Project acronym	SIMBAD
Project full title	Beyond Features: Similarity-Based Pattern Analysis and Recognition
Deliverable Responsible	Dep. de Engenharia Electrotécnica e Computadores Instituto Superior Técnico Av. Rovisco Pais, 1 1049-001 Lisboa (Portugal) http://www.ist.utl.pt
Project web site	http://simbad-fp7.eu
EC project officer	Teresa De Martino
Document title	Learning Similarities from Examples
Deliverable n.	D2.3
Document type	Report
Dissemination level	Public
Contractual date of delivery	M 42
Project reference number	213250
Status & version	Definitive version
Work package	WP 2.3
Deliverable responsible	IST
Contributing Partners	UNIVE, DELFT
Author(s)	Ana Fred, Mário Figueiredo
Additional contributor(s)	Marcello Pelillo, Bob Duin

SIMBAD Deliverable D2.3
Learning and Combining Similarities

Ana Fred, Mário Figueiredo

September 18, 2011

Contents

1 Overview

This deliverable summarizes results obtained in the context of work package W2.3, “Learning and Combining Similarities". The text is structured into two main areas:

- Unsupervised learning, where the following sub-areas are addressed: (a) learning and combining similarities from clustering ensembles, under the Evidence Accumulation Clustering (EAC) paradigm; (b) higher order dissimilarities using Dissimilarity Increments (DI).
- Supervised learning, where the key topic is Multiple Kernel Learning (MKL).

The report is organized as follows. We present progress and the main technical achievements, according to the two broad areas referred above, in sections 2 and 3. A more detailed description of the later are found as appendices, formed by a collection of published or submitted papers within the context of the SIMBAD project.

2 Unsupervised Learning

2.1 Learning and Combining Similarities from Evidence Accumulation

There is a close connection between the concepts of pairwise similarity and probability in the context of unsupervised learning. It is a common assumption that, if two objects are similar, it is very likely that they are grouped together by some clustering algorithm, the higher the similarity, the higher the probability of co-occurrence in a cluster. Conversely, if two objects co-occur very often in the same cluster (high co-occurrence probability), then it is very likely that they are very similar. This duality and correspondence between pairwise similarity and pairwise probability within clusters forms the core idea of

the clustering ensemble (CE) approach known as Evidence Accumulation Clustering (EAC) [1].

Each clustering algorithm induces a pair-wise similarity. Evidence accumulation clustering combines the results of multiple clusterings into a single data partition by viewing each clustering result as an independent evidence of pairwise data organization. Using a pairwise frequency count mechanism amongst a clustering committee, the method yields, as an intermediate result, a co-association matrix that summarizes the evidence taken from the several members in the clustering ensemble. This matrix corresponds to the maximum likelihood estimate of the probability of pairs of objects being in the same group, as assessed by the clustering committee, and can be regarded as a pair-wise similarity induced by the CE. One of the main advantages of EAC is that it allows for a big diversification within the clustering committee. Indeed, no assumption is made about the algorithms used to produce the data partitions, it is robust to incomplete information, i.e., we may include partitions over sub-sampled versions of the original data set, and no restriction is made on the number of clusters of the partitions.

The EAC method can be decomposed into three major steps:

1. construction of the clustering ensemble;
2. accumulation of the "clustering evidence" provided by the ensemble;
3. extraction of the final consensus partition from the accumulated evidence.

In addition, validation of the combined clustering results is a desirable final step.

In the following sections we outline contributions on learning similarities under the EAC framework focusing on new combination methods, constrained clustering, cluster validity, scalability issues, and ap-

plications to electrophysiological temporal data. Finally, we refer the development status of a toolbox for Matlab, built under an object-oriented paradigm, that provides an up-to-date environment for the application of the clustering ensemble approach.

2.1.1 Combining Evidence from Multiple Clusterings

The EAC approach combines evidence, from pairwise associations performed by the clustering committee, based on a voting mechanism that yields, as summarizing entity, a co-association matrix. This constitutes the intermediate step of evidence accumulation. A consensus partition is obtained by applying some clustering strategy over this matrix. Progress undertaken in the combination process explored the dual interpretation of the co-association matrix as expressing similarities and as probabilities.

- Taking the pair-wise similarity, learned with the EAC method, as estimate of the probability of pairs of objects to belong to the same cluster, we proposed a probabilistic formulation for the combination process, leading to a consensus soft partition solution, where each object is probabilistically assigned to a cluster. The method reduces the clustering problem to a polynomial optimization in probability domain, which is attacked by means of the Baum-Eagon inequality. This work presents a principled probabilistic solution for consensus clustering, going one step further by extending the EAC paradigm from hard data partitioning to soft clustering solutions. Details and results can be found in [2].
- Taking co-occurrence information as the starting point, we have proposed a probabilistic generative model for consensus clustering, based on a dyadic aspect model for the evidence accumulation clustering framework [3]. Starting from the observation that co-occurrences are a special type of dyads, we proposed to model co-association using a generative aspect model for dyadic data.

Under the proposed model, the extraction of a consensus clustering corresponds to solving a maximum likelihood estimation problem, which we address using the expectation-maximization algorithm. We referred to the resulting method as probabilistic ensemble clustering algorithm (PEnCA). Moreover, the fact that the problem is placed in a probabilistic framework allows using model selection criteria to automatically choose the number of clusters. The output of the method is a probabilistic assignment of each sample to each cluster. One of the advantages of this framework is the possibility of inclusion of a model selection criterion. We hope to further address this issue in the near future. Ongoing work on different initialization schemes and strategies to escape from local solutions is being carried out.

- Different clustering techniques can be applied to the co-association matrix to obtain the combined data partition, and different clustering strategies may yield too different combination results. In an attempt to reduce the sensitivity of the final partition to this clustering method, and still obtain competitive and consistent results, we have proposed to apply embedding methods over this matrix [4]. We performed a study of several embedding methods over the co-association matrix, interpreting it in two ways: (i) as a feature space and (ii) as a similarity space. In the first case dimensionality reduction is performed of the feature space; in the second case we obtain a new representation constrained to the similarity matrix. When applying several clustering techniques over these new representations, we evaluated the impact of these transformations in terms of performance and coherence of the obtained data partitions. Experimental results, on synthetic and real benchmark datasets, have shown that extracting the relevant features through dimensionality reduction yields more consistent results than applying the clustering algorithms directly to

the co-association matrix.

The work undertaken involved a close collaboration between the IST and the UNIVE partners, which should be further strengthened in the future. In the later work, pair-wise similarities were used; extension of this method to higher order similarities is object of current joint research.

2.1.2 Constrained Clustering

Recent work on clustering has focused on the incorporation of a priori knowledge, mostly in the form of pairwise constraints, aiming to improve clustering quality of individual clustering algorithms, and find appropriate clustering solutions to specific tasks or interests. In the context of SIMBAD, we proposed to extend and integrate the constrained clustering idea into the CE framework. Such integration can be implemented at three main levels, and combinations thereof: (a) on the construction of the CE, by explicitly applying constrained clustering algorithms; (b) during information combination phase, by forcing (hard constraints) or encouraging (soft constraints) pairwise associations; (c) at the step of extraction of the final (combined) data partition. In the work developed so far, we proposed an extension to EAC (termed CEAC), and a novel algorithm (ACCCS) to solve the CE problem using pairwise constraints (must-link and cannot-link). CEAC consists of enforcing the clustering algorithm, which produces the consensus partition from the learned similarity, to support the incorporation of must-link and cannot-link constraints. The ACCCS approach comprises the maximization of both the similarity between CE partitions and a target consensus partition, and constraints satisfaction. Experimental results have shown the proposed constrained clustering combination methods performances are superior to the unconstrained Evidence Accumulation Clustering. Part of this work is reported in [5].

2.1.3 Clustering Validity

Consider the following question:"For a given data set, which clustering solution should be selected". The solution to this problem is based on clustering validation. While there is much work reported in the literature on validating data partitions produced by single clustering algorithms, little has been done in order to validate data partitions produced by clustering combination methods. Most of these works use measures of consistency between consensus solutions and the clustering ensemble, such as the Average Normalized Mutual Information proposed by Strehl and Ghosh.

