of the message. So, to increase the separation between the intra-personal and inter-personal dissimilarity ranges, we select features that decrease the intra-personal distances and that increase the inter-personal distances.

The enrolling signature images are first represented as vectors in a high dimensional feature space F. This representation is projected on an intermediate space, which we call a "feature-dissimilarity" space FD, where pairwise feature distances are computed. Figure 1 illustrates the transformation from space F to space FD. In the left side, signatures of three writers are represented in F. For simplicity, only two features f_1 and f_2 are shown in this figure, while typical representations might have high dimensionality. In this example, we assume that writer 1 is the only authentic person, whose signatures should succeed to decode the cryptographic key K. Two signatures are considered as prototypes for this user, p_1 and p_2. Euclidean distance is employed as a dissimilarity measure. It is clear that a dissimilarity representation that is built on top of this feature representation is discriminative. Distances among intra-personal signatures (like $\delta^{Q_1 p_1}$) are generally smaller than the distances among inter-personal signatures (like $\delta^{Q_2 p_1}$). However, in this space it is not clear which feature is more discriminative. With representations of high dimensionality, high number of system users, unknown forgeries and a small number of training samples, it is not feasible to select the most discriminative features in the feature space F.

Accordingly, we project this representation on a feature-dissimilarity space FD, as shown in the right side of Figure 1. In this space, distance between each corresponding features, for each pair of signatures, is computed and used as new set of features $\{\delta f_i\}_{i=1}^{t}$. So, dimensionality of the F and FD spaces is equal. A distance $\delta^{Q_j p_r}$ between a query Q_j and a prototype p_r is mapped from F to FD as a point $d^{Q_j p_r}$:

$$d^{Q_j p_r} = \{\delta f_i^{Q_j p_r}\}_{i=1}^{t} \tag{4}$$

where, $\delta^{Q_j p_r}$ is represented by the distance from the origin point to $d^{Q_j p_r}$. Here, the impact of every individual feature on the signature dissimilarities is clear. It is obvious that f_2 is more discriminative than f_1. For all genuine query samples like Q_1, $\delta f_2^{Q_1 p_r} < \delta_2$ and for all forgery query samples like Q_2 and Q_3, $\delta f_2^{Q_j p_r} > \delta_2$. On the other hand, f_1 is less discriminant. For the forgery query Q_2, $\delta f_1^{Q_2 p_1} < \delta_1$, same as that for the genuine sample Q_1. Accordingly, it is easier to rank and select features in the FD space, as the impact of the individual features on the overall dissimilarity is clear in this space. Moreover, the multi-class problem with few training samples per class in F space is transformed to a two-class problem in FD space, with more training samples per class.

Ranking and selecting the most discriminant features in the FD space, produces encoding/decoding messages with low dissimilarities between intra-personal instances and with high dissimilarities between inter-personal instances.

However, some of the intrinsic fuzziness of the signature signal will not be canceled through this feature selection approach. To alleviate that, we propose an adaptive distance measure that is computed in the DF space, and absorbs some of the residual fuzziness. For a feature representation $F = \{f_i\}_{i=1}^{t}$, the feature dissimilarity vector $\Delta = \{\delta_i\}_{i=1}^{t}$ is learnt in FD space, where δ_i discriminates between the intra-personal and the inter-personal dissimilarities for a feature f_i. Based on this modeled dissimilarity, we replace the Euclidean distance measure ($\delta^{Q_j p_r}$) by an adaptive dissimilarity measure:

$$D^{Q_j p_r} = \sum_{i=1}^{t}(D_i^{Q_j p_r}), \quad where \ D_i^{Q_j p_r} = \begin{cases} 0 & if \ (\delta f_i^{Q_j p_r} < \delta_i) \\ 1 & otherwise \end{cases} \tag{5}$$

Employing this adaptive distance measure absorbs some of the intrinsic feature variability and increases its discriminative power. For instance, according to Eq.5, distances among the genuine query and its prototypes $D^{Q_1 p_r} = 0$. Moreover, most of the distances between the unauthorized queries and the genuine prototypes $D^{Q_j p_r} = 2$, $\vee j \in [2, 3]$. Hence, some of the variability of the dissimilarity values is canceled.

Ranking the features $\{f_i\}_{i=1}^{t}$ and learning the dissimilarity vector $\{\delta_i\}_{i=1}^{t}$ in the FD space is a general approach, that can be achieved by employing different feature selection methods. However in this paper, this concept is realized by employing a two-step boosting feature selection (BFS) method [12], for fast searching in high dimensional spaces. Decision-stumps (DS) [19], that are single-split single-level classification trees, are trained through a boosting process [18]. Training of a DS is equivalent to selection of a single feature that discriminats between two classes based on a splitting threshold. If the BFS runs in the FD space, a DS_i at a learning iteration i, locates the best dissimilarity feature δf_i, that splits the two classes around a splitting dissimilarity threshold δ_i.

In the first step, a development database ($DevDB$) containing samples of simulated users, is used for training. The reason is that the signature samples of real users are not enough for feature selection in high dimensional spaces. Then, population-based representation is produced by running a BFS process in a DF space, generated by multi-feature representations extracted from the $DevDB$ database. This approach is employed by Rivard et al., to design a writer-independent (WI) classical offline signature verification system [11]. However, the produced population-based spaces have high dimensionality. This is not suitable for encoding bio-cryptographic systems, as the encoding/decoding messages should be concise.

In the second step, the exploitation database ($ExpDB$), containing samples of the real users, is used for training. Signature samples are represented in the population-based space defined through the first step, and additional BFS process runs in this user-based space. Recently, we employed this approach to adapt WI systems to specific writers [13]. Reliable writer-dependent (WD) systems are achieved based on concise and discriminative user-based feature spaces. In this paper, a similar two-step BFS process is employed, however, the user-based BFS

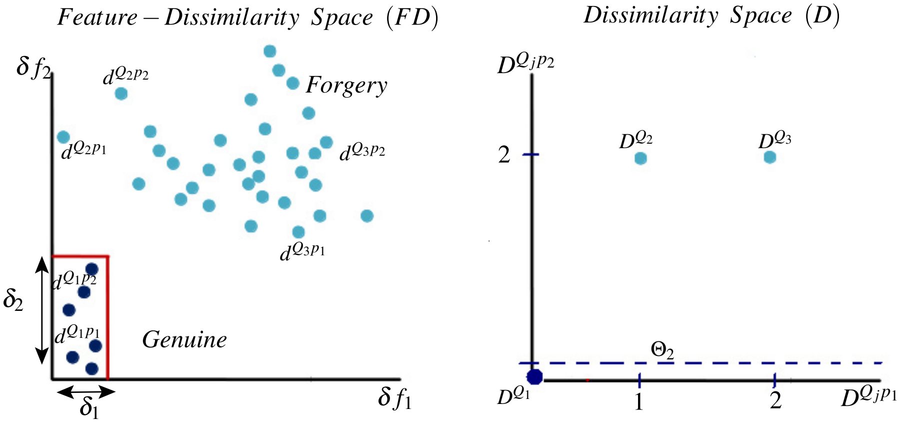


Fig. 2. Illustration of the transformation from the feature-dissimilarity space (left) to the dissimilarity space (right)

step is employed in a FD space, in order to model the feature dissimilarity vector $\varDelta = \{\delta_i\}_{i=1}^{t}$.

3.2 Prototype Selection and Dissimilarity Threshold Optimization

The aforementioned approach enlarges the separation between the dissimilarity distributions of the genuine and impostor encoding messages. However, the distributions differ based on the prototype used for the dissimilarity computations (Eq.5). To get the best possible dissimilarity representation, we propose a prototype selection method.

To this end, the user-based representation, produced through the two-step BFS process, is projected from the FD space to a dissimilarity space D. Consider the available set of R prototypes $P = \{p_1, p_2, ..., p_R\}$. The adaptive dissimilarity distance for a query Q_j is computed for every prototype $p_r \in P$, according to Eq.5. This operation produces a dissimilarity vector D^{Q_j} in the dissimilarity space, where

$$D^{Q_j} = \{D^{Q_jp_1}, D^{Q_jp_2}, ..., D^{Q_jp_R}\}. \tag{6}$$

Figure 2 illustrates the transformation between the FD and D spaces. In the left side, distances between prototype and query messages are represented in the FD space. It is obvious that different prototypes produce different distance values, where significant variability exists for the genuine and the forgery classes. Also, in this space, it is not clear which prototype is the most informative. For space D shown in the right figure, it is obvious that some variability is absorbed through employing the adaptive dissimilarity measure. For instance, $D^{Q_j} = \{0, 0\}$ for all genuine queries (see Eq.5 and Eq.6), as feature dissimilarities $\delta f_1^{Q_j p_r} < \delta_1$ and $\delta f_2^{Q_j p_r} < \delta_2$, for the genuine queries. Also, for most of the forgery queries, $D^{Q_j} = \{2, 2\}$, as $\delta f_1^{Q_j p_r} > \delta_1$ and $\delta f_2^{Q_j p_r} > \delta_2$ for the forgery queries.