We first addressed the validation issue at the clustering ensemble level, proposing the Average Cluster Consistency (ACC) index [6]. The main idea consists of measuring how well the clusters in the clustering ensemble fit in the clusters of the consensus partition. The similarity between each partition in the CE and the combined partition is measured based on a weighting of shared samples in matching clusters. The ACC validity index accounts for the average of these similarities over the CE. Details and results on this work can be found in [6].

Further work in this research line have proposed the validation of clustering combination results at three levels:

1. Original data representation - measure the consistency of clustering solutions with the structure of the data, perceived from the original representation (either feature-based or similarity-based);

2. Clustering ensemble level - measure the consistency of consensus partitions with the clustering ensemble;

3. Learned pairwise similarity - measure the coherence between clustering solutions and the co-association matrix induced by the clustering ensemble.

Taking pair-wise similarities as the underlying representation, traditional clustering validity indices (namely the Silhouette, Dunn's and Davies and Bouldin's validity indices) were adapted to validate consensus solutions, when compared to the original data representation, and the learned similarity. These validity indices roughly account for intra-cluster compactness and inter-cluster separation. We then proposed a statistical validity index based on pair-wise similarity. According to the new index, the quality of the consensus partition is measured in terms of the likelihood of the data constrained to this partitioning. Inspired on the Parzen-window density estimation technique with variable size windows, a k-nearest neighbor density estimate from pair-wise similarities was defined. Taking as reference the learned similarity, the proposed validity index corresponds to a measure of goodness of fit of the consensus partition with the clustering ensemble and the pair-wise information extracted from it. When assessed from the original data representation, this validity index measures the goodness of fit of the combined partition with the statistical properties of the data on the baseline representation. A comparative study of the several validation approaches was undertaken on synthetic and real data. Details and results can be found in [7].

2.1.4 Scalability

We have addressed the scalability problem of the evidence accumulation clustering method, intrinsically related to the storage of the co-association matrix. This topic was dealt in collaboration with Prof. Anil K. Jain, from the Michigan State University, USA. The bottleneck of the evidence accumulation paradigm is the quadratic (on the number of samples) space complexity associated with the full representation of the co-association matrix. Taking advantage of the sparseness of this matrix, we adopted a sparse matrix representation, reducing the space complexity of the method. In order to further

reduce the space complexity, we have proposed a clustering ensemble construction rule, following a split and merge strategy, according to which the clustering algorithms are applied with K, the number of clusters, randomly chosen in the interval [Kmin,Kmax]. Criteria for the choice of these extreme values were also proposed and analyzed, showing that both space complexity and quality of combination results dependent on the partitioning granularity, dictated by the value of Kmin. Experimental results confirmed that this strategy leads to linear space complexity of evidence accumulation clustering, enabling the scalability of this framework to large data-sets. We have shown that this significant space complexity improvements do not compromise, and may even lead to increased performance of clustering combination. Details and results can be found in [8].

2.1.5 Learning Similarity on Temporal Data

In the context of SIMBAD, the CE framework was further explored and extended to learn similarity relations of temporal data, We proposed a methodology for the analysis of data characterized by temporal evolution, such as electrophysiological signals. This methodology is based on the clustering ensemble method, and on a genetic algorithm for assessment of the existence of differentiated states in time series [9].

Taking as motivating application the evaluation of changes in ECG morphology in the course of the a stress-inducing computer-based activity, the evidence accumulation clustering method was applied and evaluated using different clustering algorithms for the construction of clustering ensembles as well as various algorithms for final extraction of the (combined) final partition; these various setups were additionally explored in conjunction with feature selection and feature extraction techniques.

The developed work presents several innovative aspects:

- Stress-related ECG morphological changes. In previous work, stress has been found to be associated with heart rate variability. However, morphological changes have not been studied so far. In our work, we addressed this issue, by assessing the temporal evolution of ECG morphology, summarized in a similarity matrix between heart beat waves, indices of the matrix corresponding to increasing time stamps. Our results confirm this morphology change hypothesis, showing clear dissimilarity between ECG patterns at the beginning and at the end of the task; furthermore, by clustering the learned similarity matrix using the CE approach, such a hypothesis is confirmed by revealing distinct clusters.
- Methodology for the analysis of temporal data based on the clustering ensemble approach.
- Genetic algorithm for temporal data denoising. Clustering of stationary temporal data with abrupt changes in the temporal organization model is a relatively simple problem. Given the continuous time evolution of stress levels, clustering algorithms are deemed to fail to detect well separated groups of patterns. Therefore, elimination of samples that correspond to the continuous transition between distinct states (denoted as noisy patterns) is one possible approach to detect if such meaningful distinct clusters are present in the data. The genetic algorithm proposed identifies and eliminates transition time frames based on a cluster separability fitness function.

A detailed description of the previous contributions can be found in [9][10][11].

2.2 High order dissimilarities: Dissimilarity Increments

We have addressed the use of high order dissimilarity models in pattern recognition and data mining problems. In this context we ex-

plored dissimilarities between triplets of nearest neighbors, called *dissimilarity increments* (DIs), previously proposed in [12]. In prior work, based on empirical observation, dissimilarity increments were modeled using an exponential distribution. This parametric model for cluster representation formed the basis for a new cluster isolation criterion, that was further integrated in a hierarchical clustering algorithm, having an intuitive design parameter. During the period covered by the current report, we have made the following progress:

- We have addressed the problem of analyzing clustering solutions based on the formalism of probabilistic attributed graphs, exploring dissimilarity increments. Assuming the previously proposed statistical model for DIs, we presented a graph generative model for the clusters. This formed the basis for the design of a new cluster validity index, that consists of the description length of the data partition, represented by a probabilistic attributed graph inferred from the data, conditioned on the given partition [13]. Decision between clustering solutions based on the new index follows a MDL criterion. We applied the proposed criterion in two distinct scenarios: the selection of the design parameter for the hierarchical clustering algorithm mentioned above, and the choice between combination results in a clustering ensemble approach. Results on several data sets, consisting of both synthetic and real data, revealed a good performance of the index in selecting a partition or design parameter.
- We have theoretically derived a statistical model of dissimilarity increments for Gaussian high-dimensional data (d-DID), and have particularized the model for $d = 2$ (2-DID). We empirically compared these two distributions with a prior model considered in [12] (exponential distribution) using two statistical distance measures: Cramér-von-Mises criterion and Jensen-Shannon divergence. Empirical evidence showed that d-DID and 2-DID are

a better approximation to the empirical distribution than the exponential distribution and that 2-DID is a good approximation to d-DID, while being simpler to compute [14]. A detailed version of the derivation of the dissimilarity increments distribution for gaussian d-dimensional data is presented in appendix A.

- We proposed the use of this distribution in clustering, having designed a novel clustering algorithm [15]: the starting point is a partition given by a Gaussian mixture decomposition and the decision of merging components is based on a likelihood ratio test between the statistical model for the combined components and the statistical model for the separate components. In [14] we proposed and evaluated another merge criterion based on the minimum description length, thus obtaining a parameter-free clustering algorithm for arbitrary shaped data, yielding state-of-the-art results in both synthetic and real-world data sets.
- We have proposed to incorporate this DID in a hierarchical clustering algorithm to decide whether two clusters should be merged or not [16]. The novel hierarchical algorithm is parameter-free and can identify classes as the union of clusters following the dissimilarity increments distribution. Experimental results have shown that the proposed algorithm has excellent performance over well separated clusters, also providing a good hierarchical structure insight into touching clusters.
- We have presented a novel maximum a posteriori (MAP) classifier [17] which uses the dissimilarity increments distribution. This classifier, which we named MAP-DID, can be interpreted as a Gaussian Mixture Model with a "harmonizing" operator which forces a class to have a common increment structure, even though each gaussian component within a class can have distinct means and covariances. We have applied the classifier to the dissimilarity data sets assembled in WP3, both in their original dissim-

ilarity representations, and over the several embeddings therein explored. We have shown that MAP-DID outperforms multiple other classifiers on the various datasets (both synthetic and real) and embedding feature spaces.

Although theoretically derived for Gaussian data, we have shown empirically that application of the DID to arbitrary data sets, without the constraint of gaussianity, leads to good performances, both under the supervised and unsupervised approaches. Ongoing work includes a more general theoretical derivation of the distribution of dissimilarity increments, with no assumption about the generating model, focusing on the distribution for the nearest neighbors. Our future plans involve further exploration of the dissimilarity increments for classification purposes. This work is being conducted in collaboration with TU-Delft.