Moreover, the dissimilarity space representation provides easier way to rank prototypes according to their discriminative power. For instance, p_2 is more discriminative than p_1, as for all forgery queries, $D^{Q_j p_2} = 2$. While for Q_2, $D^{Q_2 p_1} = 1$ (as $\delta f_1^{Q_2 p_1} < \delta_1$). So, measuring the dissimilarity relative to p_2 results in more isolated clusters.

Finally, in the D space, we optimize the dissimilarity threshold (Θ). In the illustrated example, if the selected prototype is p_2, then any $\Theta_2 < 2$ is discriminant. For p_1, any $\Theta_1 < 1$ is discriminant. Selection of prototypes with higher margin between clusters, provides wider range for selecting the dissimilarity threshold Θ. This results in more flexibility for parameter setting of the bio-cryptographic decoder and hence, higher security and recognition accuracy can be achieved [16].

Based on the proposed method, the decoding functionality DF formulated by Eq.1 can be reformulated as:

$$DF_r(Q_j) = sign(\Theta_r - D^{Q_j p_r}). \tag{7}$$

where r is the index of the selected prototype p_r, Q_j is the query encoding message, Θ_r is the dissimilarity threshold associated with this prototype, and $D^{Q_j p_r}$ is the dissimilarity value computed according to Eq. 5.

The prototype selection method can be realized by various feature selection techniques (with considering prototypes as features), however, we realized it through employing the BFS approach [12].

4 Experimental Methodology

4.1 Database

The Brazilian database [9] is used for proof-of-concept simulations. It contains 7,920 samples of signatures that were digitized as 8-bit grayscale images over 400X1000 pixels at resolution of 300 dpi. This DB contains three types of signature forgery: random, simple and simulated. Random forgeries do not know neither the signerś name nor the signature morphology. It can also happen when a genuine signature presented to the system is mislabeled to another user. For simple forgery, the forger knows the writerś name but not the signature morphology. He can only produce a simple forgery using a style of writing of his liking. Simulated forgeries have access to a sample of the signature. A forger can therefore imitate the genuine signature.

The signatures were provided by 168 writers and are organized as follows: the first 60 writers have 40 genuine signatures, 10 simple forgeries and 10 simulated forgeries per writer, and the other 108 have only 40 genuine signatures per writer. The experimental database is split into two sets: a development dataset ($DevDB$) composed of the last 108 writers, and an exploitation dataset ($ExpDB$) composed of the first 60 writers. Set $DevDB$ is used for the population-based BFS step as illustrated in Section. 3.1.

Set $ExpDB$ is split into two subsets: the reference subset (R) contains the first 30 genuine signatures, and the query subset (Q) contains the rest 10 genuine samples, 10 simple and 10 simulated forgeries. The subset R is used for the user-based BFS step as illustrated in Section. 3.1, and for the prototype selection and dissimilarity threshold optimization as illustrated in Section. 3.2. Both subsets of $ExpDB$ are used for evaluating the method performance.

4.2 Feature Extraction

Extended-Shadow-Code (ESC) [20], and Directional Probability Density Function (DPDF) [21] are employed. Features are extracted based on different grid scales, hence a range of details are detected in the signature image. A set of 30 grid scales is used for each feature type, producing 60 different single scale feature representations. These representations are then fused to produce a feature representation of huge dimensionality $(30, 201)$ [11].

4.3 Design of Encoding Messages and Dissimilarity Measure

The two-step BFS process is implemented as illustrated in section 3.1. First, the ($DevDB$) is used for the population-based BFS phase. We followed the same experimental settings as in the system in [11]. This phase produced a population-based representation (PR) of dimensionality $L = 555$. Second, the reference subset (R) is used for the user-based BFS phase. For each user in $ExpDB$, the signatures in R are used to represent the genuine class, and some signatures from the $DevDB$ are used to represent the forgery class. Then, signatures of both classes are represented in the PR space of L dimensionality. This representation is then transformed to the FD space, where the user-based BFS step runs for t boosting iterations. The process outputs the message elements $\{f_i\}_{i=1}^{20}$, along with their dissimilarities $\Delta = \{\delta_i\}_{i=1}^{20}$, that are used for computing the adaptive dissimilarity measure defined by Eq. 5.

4.4 Prototype Selection and Dissimilarity Threshold Optimization

The thirty signatures in the reference subset (R) are used as a prototype set $P = \{p_r\}_{r=1}^{30}$. To constitute the dissimilarity space D, the adaptive dissimilarity value is computed for every signature in R against all of the thirty signatures (Eq.6). To constitute the forgery class, samples from $DevDB$ are chosen randomly, and dissimilarities between them and the prototypes are computed. BFS runs in this dissimilarity space, to select the best prototype of p_r with the associated threshold Θ_r.

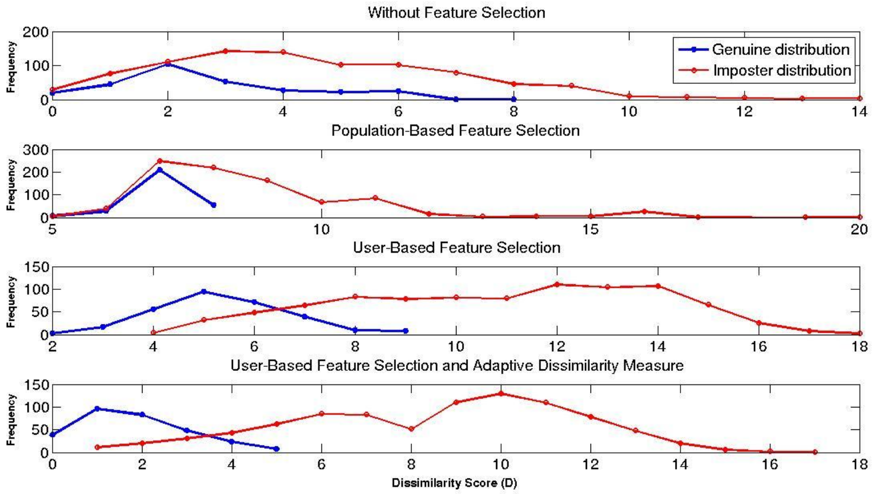


Fig. 3. Dissimilarity score distribution for a specific user

4.5 Performance Measures

To assess the impact of the proposed dissimilarity representation approach on the separability of the genuine and impostor clusters, we use the Hellinger distance. Assuming normal distributions G and I for the genuine and impostor classes, respectively. the squared Hellinger distance between them is give by:

$$H^2(G,I) = 1 - \sqrt{\frac{2\sigma_1\sigma_2}{\sigma_1^2+\sigma_2^2}}\, e^{-\frac{1}{4}\frac{(\mu_1-\mu_2)^2}{\sigma_1^2+\sigma_2^2}} . \tag{8}$$

where, μ_1, μ_2 and σ_1, σ_2 are the mean and variance values for G and I, respectively.

To measure the clusters separability for the different types of forgeries, we report H_{random}, H_{simple} and $H_{simulated}$, where the parameters μ and σ of the impostor cluster I are computed each time, based on the dissimilarities against samples of a specific type of forgeries. Also, we report H_{all}, where the distribution parameters are computed according to dissimilarities of all forgery types.

Also, as the recognition accuracy of bio-cryptographic decoders relies on the dissimilarity ranges separability and on the employed dissimilarity threshold, we measure the recognition errors for all of the dissimilarity scores and use them to generate ROC curves. A ROC curve plots the False Accept Rate (FAR) against the Genuine Accept Rate (GAR) for all possible thresholds (all generated dissimilarity scores). FAR for a specific threshold is the ratio of forgery samples with a dissimilarity score smaller than this threshold. GAR is the ratio of genuine samples with a dissimilarity score smaller than the threshold.

In order to have a global assessment on the quality of encoding messages representation, we compute and average the area under the ROC curves (AUC), for all users in the *ExpDB* subset. High AUC indicates more separation between the dissimilarity score distributions for the genuine and impostor classes.

To assess the impact of the prototype and threshold selection step, we compute the recognition rates. Decoder outputs are estimated by employing Eq. 7 for the selected prototypes and thresholds. By comparing the decoder outputs to the actual class labels, we compute the average error rate (AER_{all}), where

$$AER_{all} = (FRR + FAR_{random} + FAR_{simple} + FAR_{simulated})/4 \qquad (9)$$

False Reject Rate (FRR) is the ratio of genuine queries that produce '0' decoding outputs, FAR_{random}, FAR_{simple} and $FAR_{simulated}$ are the ratio of random, simple, and simulated forgeries respectively that produce '1' decoding outputs. The error rates are also computed when no prototype selection step is employed and for a fixed threshold $\Theta = 6$. [2]

5 Experimental Results

The power of the proposed method for designing the encoding messages and employing the adaptive dissimilarity measure is assessed by its impact on the separability of the genuine and impostor dissimilarity distributions. Figure 3 illustrates the impact of each step of the proposed method for a specific user of the *ExpDB* dataset. It is obvious that, when no feature selection is employed to constitute the encoding message, the genuine and impostor distributions are overlapped. Running BFS based on population signature samples increases the separation between the two distribution. Running the user-based BFS step enhanced the separability. Employing the adaptive distance measure, increased the stability of the genuine class. For instance, the maximum dissimilarity score for the genuine class is decreased from 9 to 5. However, this impact differs for the different forgery types. For instance, in Figure 4, it is clear that while the random forgery class distribution is significantly separated, the simulated forgery distribution still has significant class overlap.

To asses the average performance of the proposed method, the average Hellinger distance is computed over the 60 Users, and for the different types of forgeries. Table 1 shows the results of this analysis. It is obvious that each processing step increased the distances between the genuine and impostor distributions, for all types of forgeries. Average distance of the all forgeries distributions H_{all} is increased from 0.2496 to 0.6617. Also, the average AUC is increased by about 47% (from 0.6577 to 0.9700).

The dissimilarity scores reported above are averaged for all prototypes in the subset R. However, class separation differs for the different prototypes. For instance, Figure 5 shows distributions of the best and worst prototypes for a

[2] $\Theta = 6$ is equivalent to encoding a crypto-key of $128 - bits$ by a biometric message of length $t = 20$, by implementing the FV key-binding scheme [3]. Also, for technical issues, the message elements $\{f_i\}_{i=1}^{t}$ are quantized in 8-bit words before computing the dissimilarities.

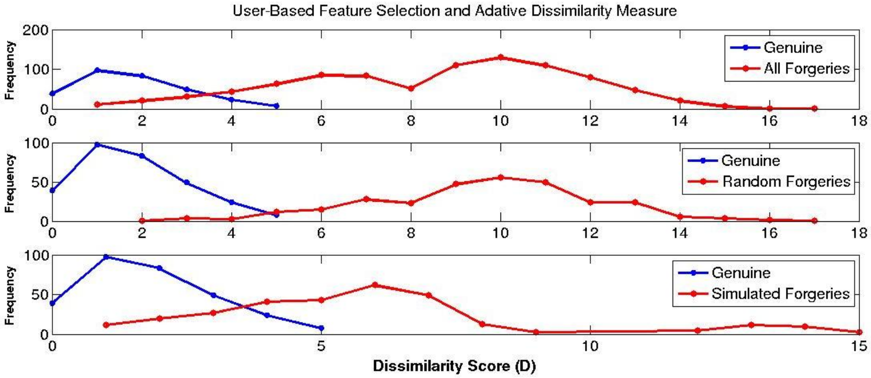


Fig. 4. Dissimilarity score distribution for different forgery types

Table 1. Average Hellinger distance over all Users for the different design scenarios

Design Aspect	Without Feature Selection	Population-based Feature Selection	User-based Feature Selection	User-based Feature Selection with Adaptive Distance Measure
Average H_{random}	0.2976	0.6093	0.6617	0.7398
Average H_{simple}	0.2519	0.5531	0.6011	0.6951
Average $H_{simulated}$	0.1466	0.4395	0.4786	0.5907
Average H_{all}	0.2496	0.5590	0.5923	0.6617
Average AUC	0.6577	0.7724	0.9328	0.9700

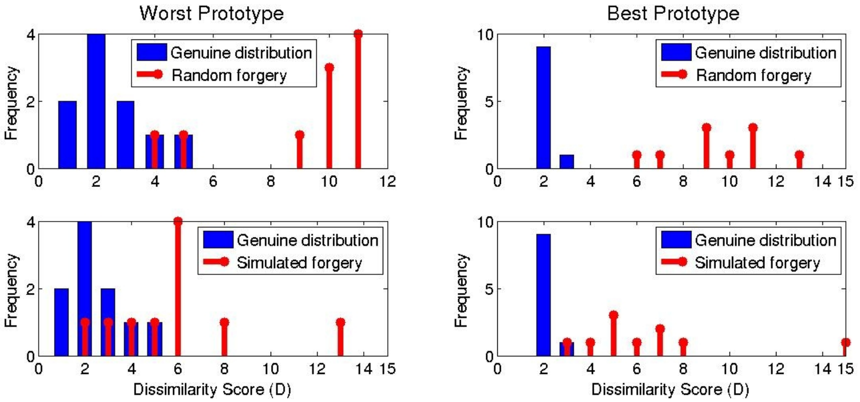


Fig. 5. Dissimilarity score distributions for different prototypes

Table 2. Impact of the Prototype Selection on Average Error Rate over all Users

Design Aspect	Without Prototype Selection	With Prototype Selection
Average FRR	5.25	4.83
Average FAR_{random}	2.74	0.6
Average FAR_{simple}	3.49	1.5
Average $FAR_{simulated}$	33.14	22.33
Average AER	11.15	7.32

specific user. For the worst prototype, a dissimilarity threshold $\Theta = 4$ results in $FRR = 10\%$, $FAR_{random} = 10\%$ and $FAR_{simulated} = 30\%$. For the best prototype, $FRR = 0\%$, $FAR_{random} = 0\%$ and $FAR_{simulated} = 20\%$.

The overall impact of running the prototype selection and threshold optimization step is investigated by computed the recognition error rates for both cases. Tabel 2 shows that AER is decreased by about 34% (from 11.15% to 7.32%), through employing this selection step.

6 Conclusions and Future Work

In this paper, a methodology for designing bio-cryptographic systems based on the dissimilarity representation approach, is proposed. Separation between genuine and impostor distributions is increased through maximizing the distance between the individual elements of the encoding messages. Some of the intrinsic variability of the messages is absorbed by employing an adaptive dissimilarity measure. A prototype selection and dissimilarity threshold optimization method is proposed, to enhance the recognition performance. Future work will employ the proposed method to build a complete signature-based bio-cryptographic system.

Acknowledgments. This work was supported by the Natural Sciences and Engineering Research Council of Canada and BancTec Inc.

References

1. Uludag, U., Pankanti, S., Prabhakar, S., Jain, A.K.: Biometric Cryptosystems: Issues and Challenges. Proceedings of the IEEE 92(6), 948–960 (2004)
2. Juels, A., Wattenberg, M.: A Fuzzy Commitment scheme. In: Sixth ACM Conference on Computer and Communications Security, pp. 28–36. ACM Press (1999)
3. Juels, A., Sudan, M.: A Fuzzy Vault scheme. In: Proc. IEEE Int. Symp. Inf. Theory, Switzerland, p. 408 (2002)
4. Berlekamp, E.R.: Algebraic Coding Theory. McGraw-Hill, New York (1968)
5. Impedovo, D., Pirlo, G.: Automatic signature verification: the state of the art. IEEE Transactions on SMC, Part C: Applications and Reviews 38(5), 609–635 (2008)