3 Supervised Learning: Multiple Kernel Learning

Despite all the advances in kernel-based machine learning, obtaining good predictors still requires a large effort in feature/kernel design and tuning (often done via cross-validation). Because discriminative training of structured predictors can be quite slow, especially in large-scale settings, it is appealing to learn the kernel function simultaneously.

In multiple kernel learning (MKL, [18, 19]), the kernel is learned as a linear combination of prespecified base kernels. This framework has been made scalable with the advent of wrapper-based methods, in which a standard learning problem (*e.g.*, an SVM) is repeatedly solved in an inner loop up to a prescribed accuracy [20, 21, 22, 23]. Unfortunately, extending such methods to large-scale (namely, structured prediction) still raises practical hurdles: when the output space is large, so are the kernel matrices, and the number of support vec-

tors; when it is prohibitive to tackle the inner learning problem in its batch form, one often needs to resort to online algorithms [24, 25, 26]; the latter are fast learners but slow optimizers [27], hence using them in the inner loop with early stopping may misguide the overall MKL optimization.

In our work in this context, we have proposed to overcome the above difficulties by proposing a stand-alone online MKL algorithm, which exploits the large-scale tradeoffs directly. The algorithm, which when applied to structured prediction problems is termed SPOM (*Structured Prediction by Online MKL*), iterates between subgradient and proximal steps, and has important advantages:

(i) it is simple, flexible, and compatible with sparse and non-sparse variants of MKL;

(ii) it is adequate for structured prediction;

(iii) it offers regret, convergence, and generalization guarantees. Our approach can be seen as a kernelization of the recent forward-backward splitting scheme FOBOS [28], whose regret bound we improve.

This work, with a special emphasis on its application to structured prediction problems, was reported in detail in an AISTAST'2011 (April, 2011) paper [30] and in a paper presented at the NIPS Workshop in New Directions in Multiple Kernel Learning (December, 2010).

References

[1] Fred, A.L., Jain, A.K.: Combining multiple clusterings using evidence accumulation. IEEE Trans. Pattern Anal. Machine Intell. **27**(6) (2005) 835–850

[2] Bulò, S., Lourenço, A., Fred, A., Pelillo, M.: Pairwise probabilistic clustering using evidence accumulation. In Hancock,

E., Wilson, R., Windeatt, T., Ulusoy, I., Escolano, F., eds.: Structural, Syntactic, and Statistical Pattern Recognition. Volume 6218 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2010) 395–404 10.1007/978-3-642-14980-1_38.

[3] Lourenço, A., Fred, A.L.N., Figueiredo, M.: A generative dyadic aspect model for evidence accumulation clustering. In: SIMBAD workshop. Lecture Notes in Computer Science. Springer (2011) To appear.

[4] Aidos, H., Fred, A.L.N.: A study of embedding methods under the evidence accumulation framework. In: SIMBAD workshop. Lecture Notes in Computer Science. Springer (2011) To appear.

[5] Duarte, J.M.M., Fred, A.L.N., Duarte, J.F.: Combining data clusterings with instance level constraints. In Fred, A., ed.: Intl. Workshop on Pattern Recognition in Information Systems, Milan, Itally, INSTICC Press (2009) 49–60

[6] Duarte, F.J., Duarte, J.M.M., Fred, A.L.N., Rodrigues, M.F.: Average cluster consistency for cluster ensemble selection. In Fred, A., Dietz, J.L.G., Liu, K., Filipe, J., eds.: Knowledge Discovery, Knowlege Engineering and Knowledge Management. Volume 128 of Communications in Computer and Information Science. Springer (2011) 133–148 First International Joint Conference, IC3K 2009, Funchal, Madeira, Portugal, October 6-8, 2009, Revised Selected Papers.

[7] Duarte, J., Fred, A., Lourenço, A., Duarte, F.: On consensus clustering validation. In Hancock, E., Wilson, R., Windeatt, T., Ulusoy, I., Escolano, F., eds.: Structural, Syntactic, and Statistical Pattern Recognition. Volume 6218 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2010) 385–394 10.1007/978-3-642-14980-1_37.

[8] Lourenço, A., Fred, A.L., Jain, A.K.: On the scalability of evidence accumulation clustering. In: 20th International Conference on Pattern Recognition (ICPR 2010), Istanbul Turkey, IEEE Computer Society (August 23-26 2010)

[9] Medina, L.A.S., Fred, A.L.N.: Genetic algorithm for clustering temporal data - application to the detection of stress from ECG signals. In: Proc. Intl. Conference on Agents and Artificial Intelligence. (2010) 135–142

[10] Medina, L.A.: Identification of stress states from ECG signals using unsupervised learning methods. Msc. dissertation on electrical engineering and computers, Instituto Superior Tcnico, Universidade Tcnica de Lisboa (April 2009) Supervisor: Prof. Ana Fred. (in Portuguese).

[11] Medina, L.A., Fred, A.L.N.: Clustering data with temporal evolution: Application to electrophysiological signals. In Filipe, J., Fred, A., Sharp, B., eds.: Agents and Artificial Intelligence. Volume 129 of Communications in Computer and Information Science. Springer Berlin Heidelberg (2011) 101–115 10.1007/978-3-642-19890-8_8.

[12] Fred, A., Leitão, J.: A new cluster isolation criterion based on dissimilarity increments. IEEE Trans. on Pattern Analysis and Machine Intelligence **25**(8) (2003) 944–958

[13] Fred, A.L.N., Jain, A.K.: Cluster validation using a probabilistic attributed graph. In: 19th International Conference on Pattern Recognition (ICPR 2008). Volume 1-6., Tampa, Florida, USA, IEEE (December 2008) 2360–2363

[14] Aidos, H., Fred, A.: Statistical modeling of dissimilarity increments for d-dimensional data: Application in partitional clustering. Submitted to Journal Pattern Recognition (2011)

[15] Aidos, H., Fred, A.L.N.: On the distribution of dissimilarity increments. In Vitrià, J., Sanches, J.M., Hernández, M., eds.: Pattern Recognition and Image Analysis. Volume 6669 of Lecture Notes in Computer Science. Springer (2011) 192–199 Iberian Conference on Pattern Recognition and Image Analysis - IbPRIA 2011, Las Palmas de Gran Canaria, Spain.

[16] Aidos, H., Fred, A.L.N.: Hierarchical clustering with high order dissimilarities. In Perner, P., ed.: Machine Learning and Data Mining in Pattern Recognition. Volume 6871 of Lecture Notes in Computer Science. Springer (2011) 280–293 International Conference on Machine Learning and Data Mining - MLDM 2011, New York, NY, USA.

[17] Aidos, H., Fred, A.L.N., Duin, B.: Classification using high order dissimilarities in non-euclidean spaces. Submitted to ICPRAM 2012 (2011)

[18] Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E., and Jordan, M. I. Learning the kernel matrix with semidefinite programming. *JMLR* (2004), 5:27–72.

[19] Bach, F., Lanckriet, G., and Jordan, M. Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML* (2004).

[20] Sonnenburg, S., Rätsch, G., Schäfer, C., and Schölkopf, B. (2006). Large scale multiple kernel learning. *JMLR* (2006), 7:1531–1565.

[21] Zien, A. and Ong, C. Multiclass multiple kernel learning. In *ICML* (2007).

[22] Rakotomamonjy, A., Bach, F., Canu, S., and Grandvalet, Y. SimpleMKL. *JMLR* (2008), 9:2491–2521.

[23] Kloft, M., Brefeld, U., Sonnenburg, S., and Zien, A. Non-Sparse Regularization and Efficient Training with Multiple Kernels (2010). *Arxiv preprint arXiv:1003.0079*

[24] Ratliff, N., Bagnell, J., and Zinkevich, M. Subgradient methods for maximum margin structured learning. In *ICML Workshop on Learning in Structured Outputs Spaces* (2006).

[25] Collins, M., Globerson, A., Koo, T., Carreras, X., and Bartlett, P. Exponentiated gradient algorithms for conditional random fields and max-margin Markov networks. *JMLR* (2008).

[26] Shalev-Shwartz, S., Singer, Y., and Srebro, N. Pegasos: Primal estimated sub-gradient solver for svm. In *ICML* (2007).

[27] Bottou, L. and Bousquet, O. The tradeoffs of large scale learning. *NIPS* (2007).

[28] Duchi, J. and Singer, Y. Efficient online and batch learning using forward backward splitting. *JMLR* (2009), 10:2873–2908.