6. Pekalska, E., Duin, R.P.W.: Dissimilarity representations allow for building good classifiers. PR Letters 23(8), 161–166 (2002)
7. Duin, R.P.W., Loog, M., Pȩkalska, E.z., Tax, D.M.J.: Feature-Based Dissimilarity Space Classification. In: Ünay, D., Çataltepe, Z., Aksoy, S. (eds.) ICPR 2010. LNCS, vol. 6388, pp. 46–55. Springer, Heidelberg (2010)
8. Pekalska, E., Duin, R.P.W., Paclik, P.: Prototype selection for dissimilarity-based classifiers. PR 39, 189–208 (2006)
9. Freitas, C., Morita, M., Oliveira, L., Justino, E., Yacoubi, A., Lethelier, E., Bortolozzi, F., Sabourin, R.: Bases de dados de cheques bancarios brasileiros. In: XXVI Conferencia Latinoamericana de Informatica, Mexico (2000)
10. Dimauro, G., Guerriero, A., Impedovo, S., Pirlo, G., Salzo, A., Sarcinella, L.: Selection of reference signatures for automatic signature verification. In: Proceedings of the Fifth International Conference on Document Analysis and Recognition (ICDAR 1999), pp. 597–600 (1999)
11. Rivard, D., Granger, E., Sabourin, R.: Multi-Feature extraction and selection in writer-independent offline signature verification. IJDAR 16(1), 83–103 (2013)
12. Tieu, K., Viola, P.: Boosting image retrieval. International Journal of Computer Vision 56(1), 17–36 (2004)
13. Eskander, G.S., Sabourin, R., Granger, E.: Adaptation of writer-independent systems for offline signature verification. In: The 13th International Conference on Frontiers in Handwriting Recognition (ICFHR-2012), Bari, Italy, pp. 432–437 (2012)
14. Freire-Santos, M., Fierrez-Aguilar, J., Ortega-Garcia, J.: Cryptographic key generation using handwritten signatures. In: Proc. of SPIE, vol. 6202, pp. 225–231 (2006)
15. Freire-Santos, M., Fierrez-Aguilar, J., Martinez-Diaz, M., Ortega-Garcia, J.: On the applicability of off-line signatures to the Fuzzy Vault construction. In: Proc. of ICDAR 2007, Curitiba, Brazil (2007)
16. Nandakumar, K., Jain, A.K., Pankanti, S.: Fingerprint based Fuzzy Vault: Implementation and Performance. IEEE Transactions on IFS 2(4), 744–757 (2007)
17. Eskander, G.S., Sabourin, R., Granger, E.: Signature based Fuzzy Vaults with boosted feature selection. In: IEEE Workshop on Computational Intelligence and Identity Management (SSCI-CIBIM 2011), Paris, pp. 131–138 (2011)
18. Schapire, R.: The boosting approach to machine learning: An overview. In: Proc. MSRI Workshop on Nonlinear Estimation and Classification (2002)
19. Iba, W., Langley, P.: Induction of one-level decision trees. In: Proc. of the Ninth International Machine Learning Conference, Scotland, pp. 233–240 (1992)
20. Sabourin, R., Genest, G.: An Extended-Shadow-Code based Approach for Off-Line Signature Verification. In: Proc. of the 12th International Conference on PR, Jerusalem, vol. 2, pp. 450–453 (1994)
21. Drouhard, J., Sabourin, R., Godbout, M.: A neural network approach to off-line signature verification using directional pdf. PR 29(3), 415–424 (1996)

A Repeated Local Search Algorithm for BiClustering of Gene Expression Data

Duy Tin Truong, Roberto Battiti, and Mauro Brunato

University of Trento, Italy
{truong,battiti,brunato}@disi.unitn.it

Abstract. Given a gene expression data matrix where each cell is the expression level of a gene under a certain condition, *biclustering* is the problem of searching for a *subset* of genes that coregulate and coexpress only under a *subset* of conditions. The traditional clustering algorithms cannot be applied for biclustering as one cannot measure the similarity between genes (or rows) and conditions (or columns) by normal geometric similarities. Identifying a network of collaborating genes and a subset of experimental conditions which activate the specific network is a crucial part of the problem. In this paper, the BIClustering problem is solved through a REpeated Local Search algorithm, called **BICRELS**. The experiments on real datasets show that our algorithm is not only fast but it also significantly outperforms other state-of-the-art algorithms.

Keywords: biclustering, co-clustering, local search, gene expression, microarray.

1 Introduction

Gene expression is the process by which information from a gene is used in the synthesis of proteins. Microarray experiments provide us with the expression level of a large number of genes under different experimental conditions [2]. The conditions can be different time points, different types of tissues, or individuals, etc. The gene expression results are presented as a matrix where each gene is a row and each condition is a column. Each cell of the data matrix is the expression level of a gene under a certain condition. The expression level measures the relative abundance of a gene, usually as the logarithmic ratio between the intensities of the dyes used in the experimental process.

Given the gene expression data, one would like to find a *subset* of genes that coregulate and coexpress (think "behave in a coherent manner") only for a *subset* of conditions. This problem is called *biclustering* by Cheng and Church [4]. The objective is to find sub-matrices, i.e. maximal subgroups of genes and subgroups of conditions where the genes exhibit highly correlated activities over a range of conditions, and therefore often related to an underlying *gene regulatory network* of biological interest.

Biclustering is also known as co-clustering, bidimensional clustering, or subspace clustering, and used in other areas like marketing and collaborative

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 281–296, 2013.

recommendations, although with different underlying models. In the traditional clustering problem, only rows or columns of a data matrix are partitioned in different groups based on some geometric similarity measures like the *cosine similarity*, or *Euclidean* distance. Meanwhile, in the biclustering problem, both rows and columns are clustered simultaneously and the identification of a relevant subset of genes and a subset of experimental conditions is a prerequisite to obtain a clustering of biological interest. Traditional clustering algorithms cannot be applied for biclustering as they cannot be based on Euclidean or other geometric properties. This raises the need of developing a new class of algorithms for biclustering, which aim at identifying a relevant network of cooperating genes.

Several algorithms have been proposed for solving the biclustering problem [8]. The algorithms can return a single large and "coherent" bicluster or a set of biclusters. The bicluster "coherence" must be related to experimental process used to identify biological gene regulatory networks. An additive model of the gene expression is often used: the expression of a gene in a network is proportional to the sum of a term associated to the specific gene and a term associated to the specific experimental condition. The squared error w.r.t. this linear model, averaged over the entire bicluster expression levels, called *mean squared residue*, measures the lack of "coherence" of the network [4].

In this paper, we consider the problem of searching for a largest bicluster under the constraint that the *mean squared residue* is below a threshold. A bicluster with mean squared residue less than or equal to a threshold δ is called a δ-bicluster. The problem of finding the largest δ-bicluster is NP-hard [4].

We introduce a Repeated Local Search algorithm for BIClustering (abbreviated as **BICRELS**). Our algorithm is not only reasonably fast due to an incremental evaluation scheme, but it also significantly outperforms other state-of-the-art algorithms in both objectives, leading to larger biclusters with smaller residues.

The rest of this paper is organized as follows. In Section 2, we summarize the related work. Then, we describe formally the biclustering problem in Section 3 and our algorithm in Section 4. Finally, we report on the experimental results in Section 5.

2 Related Work

The algorithms proposed for solving the biclustering problem can be classified into different groups:

- Iterative row and column clustering combination: applying the standard clustering methods on rows and columns of the data matrix and then combining the row and column clusters to form biclusters [5].
- Divide and conquer: breaking the problem into smaller problems, solving them recursively, and combining the solutions of sub-problems to form the solution for the original problem [6].
- Greedy iterative search: removing rows or columns to reduce the bicluster residue below the threshold and adding rows or columns to increase the bicluster volume while the constraint on residue is still satisfied [4].

- Exhaustive bicluster enumeration: enumerating all possible biclusters to identify the best ones in exponential time [10].
- Distribution parameter identification: assuming the data is generated from a model and trying to fit parameters of that model by minimizing a certain criterion [7].

Not all algorithms optimize the same "coherence" criterion, therefore we only compare our algorithm with those based on the same *additive mode* and searching for the largest bicluster with residue below a threshold.

One of the first biclustering algorithms searching for the largest δ-bicluster is proposed by Cheng and Church [4]. The algorithm starts from the initial bicluster which contains all genes and conditions. Each gene or condition is considered as a node. The method iteratively deletes a set of nodes until the mean squared residue of that bicluster below the threshold. Then, a set of nodes is added to the bicluster to increase its volume until any further addition would cause the residue to exceed the threshold. This algorithm is deterministic and very fast, as a set of nodes can be deleted or added at the same time. Its complexity is $O(MN)$ where M and N are the number of genes and conditions. However, modifying a set of nodes simultaneously can also make the algorithm stuck in local minima. We refer this algorithm as **ChengChurch** in this paper.

Yang et al.[12] propose a probabilistic algorithm named **FLOC** (Flexible Overlapped Clusters) which can discover a set of K biclusters in one run. The algorithm starts from a set of random initial biclusters. Each initial bicluster is formed by selecting randomly a subset of rows and a subset of columns from the dataset, such that the bicluster residue is below a predefined threshold. Then, it iteratively performs the best action for each row and column to improve the bicluster quality. The actions are deleting or adding a row or a column to one of K biclusters. The best action is the one that gives the highest improvement in a gain function which is the sum of the reduction ratio in mean squared residue and the increase ratio in volume. As two objectives (volume and residue) are considered at the same time, **FLOC** can return biclusters with very small volume while their residues are much lower than the threshold. Besides, **FLOC** is very sensitive to the initial biclusters and its complexity is $O((M+N)^2 \times K \times p)$ where M, and N are the number of genes and conditions, K is the number of biclusters, and p is the number of iterations the algorithm runs until convergence.