[29] Martins, A., Smith, N., Xing, E., Aguiar, P., Figueiredo, M. Online MKL for Structured Prediction, NIPS Workshop in New Directions in Multiple Kernel Learning (2010).

[30] Martins, A., Smith, N., Xing, E., Aguiar, P., Figueiredo, M. Online Learning of Structured Predictors with Multiple Kernels, AISTATS (2011).

[31] Johnson, N.L., Kotz, S., Balakrishnan, N.: Continuous Univariate Distributions. 2 edn. Volume 1 of Applied Probability and Statistics. John Wiley & Sons Ltd. (1994)

[32] Olver, F.W.J., Lozier, D.W., Boisvert, R.F., Clark, C.W., eds.: NIST Handbook of Mathematical Functions. Cambridge University Press (2010)

A Dissimilarity Increments Distribution for Gaussian Data

B d-dimensional Gaussian distribution

Assume X is a d-dimensional set of patterns and an element of X is drawn from a normal distribution, $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, and the Euclidean distance is used as the dissimilarity measure. Without loss of generality, let's assume $\boldsymbol{\mu} = 0$ and Σ is a diagonal covariance matrix.

We want to find the distribution of the Euclidean distance between patterns, $D = d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$.

B.1 Distribution of the Euclidean Distance

Let's denote the new data pattern by $\mathbf{x}^*$, so $x_i^* = x_i/\sqrt{\Sigma_{ii}}$ follows the standard normal distribution, $\mathcal{N}(0,1)$. Now, $x_i^* - y_i^* \sim \mathcal{N}(0,2)$. Consider $z_i^* = (x_i^* - y_i^*)/\sqrt{2}$ which will follow the standard normal distribution. Then

$$(D^*)^2 = \|\mathbf{z}^*\|^2 = \sum_{i=1}^{d}(z_i^*)^2 = \sum_{i=1}^{d}\frac{(x_i^* - y_i^*)^2}{2}$$

$(D^*)^2$ follows a chi-square distribution with d degrees of freedom [31]. The probability density function (pdf) is given by:

$$p_{(D^*)^2}(x) = \frac{2^{-d/2}}{\Gamma(d/2)}\, x^{d/2-1} \exp\left(-\frac{x}{2}\right),\ x \in [0, \infty) \tag{1}$$

Define $\mathbf{D}^* = \mathbf{z}^* = \frac{\mathbf{x}^* - \mathbf{y}^*}{2}$. We have a $(d-1)$-sphere, then $\theta_i \sim Unif([0, \pi[), i = 1, \ldots, d-2$ and $\theta_{d-1} \sim Unif([0, 2\pi[)$,

$$\begin{aligned}\mathbf{D}^* =& D^* \cos\theta_1\, \mathbf{e}_1 + D^* \sin\theta_1 \cos\theta_2\, \mathbf{e}_2 + \ldots \\ &+ D^* \sin\theta_1 \sin\theta_2 \ldots \sin\theta_{d-2} \cos\theta_{d-1}\, \mathbf{e}_{d-1} \\ &+ D^* \sin\theta_1 \sin\theta_2 \ldots \sin\theta_{d-2} \sin\theta_{d-1}\, \mathbf{e}_d.\end{aligned}$$

Furthermore, $\mathbf{D} = \mathbf{x} - \mathbf{y} = (b_1, b_2, \ldots, b_d)$, where

$$\begin{aligned}
b_1 =& \sqrt{2\Sigma_{11}}\, D^* \cos\theta_1 \\
b_2 =& \sqrt{2\Sigma_{22}}\, D^* \sin\theta_1 \cos\theta_2 \\
b_3 =& \sqrt{2\Sigma_{33}}\, D^* \sin\theta_1 \sin\theta_2 \cos\theta_3 = \sqrt{2\Sigma_{33}}\, D^* \left[\prod_{k=1}^{2} \sin\theta_k\right] \cos\theta_3 \\
&\vdots \\
b_{d-1} =& \sqrt{2\Sigma_{d-1,d-1}}\, D^* \sin\theta_1 \sin\theta_2 \ldots \sin\theta_{d-2} \cos\theta_{d-1} = \sqrt{2\Sigma_{d-1,d-1}}\, D^* \left[\prod_{k=1}^{d-2} \sin\theta_k\right] \cos\theta_{d-1} \\
b_d =& \sqrt{2\Sigma_{dd}}\, D^* \sin\theta_1 \sin\theta_2 \ldots \sin\theta_{d-2} \sin\theta_{d-1} = \sqrt{2\Sigma_{dd}}\, D^* \left[\prod_{k=1}^{d-1} \sin\theta_k\right].
\end{aligned}$$

We know that $\|ax\| = |a|\,\|x\|$, so

$$\begin{aligned}
D^2 = \|\mathbf{D}\|^2 &= (\sqrt{2\Sigma_{11}}\, D^* \cos\theta_1)^2 + (\sqrt{2\Sigma_{22}}\, D^* \sin\theta_1 \cos\theta_2)^2 + \dots \\
&+ \left(\sqrt{2\Sigma_{d-1,d-1}}\, D^* \left[\prod_{k=1}^{d-2} \sin\theta_k\right] \cos\theta_{d-1}\right)^2 + \left(\sqrt{2\Sigma_{dd}}\, D^* \left[\prod_{k=1}^{d-1} \sin\theta_k\right]\right)^2 \\
&= 2\left(\Sigma_{11}\cos^2\theta_1 + \Sigma_{22}\sin^2\theta_1\cos^2\theta_2 + \dots + \Sigma_{d-1,d-1}\left[\prod_{k=1}^{d-2}\sin^2\theta_k\right]\cos^2\theta_{d-1} + \Sigma_{dd}\left[\prod_{k=1}^{d-1}\sin^2\theta_k\right]\right)(D^*)^2,
\end{aligned} \tag{2}$$

with $A(\Theta)^2$ (with $\Theta = (\theta_1, \theta_2, \dots, \theta_{d-1})$) the expansion factor. Naturally this expansion factor will depend on the angles Θ. In practice it is hard to properly deal with this dependence. Therefore we will use the approximation that the expansion factor is constant and equal to the average value of the true expansion factor.

We must therefore find $\mathbb{E}[A(\Theta)^2]$. Θ is the angular coordinate of a point in the $(d-1)$-sphere. Therefore the volume element of the integral will be

$$\begin{aligned}
\mathrm{d}_{S^{d-1}}V &= \sin^{d-2}\theta_1 \sin^{d-3}\theta_2 \dots \sin^2\theta_{d-3} \sin\theta_{d-2}\, \mathrm{d}\theta_1\, \mathrm{d}\theta_2\, \dots \mathrm{d}\theta_{d-3}\, \mathrm{d}\theta_{d-2}\, \mathrm{d}\theta_{d-1} \\
&= \left(\prod_{i=1}^{d-2} \sin^{d-(i+1)}\theta_i\right)\, \mathrm{d}\theta_1\, \mathrm{d}\theta_2\, \dots \mathrm{d}\theta_{d-3}\, \mathrm{d}\theta_{d-2}\, \mathrm{d}\theta_{d-1},
\end{aligned}$$

and the expected value is given by

$$\mathbb{E}[A(\Theta)^2] = \int_{S^{d-1}} \left[\prod_{i=1}^{d-2} p_{\theta_i}(\theta_i)\right] p_{\theta_{d-1}}(\theta_{d-1}) A(\Theta)^2\, \mathrm{d}_{S^{d-1}}V$$

Since $\theta_i \sim Unif([0,\pi[), \forall i = 1, \dots, d-2$ and $\theta_{d-1} \sim Unif([0, 2\pi[)$, then $p_{\theta_i}(\theta_i) = \frac{1}{\pi}$ and $p_{\theta_{d-1}}(\theta_{d-1}) = \frac{1}{2\pi}$. So,

$$\begin{aligned}
\mathbb{E}[A(\Theta)^2] = \frac{1}{2\pi^{d-1}} \int_0^{2\pi}\int_0^{\pi}\cdots\int_0^{\pi} &\left(2\Sigma_{11}\cos^2\theta_1 + 2\Sigma_{22}\sin^2\theta_1\cos^2\theta_2 + \dots\right. \\
&\left. + 2\Sigma_{d-1,d-1}\left[\prod_{k=1}^{d-2}\sin^2\theta_k\right]\cos^2\theta_{d-1} + 2\Sigma_{dd}\left[\prod_{k=1}^{d-1}\sin^2\theta_k\right]\right)\cdot \\
&\cdot \left(\prod_{i=1}^{d-2}\sin^{d-(i+1)}\theta_i\right)\, \mathrm{d}\theta_1\, \mathrm{d}\theta_2\, \dots \mathrm{d}\theta_{d-3}\, \mathrm{d}\theta_{d-2}\, \mathrm{d}\theta_{d-1}
\end{aligned}$$