Bleuler et al.[3] introduce a single-objective genetic framework for solving the biclustering problem. In their framework, a bicluster is presented as a binary string with the length of $M + N$ where M and N are the number of genes and conditions, respectively. Normal uniform crossover and bit mutation operators are performed on the population. The minimized objective is the inverse of the bicluster volume if the residue is below the threshold. Otherwise, the algorithm minimizes the residue. However, the authors conclude that without the help of local searchers, the genetic algorithm cannot produce the bicluster which is larger than the one returned by the **ChengChurch** algorithm. Therefore, they hybridize the genetic algorithm with the local search algorithm of Cheng and Church, i.e., each instance in the population is improved by the local searcher before moving to the next generation. We denote this algorithm as **SOGA** (Single Objective Genetic Algorithm). The complexity of **SOGA** is $O(TPMN)$ where T is the number of generations, P is the population size, M and N are the number of genes and conditions, respectively.

Mitra et al.[9] also propose a multi-objective genetic algorithm for biclustering (denoted as **MOGA** in this paper). The authors focus on searching for the largest bicluster and present each bicluster as a binary string. **MOGA** maximizes the volume and the residue simultaneously. When a bicluster has the residue exceeding the threshold, its residue is set to zero. In other words, the quality of a solution violating constraints is considered as zero. Similarly to the case of **SOGA**, the **ChengChuch** algorithm is adapted as the local searcher for **MOGA**, i.e. each instance in the population is improved by the local searcher before moving to the next generation. The complexity of **MOGA** is $O(TP^2MN)$ where T is the number of generations, P is the population size, M and N are the number of genes and conditions, respectively.

3 The Biclustering Problem

We follow the same notation used in [4]. The biological motivation for the model is that, in a gene regulatory network, the gene expression level is proportional to a sum of a term characterizing the gene plus a term characterizing the experimental condition which is activating the specific network. Let's note that, if logarithms of the original measures are taken, the model is multiplicative in the original measures. Fig.1 shows an example of a bicluster with 9 genes and 6 conditions.

Let X be the set of genes and Y be the set of conditions. Let a_{ij} be the element of the expression matrix A representing the expression level of the gene i under

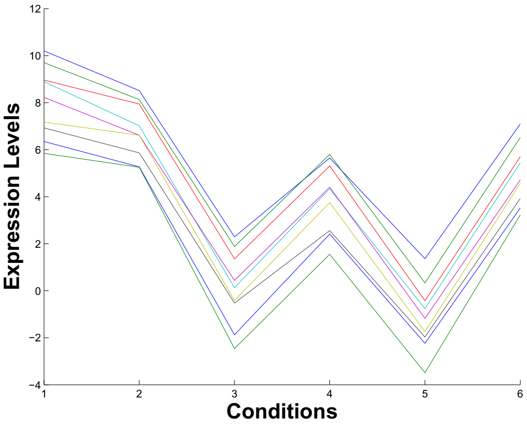


Fig. 1. An example of a bicluster with 9 genes and 6 conditions

condition j. Let $I \subset X$ and $J \subset J$ be subsets of genes and conditions. The pair (I, J) specifies a submatrix A_{IJ} with the following *mean squared residue* score:

$$MSR(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} - a_{IJ})^2 \tag{1}$$

where

$$a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij} \tag{2}$$

$$a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij} \tag{3}$$

$$a_{IJ} = \frac{1}{|I||J|} \sum_{i \in I} \sum_{j \in J} a_{ij} = \frac{1}{|I|} \sum_{i \in I} a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{Ij} \tag{4}$$

The biclustering problem is the problem of searching for a bicluster (I, J) such that its volume is maximized and its mean squared residue is below a threshold. Formally, the biclustering problem is defined as:

$$(I', J') = \underset{I \subset X, J \subset Y}{\operatorname{argmax}} |I||J| \tag{5}$$

subject to

$$MSR(I,J) \leq \delta \tag{6}$$

4 A Repeated Local Search Algorithm for Biclustering

Let $rowMSR(i)$ and $colMSR(j)$ are the mean squared residues of row i and column j w.r.t a bicluster (I, J), respectively.

$$rowMSR(i) = \frac{1}{|J|} \sum_{j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2 \tag{7}$$

$$colMSR(j) = \frac{1}{|I|} \sum_{i \in I} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2 \tag{8}$$

The pseudo code of our local search algorithm for biclustering (**BICRELS**) is shown in Algorithm 1. We first generate the initial set of biclusters by combing gene and condition clusters. In detail, we partition the gene set into 100 clusters by applying **K-Means** (with the *cosine similarity* distance) on the column-normalized data where each gene is an instance, and each condition is a feature. The column-normalized data is obtained from the original data by subtracting the mean value from each column and then dividing the results by its sample standard deviation. Similarly, we divide the conditions into $ceil(N/10)$ clusters on the row-normalized data where N is the number of conditions and $ceil(x)$ returns the nearest integer equal to or greater than x. Then, we pick randomly a

Algorithm 1. BICRELS

```
   Input  : data matrix $A$, residue threshold $\delta$
   Output: A bicluster $(I, J)$
1  begin
2      $pool$ = create a set of initial biclusters.
3      $biclusterSet = \emptyset$
4      for $i = 1$ to $numberOfRestarts$ do
5          $bicluster$ = pick randomly a bicluster from $pool$.
6          Remove $bicluster$ from $pool$.
7          $bicluster = replaceNodes(bicluster)$
8          $bicluster = deleteNodes(bicluster, \delta)$
9          repeat
10             $bicluster = replaceNodes(bicluster)$
11             $bicluster = addNodes(bicluster, \delta)$
12         until no change ;
13         $bicluster = deleteNodes(bicluster, \delta)$
14         $biclusterSet = biclusterSet \cup \{bicluster\}$
15 end
16 return $bicluster \in biclusterSet$ with maximum size
```

cluster of gene and a cluster of condition to form a bicluster. In the experiments of this paper, we create 100 biclusters for the initial bicluster set.

The local search procedure from line 4 to 14 of the algorithm is restarted for a number of runs to explore different local minima. Note that the normalized data is used only to create the initial bicluster set, and the local search procedure is run on the original data. In each run, the algorithm first picks randomly a bicluster from the initial bicluster set. That bicluster is then removed from the initial set. Next, the algorithm reduces the residue of that bicluster by the procedure *replaceNodes* in Algorithm 2. This procedure shrinks the residue of a bicluster by replacing the column (or row) with the highest residue in that bicluster by a column (or row) not in that bicluster with the smallest residue if the replacement can reduce the bicluster residue. The replacement process is repeated until no columns or rows are replaced. After shrinking the bicluster residue by replacing rows or columns, if the residue is still greater than the threshold, some rows or columns are deleted in the *deleteNodes* procedure of Algorithm 3 (which is the single-node deletion procedure proposed by Cheng and Church [4]). The *deleteNodes* procedure keeps deleting the columns and rows with highest mean residues until the residue of the bicluster drops below the threshold. Note that, although both the *replaceNodes* and *deleteNodes* procedure can decrease the residue, the *replaceNodes* procedure keeps the bicluster size unchanged whereas the *deleteNodes* procedure also reduces the bicluster size.

Now, the bicluster residue is guaranteed to be lower than or equal to the threshold. The algorithm starts optimizing the bicluster by repeating two steps: *replaceNodes* and *addNodes* until convergence. The main idea is that while fixing the volume, we try to reduce the residue of the bicluster and then while

Algorithm 2. replaceNodes

```
Input  : $(I, J)$ are the sets of rows and columns
Output: $(I', J')$ with smaller or equal residue
begin
    repeat
        // Replace columns
        repeat
            $maxJ = \operatorname{argmax}_{j \in J} colMSR(j)$
            $minJ = \operatorname{argmin}_{j \in Y \setminus J} colMSR(j)$
            $J' = J \cup \{minJ\} \setminus \{maxJ\}$
            if $MSR(I, J) > MSR(I, J')$ then
                $J = J'$
        until $J$ is not modified ;
        // Replace rows
        repeat
            $maxI = \operatorname{argmax}_{i \in I} rowMSR(i)$
            $minI = \operatorname{argmin}_{i \in X \setminus I} rowMSR(i)$
            $I' = I \cup \{minI\} \setminus \{maxI\}$
            if $MSR(I, J) > MSR(I', J)$ then
                $I = I'$
        until $I$ is not modified ;
    until $I, J$ are not modified ;
end
return $(I, J)$
```

keeping the residue below the threshold, we try to increase the bicluster volume. The *addNodes* procedure is presented in Algorithm 4. This procedure iteratively adds a column or a row with the smallest residue until the residue of the bicluster exceeds the threshold. Finally, to guarantee that the bicluster mean squared residue is less than or equal to the threshold, we perform the *deleteNodes* procedure before returning the bicluster. As the number of rows and columns is finite, the loop of two steps *replaceNodes* and *addNodes* always terminates after a finite number of iterations (which is less than or equal to $(|X| + |Y|)$).

Algorithm Complexity. The most expensive steps in three procedures *replaceNodes*, *deleteNodes*, and *addNodes* are the computation of $MSR(I, J)$ and all $rowMSR(i)$, $colMSR(j)$ which have the complexity of $O(|X||Y|)$ where $|X|$ is the number of rows, and $|Y|$ is the number of columns. Therefore, the complexity of these three procedures as well as the whole algorithm is also $O(|X||Y|)$. However, as each iteration, only one row or column is modified, the incremental update strategy can be applied to reduce the complexity of computing $MSR(I, J)$ from $O(|X||Y|)$ to $O(max(|X|, |Y|))$. Besides, the cost of updating a_{iJ}, a_{Ij}, a_{IJ}

Algorithm 3. deleteNodes

```
Input  : $(I, J)$ are the sets of rows and columns, threshold $\delta$
Output: $(I', J')$ with residue smaller than threshold $\delta$
begin
    while $MSR(I, J) > \delta$ do
        $maxJ = \operatorname{argmax}_{j \in J} colMSR(j)$
        $maxI = \operatorname{argmax}_{i \in I} rowMSR(i)$
        if $colMSR(maxJ) > rowMSR(maxI)$ then
            $J = J \setminus \{maxJ\}$
        else
            $I = I \setminus \{maxI\}$
end
return $(I, J)$
```

Algorithm 4. addNodes

```
Input  : $(I, J)$ are the sets of rows and columns, threshold $\delta$
Output: $(I', J')$ with greater or equal size
begin
    while $MSR(I, J) < \delta$ do
        $minJ = \operatorname{argmin}_{j \in Y \setminus J} colMSR(j)$
        $minI = \operatorname{argmin}_{i \in X \setminus I} rowMSR(i)$
        if $colMSR(minJ) < rowMSR(minI)$ then
            $J = J \cup \{minJ\}$
        else
            $I = I \cup \{minI\}$
end
return $I, J$
```

necessary for the computation of all $rowMSR(i)$, $colMSR(j)$ can also be reduced from $O(|X||Y|)$ to $O(max(|X|, |Y|))$.

Incremental Update. As can be seen from Equation 7, before computing the value of $rowMSR(i)$ we need to update the values of a_{iJ}, a_{Ij}, a_{IJ}. There are six cases where a bicluster (I, J) can be modified: add or delete a row or column, replace a row (or a column) by another row (or column).

When we add, delete or replace a row from a bicluster (I, J), all columns $j \in J$ are unchanged, thus all mean rows a_{iJ} (where $i \in I$) are also unaffected. Therefore, we only need to update the mean columns a_{Ij} and the overall average value a_{IJ}. The update procedure for a_{Ij}, a_{IJ} in each case is presented as follows.

a) Adding a row $r \in X \backslash I$ *to the bicluster* (I, J)*:* In this case, besides updating the mean columns a_{Ij} and the overall average value a_{IJ}, we also need to compute the new mean row a_{rJ}:

$$a_{rJ} = \frac{1}{|J|} \sum_{j \in J} a_{rj} \tag{9}$$

$$a_{Ij} = \frac{1}{|I|+1}(a_{Ij} * |I| + a_{rj}) \tag{10}$$

$$a_{IJ} = \frac{1}{|I|+1}(a_{IJ} * |I| + a_{rJ}) \tag{11}$$

Then, we update $I = I \cup \{r\}$. The complexity of computing a_{rJ} is $O(|Y|)$. Because each a_{Ij} is updated with the complexity of $O(1)$, the complexity of updating all a_{Ij} is $O(|Y|)$ (as $J \subset Y$).

b) Deleting a row $r \in I$ *from the bicluster* (I, J)*:* In this case, we only need to update the mean columns a_{Ij} and the overall average value a_{IJ}:

$$a_{Ij} = \frac{1}{|I|-1}(a_{Ij} * |I| - a_{rj}) \tag{12}$$

$$a_{IJ} = \frac{1}{|I|-1}(a_{IJ} * |I| - a_{rJ}) \tag{13}$$

Then, we update $I = I \setminus \{r\}$. Because each a_{Ij} is updated with the complexity of $O(1)$, the complexity of updating all a_{Ij} is $O(|Y|)$ (as $J \subset Y$).

c) Replacing a row $r_1 \in I$ *by a row* $r_2 \in X \setminus I$*:* In this case, besides updating the mean columns a_{Ij} and the overall average value a_{IJ}, we also need to compute the new mean row a_{r_2J}:

$$a_{r_2J} = \frac{1}{|J|} \sum_{j \in J} a_{r_2j} \tag{14}$$

$$a_{Ij} = \frac{1}{|I|}(a_{Ij} * |I| - a_{r_1j} + a_{r_2}j) \tag{15}$$

$$a_{IJ} = \frac{1}{|I|}(a_{IJ} * |I| - a_{r_1J} + a_{r_2J}) \tag{16}$$

Then, we update $I = I \backslash \{r_1\} \cup \{r_2\}$. The complexity of computing a_{r_2J} is $O(|Y|)$. Because each a_{Ij} is updated with the complexity of $O(1)$, the complexity of updating all a_{Ij} is $O(|Y|)$ (as $J \subset Y$).

Similarly, when we add, delete or replace a column from a bicluster (I, J), all rows $i \in I$ are unchanged, thus all mean columns a_{Ij} (where $j \in J$) are also unaffected. Therefore, we only need to update the mean rows a_{iJ} and the overall average value a_{IJ}. The update formulas can be derived similarly as in the cases of rows. The update complexity for all mean rows a_{iJ} is $O(|X|)$. In other words, the complexity of updating a_{iJ}, a_{Ij} and a_{IJ} in all cases is $O(max(|X|, |Y|)$.

In addition, in each iteration of three procedures *replaceNodes*, *deleteNodes*, and *addNodes*, we also need to recompute the mean squared residue of the updated bicluster. Let (I', J') be the updated bicluster obtained from the bicluster (I, J) by applying one of six operations: add, delete or replace rows or columns. To reduce the complexity of computing $MSR(I, J)$ from $O(|X||Y|)$ to $O(max(|X|, |Y|))$, we first derive another way to compute $MSR(I, J)$:

$$MSR(I,J) = \frac{1}{|I||J|}\sum_{i\in I}\sum_{j\in J}(a_{ij} - a_{iJ} - a_{Ij} - a_{IJ})^2 \tag{17}$$

$$= \frac{1}{|I||J|}\big(\sum_{i\in I}\sum_{j\in J}a_{ij}^2 + |I||J|a_{IJ}^2 - |I|\sum_{j\in J}a_{Ij}^2 - |J|\sum_{i\in I}a_{iJ}^2\big) \tag{18}$$