We have d portions of a sum, so we need to solve the $d-1$ integrals for each portion. We will use the fact that

$$\int_0^{\pi}\cos^2(x)\sin^k(x)\,\mathrm{d}x = \frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{2\,\Gamma(2+\frac{k}{2})} \quad \text{and} \quad \int_0^{\pi}\sin^k(x)\,\mathrm{d}x = \frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{\Gamma(1+\frac{k}{2})}, \text{ with } k > 0 \text{ integer.}$$

Also,

$$\prod_{k=1}^{M}\frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{\Gamma(1+\frac{k}{2})} = \frac{\pi^{M/2}}{\Gamma(1+\frac{M}{2})} \quad \text{and} \quad \prod_{k=M}^{d}\frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{\Gamma(1+\frac{k}{2})} = \frac{\pi^{\frac{d-M+1}{2}}\,\Gamma(\frac{M+1}{2})}{\Gamma(1+\frac{d}{2})}$$

We will solve the integrals for each portion of the sum separately, and then combine all.

(1)

$$\int_0^{2\pi}\int_0^{\pi}\cdots\int_0^{\pi} 2\Sigma_{11}\cos^2\theta_1\left(\prod_{i=1}^{d-2}\sin^{d-(i+1)}\theta_i\right)\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2\,\ldots\mathrm{d}\theta_{d-3}\,\mathrm{d}\theta_{d-2}\,\mathrm{d}\theta_{d-1}$$

$$= 2\Sigma_{11}\underbrace{\int_0^{2\pi}\mathrm{d}\theta_{d-1}}_{2\pi}\underbrace{\int_0^{\pi}\sin\theta_{d-2}\,\mathrm{d}\theta_{d-2}}_{2}\cdots\underbrace{\int_0^{\pi}\sin^{d-3}\theta_2\,\mathrm{d}\theta_2}_{\frac{\sqrt{\pi}\,\Gamma(\frac{d-2}{2})}{\Gamma(1+\frac{d-3}{2})}}\underbrace{\int_0^{\pi}\cos^2\theta_1\sin^{d-2}\theta_1\,\mathrm{d}\theta_1}_{\frac{\sqrt{\pi}\,\Gamma(\frac{d-1}{2})}{2\,\Gamma(2+\frac{d-2}{2})}}$$

$$= 2\Sigma_{11}2\pi\left(\prod_{k=1}^{d-3}\frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{\Gamma(1+\frac{k}{2})}\right)\frac{\sqrt{\pi}\,\Gamma(\frac{d-1}{2})}{2\,\Gamma(2+\frac{d-2}{2})}$$

$$= 2\Sigma_{11}\pi\frac{\pi^{d-3/2}}{\Gamma(\frac{d-1}{2})}\,\frac{\sqrt{\pi}\,\Gamma(\frac{d-1}{2})}{\Gamma(1+\frac{d}{2})} = 2\Sigma_{11}\frac{\pi^{d/2}}{\Gamma(1+\frac{d}{2})}$$

(2)

$$\int_0^{2\pi}\int_0^{\pi}\cdots\int_0^{\pi} 2\Sigma_{22}\sin^2\theta_1\cos^2\theta_2\left(\prod_{i=1}^{d-2}\sin^{d-(i+1)}\theta_i\right)\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2\,\ldots\mathrm{d}\theta_{d-3}\,\mathrm{d}\theta_{d-2}\,\mathrm{d}\theta_{d-1}$$

$$= 2\Sigma_{22}\underbrace{\int_0^{2\pi}\mathrm{d}\theta_{d-1}}_{2\pi}\underbrace{\int_0^{\pi}\sin\theta_{d-2}\,\mathrm{d}\theta_{d-2}}_{2}\cdots\underbrace{\int_0^{\pi}\cos^2\theta_2\sin^{d-3}\theta_2\,\mathrm{d}\theta_2}_{\frac{\sqrt{\pi}\,\Gamma(\frac{d-2}{2})}{2\,\Gamma(2+\frac{d-3}{2})}}\underbrace{\int_0^{\pi}\sin^{d}\theta_1\,\mathrm{d}\theta_1}_{\frac{\sqrt{\pi}\,\Gamma(\frac{d+1}{2})}{\Gamma(1+\frac{d}{2})}}$$

$$= 2\Sigma_{22}2\pi\left(\prod_{k=1}^{d-4}\frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{\Gamma(1+\frac{k}{2})}\right)\frac{\sqrt{\pi}\,\Gamma(\frac{d-2}{2})}{2\,\Gamma(2+\frac{d-3}{2})}\,\frac{\sqrt{\pi}\,\Gamma(\frac{1+d}{2})}{\Gamma(1+\frac{d}{2})}$$

$$= 2\Sigma_{22}2\pi\left(\prod_{k=1}^{d-4}\frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{\Gamma(1+\frac{k}{2})}\right)\frac{\sqrt{\pi}\,\Gamma(\frac{d-2}{2})}{2\,\Gamma(2+\frac{d-3}{2})}\,\left(\prod_{k=d}^{d}\frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{\Gamma(1+\frac{k}{2})}\right)$$

$$= 2\Sigma_{22}\pi\frac{\pi^{\frac{d-4}{2}}}{\Gamma(\frac{d-2}{2})}\,\frac{\sqrt{\pi}\,\Gamma(\frac{d-2}{2})}{\Gamma(\frac{d+1}{2})}\,\frac{\sqrt{\pi}\,\Gamma(\frac{d+1}{2})}{\Gamma(1+\frac{d}{2})} = 2\Sigma_{22}\frac{\pi^{d/2}}{\Gamma(1+\frac{d}{2})}$$

$\vdots$

$(d-3)$

$$\int_0^{2\pi}\int_0^{\pi}\cdots\int_0^{\pi} 2\Sigma_{d-3,d-3}\left[\prod_{k=1}^{d-4}\sin^2\theta_k\right]\cos^2\theta_{d-3}\left(\prod_{i=1}^{d-2}\sin^{d-(i+1)}\theta_i\right)\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2\,\ldots\mathrm{d}\theta_{d-3}\,\mathrm{d}\theta_{d-2}\,\mathrm{d}\theta_{d-1}$$

$$= 2\Sigma_{d-3,d-3}2\pi\left(\prod_{k=1}^{1}\frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{\Gamma(1+\frac{k}{2})}\right)\frac{\sqrt{\pi}\,\Gamma(\frac{3}{2})}{2\,\Gamma(3)}\,\left(\prod_{k=5}^{d}\frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{\Gamma(1+\frac{k}{2})}\right)$$

$$= 2\Sigma_{d-3,d-3}\pi\frac{\pi^{1/2}}{\Gamma(\frac{3}{2})}\,\frac{\sqrt{\pi}\,\Gamma(\frac{3}{2})}{\Gamma(3)}\,\pi^{\frac{d-4}{2}}\,\frac{\Gamma(3)}{\Gamma(1+\frac{d}{2})} = 2\Sigma_{d-3,d-3}\frac{\pi^{d/2}}{\Gamma(1+\frac{d}{2})}$$

$(d-2)$

$$\int_0^{2\pi}\int_0^{\pi}\cdots\int_0^{\pi} 2\Sigma_{d-2,d-2}\left[\prod_{k=1}^{d-3}\sin^2\theta_k\right]\cos^2\theta_{d-2}\left(\prod_{i=1}^{d-2}\sin^{d-(i+1)}\theta_i\right)\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2\,\ldots\mathrm{d}\theta_{d-3}\,\mathrm{d}\theta_{d-2}\,\mathrm{d}\theta_{d-1}$$
$$= 2\Sigma_{d-2,d-2}2\pi\frac{\sqrt{\pi}\,\Gamma(1)}{2\,\Gamma(2+\frac{1}{2})}\,\left(\prod_{k=4}^{d}\frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{\Gamma(1+\frac{k}{2})}\right)$$
$$= 2\Sigma_{d-2,d-2}\pi\frac{\pi^{1/2}\,\Gamma(1)}{\Gamma(\frac{5}{2})}\,\pi^{\frac{d-3}{2}}\frac{\Gamma(\frac{5}{2})}{\Gamma(1+\frac{d}{2})} = 2\Sigma_{d-2,d-2}\frac{\pi^{d/2}}{\Gamma(1+\frac{d}{2})}$$