It can be seen that the complexities of four terms in the bracket on the right hand side of Equation 18 are $O(|X||Y|)$, $O(1)$, $O(|Y|)$ and $O(|X|)$, respectively (as $I \subset X, J \subset Y$). However, the first term can be updated efficiently as follows. Let $sumAll(I, J) = \sum_{i\in I}\sum_{j\in J} a_{ij}^2$, we can compute efficiently $sumAll(I', J')$ according to one of six cases:

a) If the updated bicluster (I', J') is obtained by adding a row r to the bicluster (I, J):

$$sumAll(I', J') = sumAll(I \cup \{r\}, J) \tag{19}$$

$$= \sum_{i\in I'}\sum_{j\in J} a_{ij}^2 \tag{20}$$

$$= \sum_{i\in I}\sum_{j\in J} a_{ij}^2 + \sum_{j\in J} a_{rj}^2 \tag{21}$$

$$= sumAll(I, J) + \sum_{j\in J} a_{rj}^2 \tag{22}$$

b) If the updated bicluster (I', J') is obtained by deleting a row r from the bicluster (I, J):

$$sumAll(I', J') = sumAll(I \setminus \{r\}, J) \tag{23}$$

$$= sumAll(I, J) - \sum_{j\in J} a_{rj}^2 \tag{24}$$

c) If the updated bicluster (I', J') is obtained by replacing a row $r_1 \in I$ by a row $r_2 \in X \setminus I$ from the bicluster (I, J):

$$sumAll(I', J') = sumAll(I \setminus \{r_1\} \cup \{r_2\}, J) \tag{25}$$

$$= sumAll(I, J) - \sum_{j\in J} a_{r_1 j}^2 + \sum_{j\in J} a_{r_2 j}^2 \tag{26}$$

d) If the updated bicluster (I', J') is obtained by adding a column c to the bicluster (I, J):

$$sumAll(I', J') = sumAll(I, J \cup \{c\}) \tag{27}$$

$$\tag{28}$$

$$= sumAll(I, J) + \sum_{i \in I} a_{ic}^2 \tag{29}$$

e) If the updated bicluster (I', J') is obtained by removing a column c from the bicluster (I, J):

$$sumAll(I', J') = sumAll(I, J \setminus \{c\}) \tag{30}$$

$$\tag{31}$$

$$= sumAll(I, J) - \sum_{i \in I} a_{ic}^2 \tag{32}$$

f) If the updated bicluster (I', J') is obtained by replacing a column $c_1 \in J$ by a column $c_2 \in Y \setminus J$ from the bicluster (I, J):

$$sumAll(I', J') = sumAll(I, J \setminus \{c_1\} \cup \{c_2\}) \tag{33}$$

$$\tag{34}$$

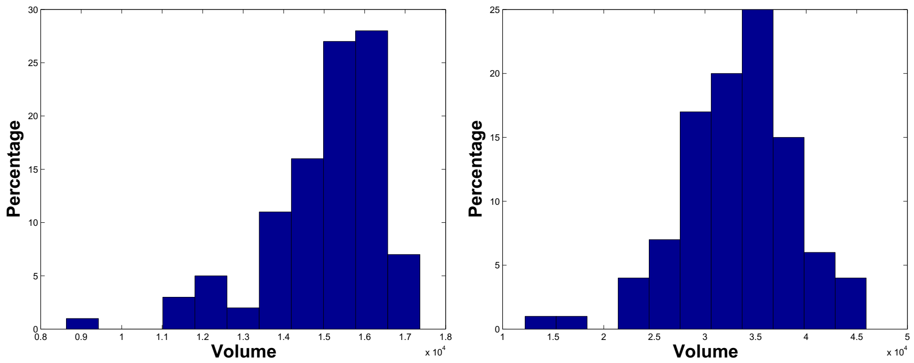
$$= sumAll(I, J) - \sum_{i \in I} a_{ic_1}^2 + \sum_{i \in I} a_{ic_2}^2 \tag{35}$$

In all cases, the update of $sumAll(I', J')$ has the complexity of $O(|X|)$ or $O(|Y|)$. In other words, the complexity of computing $MSR(I', J')$ form $MSR(I, J)$ is $O(max(|X|, |Y|)$.

5 Experiments

A preliminary set of experiments has been dedicated to analyze the distribution of final results found after individual runs of our local search technique. It is the distribution of local maxima values found after starting from a randomly picked initial seed bicluster. To estimate the distribution of bicluster volume obtained by **BICRELS**, we run **BICREL** with 100 restart times and plot the histograms of bicluster volume obtained by the individual local search processes on two datasets in Fig.2a and Fig.2b. In both cases, we observe a non-negligible probability for values of bicluster volumes that are close to the optimal one. This was the main motivation for adding a restart technique: by starting from different initial random seed biclusters the algorithm is sampling from this distribution and the best sample is reported at the end of the repetitions.

Considering now the complete **BICREL**, we compare it with four other state-of-the-art algorithms **ChengChurch** [4], **SOGA** (Single-objective GA) [3], **MOGA** (Multi-objective GA) [9], and **FLOC** [12].



(a) Volume Distribution on the *Yeast* dataset ($\delta = 300$)

(b) Volume Distribution on the *Lymphoma* dataset ($\delta = 1200$)

Fig. 2. Bicluster Volume Distribution of 100 restart times of **BICRELS**

5.1 Experimental Setup

The parameter α of **ChengChurch** is set to its default value ($\alpha = 1.2$). The number of initial rows and columns in the **FLOC** algorithm is set to 9 and 2, respectively (as in the **FLOC** paper [12], the authors did not describe which parameters are suitable for **FLOC**, thus we did some experiments to determine the suitable parameters for **FLOC**). The number of biclusters produced by **FLOC** in each run is set to 10. The parameters of **SOGA** and **MOGA** are set to their default parameters. All algorithms are implemented in Matlab and run on the same machine with Ubuntu 12.04 operating system, CPU Intel(R)Xeon(R)X3363@2.83GHz, RAM 8GiB. **FLOC**, **SOGA** and **MOGA** are run for 10 times to eliminate the randomness effect. **BICRELS** is run once but restarted for 10 times by setting its parameter *numberOfRestarts* to 10. **ChengChurch** is a deterministic algorithm and run only once.

In the experiments, we use two datasets *Yeast* [11] and *Lymphoma* [1]. The *Yeast* dataset consists of 2884 genes and 17 conditions. The *Lymphoma* dataset has 4026 genes and 96 conditions. These datasets are preprocessed by Cheng and Church[1]. Missing values of these datasets are processed as in [4]. The residue threshold of all algorithms on the *Yeast* and *Lymphoma* dataset are set to 300 and 1200 as in previous papers [4,3,9]. However, in some cases, some algorithms like **FLOC** can be stuck in local minima and return biclusters with very small volumes whereas their residues are much lower than the threshold. Therefore, for a fair comparison, we set different residue thresholds in our algorithm to produce biclusters with similar residues as in those of the other algorithms.

[1] `http://arep.med.harvard.edu/biclustering`

5.2 Experimental Results

The maximum volumes of biclusters produced by all algorithms on two datasets *Yeast* and *Lymphoma* are presented in Table 1a, Table 2a, Fig.3a, and Fig.3b.

It can be observed that **BICRELS** significantly outperforms the other algorithms on the two datasets in both objectives (larger in volume and smaller in residue). In addition, although setting the same residue threshold, **FLOC** can get stuck at local minima and can only produce very small biclusters.

Table 1b and Table 2b show the statistical information on the bicluster volume obtained by five algorithms. The average bicluster volume of our algorithm

Table 1. The comparison of five algorithms on the *Yeast* dataset

Algorithm	Max Volume ($\|I\| \times \|J\|$)	MSR
BICRELS ($\delta = 300$)	16577 (1507 × 11)	299.93
ChengChurch ($\delta = 300$)	12012 (1001 × 12)	237.33
BICRELS ($\delta = 237$)	12114 (1346 × 9)	236.95
SOGA ($\delta = 300$)	13050 (1305 × 10)	286.04
BICRELS ($\delta = 286$)	15580 (1558 × 10)	285.97
MOGA ($\delta = 300$)	8480 (848 × 10)	299.12
BICRELS ($\delta = 299$)	16511 (1501 × 11)	298.87
FLOC ($\delta = 300$)	942 (314 × 3)	143.20
BICRELS ($\delta = 143$)	5362 (766 × 7)	142.88

(a) The largest biclusters obtained from five algorithms

Algorithm	Max Volume	Min Volume	Average Volume (± Std)
BICRELS ($\delta = 300$)	16577	11473	15103.80 (±1567.04)
ChengChurch ($\delta = 300$)	12012	12012	12012.00 (±0.00)
BICRELS ($\delta = 237$)	12114	7865	10695.40 (±1470.11)
SOGA ($\delta = 300$)	13050	1443	7745.16 (±2701.19)
BICRELS ($\delta = 286$)	15580	10659	14030.80 (±1434.14)
MOGA ($\delta = 300$)	8480	3520	7271.09 (±803.33)
BICRELS ($\delta = 299$)	16511	11418	15044.00 (±1564.90)
FLOC ($\delta = 300$)	942	484	589.40 (±156.83)
BICRELS ($\delta = 143$)	5362	2709	4345.60 (±938.51)

(b) Statistical information on the bicluster volume of five algorithms

Algorithm	Runtime
BICRELS ($\delta = 300$)	2.26
ChengChurch ($\delta = 300$)	0.12
BICRELS ($\delta = 237$)	1.87
SOGA ($\delta = 300$)	15.08
BICRELS ($\delta = 286$)	2.13
MOGA ($\delta = 300$)	19.52
BICRELS ($\delta = 299$)	2.24
FLOC ($\delta = 300$)	615.85
BICRELS ($\delta = 143$)	1.12