$(d-1)$

$$\int_0^{2\pi}\int_0^{\pi}\cdots\int_0^{\pi} 2\Sigma_{d-1,d-1}\left[\prod_{k=1}^{d-2}\sin^2\theta_k\right]\cos^2\theta_{d-1}\left(\prod_{i=1}^{d-2}\sin^{d-(i+1)}\theta_i\right)\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2\,\ldots\mathrm{d}\theta_{d-3}\,\mathrm{d}\theta_{d-2}\,\mathrm{d}\theta_{d-1}$$
$$= 2\Sigma_{d-1,d-1}\pi\,\left(\prod_{k=3}^{d}\frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{\Gamma(1+\frac{k}{2})}\right)$$
$$= 2\Sigma_{d-1,d-1}\pi\,\pi^{\frac{d-2}{2}}\frac{\Gamma(2)}{\Gamma(1+\frac{d}{2})} = 2\Sigma_{d-1,d-1}\frac{\pi^{d/2}}{\Gamma(1+\frac{d}{2})}$$

(d)

$$\int_0^{2\pi}\int_0^{\pi}\cdots\int_0^{\pi} 2\Sigma_{dd}\left[\prod_{k=1}^{d-1}\sin^2\theta_k\right]\left(\prod_{i=1}^{d-2}\sin^{d-(i+1)}\theta_i\right)\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2\,\ldots\mathrm{d}\theta_{d-3}\,\mathrm{d}\theta_{d-2}\,\mathrm{d}\theta_{d-1}$$
$$= 2\Sigma_{dd}\pi\,\left(\prod_{k=3}^{d}\frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{\Gamma(1+\frac{k}{2})}\right)$$
$$= 2\Sigma_{dd}\pi\,\pi^{\frac{d-2}{2}}\frac{\Gamma(2)}{\Gamma(1+\frac{d}{2})} = 2\Sigma_{dd}\frac{\pi^{d/2}}{\Gamma(1+\frac{d}{2})}$$

Which means, if we sum (1) to (d), we get

$$\mathbb{E}[A(\Theta)^2] = \frac{1}{2\pi^{d-1}}\,\frac{\pi^{d/2}}{\Gamma(1+\frac{d}{2})}\,2\,\mathrm{tr}(\Sigma) = \frac{\pi^{-d/2+1}}{\Gamma(1+\frac{d}{2})}\,\mathrm{tr}(\Sigma)$$

Therefore, the transformation equation (2) from the normalized space to the original space is

$$\boxed{D^2 = \frac{\pi^{-d/2+1}}{\Gamma(1+\frac{d}{2})}\,\mathrm{tr}(\Sigma)(D^*)^2}$$

Now, we know that $(D^*)^2 \sim \chi^2(d)$ and if $Y = aX$, a constant, with $p_X(x)$ the pdf of X, we get $p_Y(y) = p_X(y/a)\frac{dx}{dy} = p_X(y/a)\cdot 1/a$. So, from equation (1), we have

$$p_{D^2}(w) = \frac{d^{d/2}\,\left(\Gamma(d/2)\right)^{d/2-1}}{\left(4\,\mathrm{tr}(\Sigma)\right)^{d/2}\,\pi^{-d/2(d/2-1)}}\,w^{d/2-1}\exp\left(-\frac{d\,\Gamma(d/2)\,w}{4\,\mathrm{tr}(\Sigma)\,\pi^{-d/2+1}}\right),\ w\in[0,\infty).$$

Again, we can assume that $Y = \sqrt{X} \Leftrightarrow Y^2 = X$, with $p_X(x)$ the pdf of X, we get $p_Y(y) = p_X(y^2)\frac{dx}{dy} = p_X(y^2)\cdot 2y$. Therefore,

$$\boxed{p_D(z) = \frac{2^{1-d}\,d^{d/2}\,\left(\Gamma(d/2)\right)^{d/2-1}}{\left(\mathrm{tr}(\Sigma)\right)^{d/2}\,\pi^{-d/2(d/2-1)}}\,z^{d-1}\exp\left(-\frac{d\,\Gamma(d/2)\,z^2}{4\,\mathrm{tr}(\Sigma)\,\pi^{-d/2+1}}\right),\ z\in[0,\infty).} \quad (3)$$

B.2 Probability Density Function for Increments

We already know that $D_1 = d(\mathbf{x}, \mathbf{y})$ and $D_2 = d(\mathbf{y}, \mathbf{z})$ follow the distribution with pdf in equation (3). Consider $D_1 - D_2 = W$ and $D_2 = T$, with $D_1 = T + W$, so the pdf for $D_1 - D_2$ is given by the convolution

$$p_{_W}(w) = \int_{-\infty}^{\infty} \frac{2^{2-2d}\, d^d\, \Gamma(d/2)^{d-2}}{(\mathrm{tr}(\Sigma))^d\, \pi^{-d(d/2-1)}}\, (t(t+w))^{d-1} \exp\left(-\frac{d\,\Gamma(d/2)}{4\,\mathrm{tr}(\Sigma)\,\pi^{-d/2+1}}\left(t^2 + (t+w)^2\right)\right) U(t+w)U(t)\,\mathrm{d}t \tag{4}$$

with $U(\cdot)$ the unit step function, $U(t) = 1$ if $t \geq 0$ and $U(t) = 0$ if $t < 0$.

Assume $C(d) = 2^{2-2d}(d/\,\mathrm{tr}(\Sigma))^d\, \Gamma(d/2)^{d-2}\, \pi^{d(d/2-1)}$ and $A(d) = d\,\Gamma(d/2)\, 2^{-2}\,\mathrm{tr}(\Sigma)^{-1}\, \pi^{d/2-1}$.

Case 1: $w \geq 0$

In this case the integral we need to solve will be

$$p_{_W}(w) = \int_0^{\infty} C(d)\, (t(t+w))^{d-1} \exp\left(-A(d)\left(t^2 + (t+w)^2\right)\right)\,\mathrm{d}t. \tag{5}$$

By the binomial formula we have

$$(t(t+w))^{d-1} = t^{d-1}\left[\sum_{k=0}^{d-1}\binom{d-1}{k} t^{d-1-k} w^k\right] = \sum_{k=0}^{d-1}\binom{d-1}{k} t^{2d-2-k} w^k$$

and replacing in (5), we get

$$\begin{aligned}
p_{_W}(w) &= C(d)\int_0^{\infty}\sum_{k=0}^{d-1}\binom{d-1}{k} t^{2d-2-k} w^k \exp\left(-A(d)\left(t^2 + (t+w)^2\right)\right)\,\mathrm{d}t \\
&= C(d)\left[\sum_{k=0}^{d-1}\binom{d-1}{k} w^k \int_0^{\infty} t^{2d-2-k} \exp\left(-A(d)\left(t^2 + (t+w)^2\right)\right)\,\mathrm{d}t\right] \\
&= C(d)\left[\sum_{k=0}^{d-1}\binom{d-1}{k} w^k \int_0^{\infty} t^{2d-2-k} \exp\left(-A(d)\left(\sqrt{2}t + w/\sqrt{2}\right)^2 + w^2/2\right)\,\mathrm{d}t\right] \\
&= C(d)\left[\sum_{k=0}^{d-1}\binom{d-1}{k} w^k \int_0^{\infty} t^{2d-2-k} \exp\left(-A(d)w^2/2\right)\exp\left(-A(d)\left(\sqrt{2}t + w/\sqrt{2}\right)^2\right)\,\mathrm{d}t\right]
\end{aligned}$$

Let's make a change of variables, assume $u = \sqrt{2}t + w/\sqrt{2}$. The interval of integration with the new variable is $\left[w/\sqrt{2}, \infty\right)$ and $\frac{dt}{du} = 1/\sqrt{2}$. Now the integral can be written as

$$p_{_W}(w) = C(d)\left[\sum_{k=0}^{d-1}\binom{d-1}{k}\frac{w^k}{\sqrt{2}}\,\exp\left(-A(d)w^2/2\right)\int_{w/\sqrt{2}}^{\infty}\left(\frac{u}{\sqrt{2}} - \frac{w}{2}\right)^{2d-2-k}\exp\left(-A(d)u^2\right)\,\mathrm{d}u\right].$$

Again, by the binomial formula,

$$\left(\frac{u}{\sqrt{2}} - \frac{w}{2}\right)^{2d-2-k} = \sum_{i=0}^{2d-2-k}(-1)^i\binom{2d-2-k}{i}\left(\frac{u}{\sqrt{2}}\right)^{2d-2-k-i}\left(\frac{w}{2}\right)^i,$$

substituting in the previous integral, we get

$$\begin{aligned}
p_{_W}(w) = C(d)\Bigg[\sum_{k=0}^{d-1}\binom{d-1}{k}\frac{w^k}{\sqrt{2}}\,\exp\left(-A(d)w^2/2\right)\times \\
\left[\sum_{i=0}^{2d-2-k}(-1)^i\binom{2d-2-k}{i} w^i\, 2^{-d+k/2-i/2+1}\int_{w/\sqrt{2}}^{\infty} u^{2d-2-k-i}\exp\left(-A(d)u^2\right)\,\mathrm{d}u\right]\Bigg]
\end{aligned}$$

Let's make a new change of variables, assume $x = u^2$. The interval of integration with the new variable is $\left[w^2/2, \infty\right)$ and $\frac{du}{dx} = x^{-1/2}/2$. Now the integral can be written as

$$p_W(w) = C(d)\left[\sum_{k=0}^{d-1}\binom{d-1}{k}\frac{w^k}{\sqrt{2}}\,\exp\left(-A(d)w^2/2\right)\times \left[\sum_{i=0}^{2d-2-k}(-1)^i\binom{2d-2-k}{i}w^i\,2^{-d+k/2-i/2+1}\int_{w^2/2}^{\infty}2^{-1}x^{-1/2}x^{d-1-k/2-i/2}\exp\left(-A(d)x\right)\,\mathrm{d}x\right]\right]$$

$$= C(d)\left[\sum_{k=0}^{d-1}\binom{d-1}{k}\frac{w^k}{\sqrt{2}}\,\exp\left(-A(d)w^2/2\right)\times \left[\sum_{i=0}^{2d-2-k}(-1)^i\binom{2d-2-k}{i}w^i\,2^{-d+k/2-i/2}\int_{w^2/2}^{\infty}x^{d-3/2-k/2-i/2}\exp\left(-A(d)x\right)\,\mathrm{d}x\right]\right]$$

The upper incomplete gamma function is defined as

$$\Gamma(a,x) = \int_x^{\infty} t^{a-1}e^{-t}\,\mathrm{d}t$$

and it is easy to prove that

$$\frac{\Gamma(a,bx)}{b^a} = \int_x^{\infty} t^{a-1}e^{-bt}\,\mathrm{d}t,$$

Proof: We need to change variable, $t = bu$, the new interval of integration is $[x,\infty)$ and $\frac{dt}{du} = b$.

$$\Gamma(a,bx) = \int_{bx}^{\infty} t^{a-1}e^{-t}\,\mathrm{d}t = \int_x^{\infty}(bu)^{a-1}e^{-bu}b\,\mathrm{d}u = b^a\int_x^{\infty}u^{a-1}e^{-bu}\,\mathrm{d}u \qquad \square$$

Now,

$$p_W(w) = C(d)\left[\sum_{k=0}^{d-1}\binom{d-1}{k}\frac{w^k}{\sqrt{2}}\,\exp\left(-A(d)w^2/2\right)\times \left[\sum_{i=0}^{2d-2-k}(-1)^i\binom{2d-2-k}{i}w^i\,2^{-d+k/2-i/2}\frac{\Gamma\left(d-1/2-k/2-i/2, A(d)w^2/2\right)}{A(d)^{d-1/2-k/2-i/2}}\right]\right]$$

Recall that $C(d) = 2^{2-2d}(d/\operatorname{tr}(\Sigma))^d\,\Gamma(d/2)^{d-2}\,\pi^{d(d/2-1)}$ and $A(d) = d\,\Gamma(d/2)\,2^{-2}\operatorname{tr}(\Sigma)^{-1}\,\pi^{d/2-1}$,

$$p_W(w) = 2^{1/2-d}\left(\frac{d\,\pi^{d/2-1}}{\operatorname{tr}(\Sigma)\,\Gamma(d/2)^3}\right)^{1/2}\exp\left(-\frac{d\,\Gamma(d/2)\,\pi^{d/2-1}}{8\operatorname{tr}(\Sigma)}\,w^2\right)\left[\sum_{k=0}^{d-1}\sum_{i=0}^{2d-2-k}(-1)^i\binom{d-1}{k}\binom{2d-2-k}{i}\times w^{k+i}\,2^{-k/2-3i/2}\left(\frac{d\,\Gamma(d/2)\,\pi^{d/2-1}}{\operatorname{tr}(\Sigma)}\right)^{k/2+i/2}\Gamma\left(\frac{2d-1-k-i}{2}, \frac{d\,\Gamma(d/2)\,\pi^{d/2-1}}{8\operatorname{tr}(\Sigma)}\,w^2\right)\right] \tag{6}$$

We need to use some properties of the gamma function. Namely, $\Gamma(z) = \Gamma(1+z) = z\Gamma(z)$, $\Gamma(1/2) = \sqrt{\pi}$ and $\Gamma(1/2,z) = \sqrt{\pi}\operatorname{erfc}(\sqrt{z})$. Assume a is a rational positive number that can be written $a = n + 1/2$, so

$$\Gamma(a,z) = e^{-z}\sum_{k=0}^{n-1}\frac{\Gamma(a)}{\Gamma(a-k)}\,z^{a-1-k} + \frac{\Gamma(a)}{\Gamma(a-n)}\,\Gamma(1/2,z)$$

Therefore,

$$\Gamma(a,z) = \begin{cases} e^{-z} \displaystyle\sum_{k=0}^{a-1/2-1} \frac{\Gamma(a)}{\Gamma(a-k)} z^{a-1-k} + \Gamma(a)\,\mathrm{erfc}(\sqrt{z}) & \text{if } a-1/2 \in \mathbb{Z}^+ \\ (a-1)!\; e^{-z} \displaystyle\sum_{k=0}^{a-1} \frac{z^k}{k!} & \text{if } a \in \mathbb{Z}^+ \end{cases}$$

Case 2: $w < 0$

In this case the equation (4) is given by

$$p_{_W}(w) = \int_{-w}^{\infty} C(d)\; (t(t+w))^{d-1} \exp\left(-A(d)\left(t^2 + (t+w)^2\right)\right)\; \mathrm{d}t.$$

To solve the previous integral we use the analogous ideas to the case $w \geq 0$. Therefore, for $w < 0$,

$$p_{_W}(w) = 2^{1/2-d} \left(\frac{d\;\pi^{d/2-1}}{\mathrm{tr}(\Sigma)\;\Gamma(d/2)^3}\right)^{1/2} \exp\left(-\frac{d\;\Gamma(d/2)\;\pi^{d/2-1}}{8\,\mathrm{tr}(\Sigma)}\; w^2\right) \left[\sum_{k=0}^{d-1} \sum_{i=0}^{2d-2-k} (-1)^i \binom{d-1}{k} \binom{2d-2-k}{i} \times (-w)^{k+i}\; 2^{-k/2-3i/2} \left(\frac{d\;\Gamma(d/2)\;\pi^{d/2-1}}{\mathrm{tr}(\Sigma)}\right)^{k/2+i/2} \Gamma\left(\frac{2d-1-k-i}{2}, \frac{d\;\Gamma(d/2)\;\pi^{d/2-1}}{8\,\mathrm{tr}(\Sigma)}\; w^2\right)\right] \tag{7}$$