(c) Average runtime (in seconds) of five algorithms

Table 2. The comparison of five algorithms on the *Lymphoma* dataset

Algorithm	Max Volume ($\lvert I\rvert \times \lvert J\rvert$)	MSR
BICRELS ($\delta = 1200$)	43907 (1909×23)	1199.50
ChengChurch ($\delta = 1200$)	39026 (1027×38)	1101.52
BICRELS ($\delta = 1101$)	39307 (1709×23)	1100.56
SOGA ($\delta = 1200$)	35820 (995×36)	1187.24
BICRELS ($\delta = 1187$)	42274 (1838×23)	1186.72
MOGA ($\delta = 1200$)	39032 (1394×28)	1191.62
BICRELS ($\delta = 1191$)	42458 (1846×23)	1190.98
FLOC ($\delta = 1200$)	572 (143×4)	363.22
BICRELS ($\delta = 363$)	6440 (920×7)	362.84

(a) The largest biclusters obtained by five algorithms

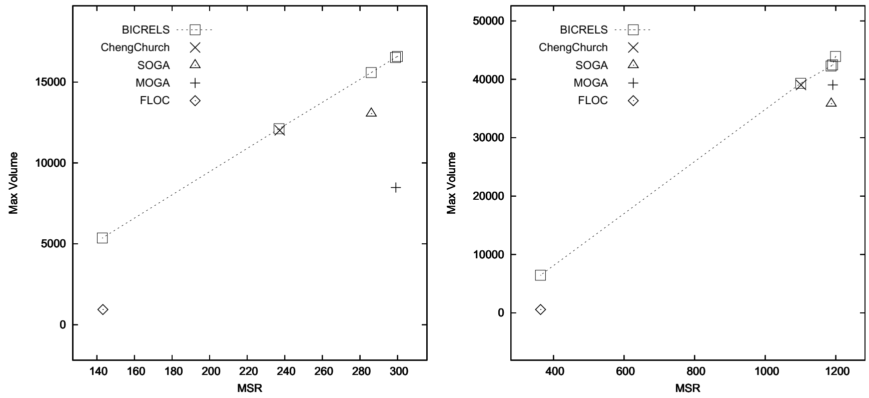
Algorithm	Max Volume	Min Volume	Average Volume (± Std)
BICRELS ($\delta = 1200$)	43907	29887	33932.50 (±4394.52)
ChengChurch ($\delta = 1200$)	39026	39026	39026.00 (±0.00)
BICRELS ($\delta = 1101$)	39307	25968	30064.10 (±4333.37)
SOGA ($\delta = 1200$)	35820	2014	23983.98 (±8441.05)
BICRELS ($\delta = 1187$)	42274	28964	33230.90 (±4191.15)
MOGA ($\delta = 1200$)	39032	30875	35451.80 (±1970.40)
BICRELS ($\delta = 1191$)	42458	29055	33348.60 (±4220.95)
FLOC ($\delta = 1200$)	572	282	354.40 (±112.25)
BICRELS ($\delta = 363$)	6440	1314	3779.00 (±1500.02)

(b) Statistical information on bicluster volume of five algorithms

Algorithm	Runtime
BICRELS ($\delta = 1200$)	13.37
ChengChurch ($\delta = 1200$)	0.34
BICRELS ($\delta = 1101$)	12.73
SOGA ($\delta = 1200$)	76.67
BICRELS ($\delta = 1187$)	13.02
MOGA ($\delta = 1200$)	75.37
BICRELS ($\delta = 1191$)	13.12
FLOC ($\delta = 1200$)	345.93
BICRELS ($\delta = 363$)	2.54

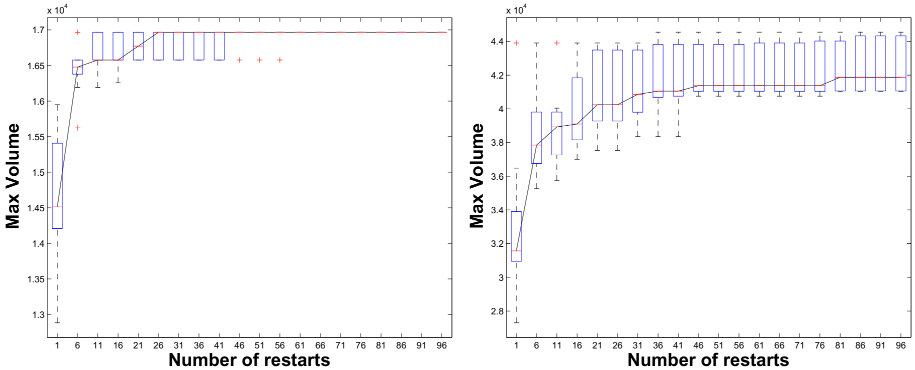
(c) Average runtime (in seconds) of five algorithms

is much larger than that of the other algorithms. In order to study the relationship between the number of restart times and the maximum volume, we run **BICRELS** with 10 different random seeds, and in each run **BICRELS** is restarted for 100 times. Fig.4a and Fig.4b show the performance curves of our algorithm on two datasets. It can be seen that our algorithm reaches very good final results within about 30 restarts. Especially, on the *Yeast* dataset, our algorithm always converges to the optimal solution after 41 restarts. **BICRELS** also enjoys the *anytime* property: it can be terminated at any time after a given number of restarts (greater than zero) delivering the best solution found so far. This characteristic endows it with more flexibility to trade off CPU time w.r.t. solution quality.



(a) On the *Yeast* dataset (b) On the *Lymphoma* dataset

Fig. 3. Performance comparison of five algorithms on two datasets. Let's remind that MSR has to be minimized, while volume has to be maximized. To improve visibility **BICRELS** results are connected by segments.



(a) On the *Yeast* dataset ($\delta = 300$) (b) On the *Lymphoma* dataset ($\delta = 1200$)

Fig. 4. The performance curves of **BICRELS** on two datasets. Each boxplot of 10 different runs with the same number of restarts shows the maximum, minimum volume and lower, upper quartile together with median. The observations which are considered as outliers are presented as red crosses.

As for the comparison on the runtime, except for the **ChengChurch** algorithm, our algorithm is faster than the other algorithms as shown in Table 1c and Table 2c. Our algorithm **BICRELS** is slower than the **ChengChurch** algorithm because in each iteration, **BICRELS** only adds one row or column whereas **ChengChurch** can add a set of rows or columns. The difference between runtime of our algorithm and **ChengChurch** is the computational cost that we pay for the improvement in the solution quality. Because of the anytime property of **BICRELS**, an early termination after a smaller number of restarts is the obvious way to reduce CPU time for a lower average solution quality.

6 Conclusion

In this paper, we proposed a repeated local search algorithm for biclustering, called **BICRELS**. We also suggested an efficient incremental update scheme to speed up the algorithm. Although our algorithm is simple, it is reasonably fast and it significantly outperforms the other state-of-the-art algorithms on two real-world datasets. Finally, as our algorithm has the any-time property, it provides users with the flexibility in trading off CPU time w.r.t. solution quality.

References

1. Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., et al.: Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. Nature 403(6769), 503–511 (2000)
2. Baldi, P., Hatfield, G.W.: DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modelling. Cambridge University Press (2002)
3. Bleuler, S., Prelic, A., Zitzler, E.: An ea framework for biclustering of gene expression data. In: Congress on Evolutionary Computation, CEC 2004, vol. 1, pp. 166–173. IEEE (2004)
4. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, vol. 8, pp. 93–103 (2000)
5. Getz, G., Gal, H., Kela, I., Notterman, D.A., Domany, E.: Coupled two-way clustering analysis of breast cancer and colon cancer gene expression data. Bioinformatics 19(9), 1079–1089 (2003)
6. Hartigan, J.A.: Direct clustering of a data matrix. Journal of the American Statistical Association 67(337), 123–129 (1972)
7. Lazzeroni, L., Owen, A., et al.: Plaid models for gene expression data. Statistica Sinica 12(1), 61–86 (2002)
8. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics 1(1), 24–45 (2004)
9. Mitra, S., Banka, H.: Multi-objective evolutionary biclustering of gene expression data. Pattern Recognition 39(12), 2464–2477 (2006)
10. Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. Bioinformatics 18(suppl. 1), S136–S144 (2002)
11. Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., Church, G.M.: Systematic determination of genetic network architecture. Nature Genetics 22, 281–285 (1999)
12. Yang, J., Wang, H., Wang, W., Yu, P.: Enhanced biclustering on expression data. In: Proceedings of the Third IEEE Symposium on Bioinformatics and Bioengineering, pp. 321–327. IEEE (2003)

Author Index