Both $W = w$ and $W = -w$ yield the same value for $|W|$, and therefore the pdf for $|W|$ obeys $p_{_{|W|}}(w) = p_{_W}(w) + p_{_W}(-w) = 2p_{_W}(w)$. Which means, the pdf for the dissimilarity increments is given by

$$p_{_{|W|}}(w) = 2^{3/2-d} \left(\frac{d\;\pi^{d/2-1}}{\mathrm{tr}(\Sigma)\;\Gamma(d/2)^3}\right)^{1/2} \exp\left(-\frac{d\;\Gamma(d/2)\;\pi^{d/2-1}}{8\,\mathrm{tr}(\Sigma)}\; w^2\right) \left[\sum_{k=0}^{d-1} \sum_{i=0}^{2d-2-k} (-1)^i \binom{2d-2-k}{i} \times \binom{d-1}{k} w^{k+i}\; 2^{-k/2-3i/2} \left(\frac{d\;\Gamma(d/2)\;\pi^{d/2-1}}{\mathrm{tr}(\Sigma)}\right)^{k/2+i/2} \Gamma\left(\frac{2d-1-k-i}{2}, \frac{d\;\Gamma(d/2)\;\pi^{d/2-1}}{8\,\mathrm{tr}(\Sigma)}\; w^2\right)\right], \tag{8}$$

where

$$\Gamma(a,z) = \begin{cases} e^{-z} \displaystyle\sum_{k=0}^{a-1/2-1} \frac{\Gamma(a)}{\Gamma(a-k)} z^{a-1-k} + \Gamma(a)\,\mathrm{erfc}(\sqrt{z}) & \text{if } a-1/2 \in \mathbb{Z}^+ \\ (a-1)!\; e^{-z} \displaystyle\sum_{k=0}^{a-1} \frac{z^k}{k!} & \text{if } a \in \mathbb{Z}^+ \end{cases}$$

B.3 Empirical Estimation using the Expected Value

Pratically, $\mathrm{tr}(\Sigma)$ is sensitive to outliers, so we will rewrite the pdf (8) to be dependent of the mean of the dissimilarity increments. Equivalently, we need to solve

$$\mathbb{E}[w] = \int_0^{\infty} w p_{_W}(w)\; \mathrm{d}w$$

Assume $A = d\;\Gamma(d/2)\;\pi^{d/2-1}\;\left(2\,\mathrm{tr}(\Sigma)\right)^{-1}$.

$$\mathbb{E}[w] = 2^{2-d} A^{1/2}\;\Gamma(d/2)^{-2} \left[\sum_{k=0}^{d-1} \sum_{i=0}^{2d-2-k} (-1)^i \binom{d-1}{k} \binom{2d-2-k}{i}\; 2^{-i}\; A^{k/2+i/2} \times \int_0^{\infty} w^{k+i+1}\; \exp\left(-\frac{A}{4}\; w^2\right)\; \Gamma\left(\frac{2d-k-i-1}{2}, \frac{A}{4}\; w^2\right)\; \mathrm{d}w\right],$$

We will use the following formula [32]

$$\int_0^\infty x^{a-1}e^{-sx}\,\Gamma(b,x)\,\mathrm{d}x = \frac{\Gamma(a+b)}{a(1+s)^{a+b}}\,{}_2F_1\left(1,a+b;1+a;\frac{s}{1+s}\right),$$
$$\text{with } Re(s)>-1,\ Re(a+b)>0,\ Re(a)>0 \tag{9}$$

where ${}_2F_1(a,b;c;z)$ is the hypergeometric function defined by

$${}_2F_1(a,b;c;z)\equiv F(a,b;c;z)=\sum_{n=0}^{\infty}\frac{(a)_n(b)_n}{(c)_n}\frac{z^n}{n!}$$

on the disk $|z|<1$. In general, $F(a,b;c;z)$ does not exist when $c=0,-1,-2,\ldots$. $(a)_n$ is the Pochhammer's symbol

$$(a)_0=1$$
$$(a)_n=a(a+1)(a+2)\ldots(a+n-1)$$

Firstly, let's make the change of variables, assume $x=\frac{A}{4}w^2$, i.e., $w=2A^{-1/2}x^{1/2}$. The interval of integration with the new variable is $[0,\infty)$ and $\frac{dw}{dx}=A^{-1/2}x^{-1/2}$. Now the integral can be written as

$$\int_0^\infty A^{-1/2}\,x^{-1/2}(2A^{-1/2}x^{1/2})^{k+i+1}\,e^{-x}\,\Gamma\left(\frac{2d-k-i-1}{2},x\right)\,\mathrm{d}x$$
$$=\int_0^\infty 2^{k+i+1}A^{-k/2-i/2-1}\,x^{k/2+i/2}\,e^{-x}\,\Gamma\left(\frac{2d-k-i-1}{2},x\right)\,\mathrm{d}x$$

Since $0\le k\le d-1$ and $0\le i\le 2d-k-2$ and $d\ge 2$, we have

$$s=1>-1$$
$$a=k/2+i/2+1>0$$
$$a+b=k/2+i/2+1+d-k/2-i/2-1/2=d+1/2>0$$

which means we are in the conditions of equation (9), so

$$2^{k+i+1}A^{-k/2-i/2-1}\int_0^\infty x^{k/2+i/2}\,e^{-x}\,\Gamma\left(\frac{2d-k-i-1}{2},x\right)\,\mathrm{d}x$$
$$=2^{k+i+1}A^{-k/2-i/2-1}\,\frac{\Gamma(d+1/2)}{(k/2+i/2+1)2^{d+1/2}}\,F\left(1,d+1/2;k/2+i/2+2;\frac{1}{2}\right)$$

Now we need to use the Euler's tranformation formula defined by

$$F(a,b;c;z)=(1-z)^{c-a-b}\,F(c-a,c-b;c;z),$$

we have, in our case,

$$c-a=k/2+i/2+1$$
$$c-b=k/2+i/2-d+3/2$$
$$c-a-b=k/2+i/2-d+1/2$$

which means

$$2^{k+i+1}A^{-k/2-i/2-1}\int_0^\infty x^{k/2+i/2}\,e^{-x}\,\Gamma\left(\frac{2d-k-i-1}{2},x\right)\,\mathrm{d}x$$
$$=2^{k+i+1}A^{-k/2-i/2-1}\,\frac{\Gamma(d+1/2)}{(k/2+i/2+1)2^{d+1/2}}\,\left(\frac{1}{2}\right)^{k/2+i/2-d+1/2}\times$$
$$F\left(\frac{k+i}{2}+1,\frac{k+i+3}{2}-d;\frac{k+i}{2}+2;\frac{1}{2}\right)$$

Also, the incomplete beta function is related to the hypergeometric function, and is defined by

$$B_z(p,q) = \frac{z^p}{p}\, F(p, 1-q; p+1; z),$$

with $p > 0$, $q > 0$ and $0 \leq z \leq 1$.

In our case, $p = \frac{k+i}{2} + 1 > 0$, $q = d - \frac{k+i+1}{2} = d - p + 1/2 > 0$ and $0 \leq \frac{1}{2} \leq 1$, we can write

$$\begin{aligned} 2^{k+i+1} A^{-k/2-i/2-1} \int_0^\infty x^{k/2+i/2}\, e^{-x}\, \Gamma\left(\frac{2d-k-i-1}{2}, x\right)\, \mathrm{d}x \\ = 2^{k+i+1} A^{-k/2-i/2-1}\, \frac{\Gamma(d+1/2)}{(k/2+i/2+1)2^{d+1/2}}\, \left(\frac{1}{2}\right)^{k/2+i/2-d+1/2} \times \\ \frac{\frac{k+i}{2}+1}{\left(\frac{1}{2}\right)^{k/2+i/2+1}}\, B_{\frac{1}{2}}\left(\frac{k+i}{2}+1, d-\frac{k+i+1}{2}\right), \end{aligned}$$

where the incomplete beta function is defined by

$$\begin{cases} B_z(m+1, d-m-1/2) = 2\displaystyle\sum_{j=0}^{m} \binom{m}{j} (-1)^j\, \frac{1-(1-z)^{d-m-1/2+j}}{2d-2m-1+2j} \text{ if } m = 0, 1, \ldots, d-1 \\ B_z(m+3/2, d-m-1) = \displaystyle\sum_{j=0}^{d-m-2} \binom{d-m-2}{j} (-1)^j\, \frac{z^{m+3/2+j}}{m+3/2+j} \text{ if } m = 0, 1, \ldots, d-2 \end{cases}$$

Proof: The incomplete beta function is defined by

$$B_z(a,b) = \int_0^z t^{a-1}(1-t)^{b-1}\, \mathrm{d}t.$$

Firstly, if $m+1 \in \mathbb{N}$ with $m = 0, 1, \ldots, d-1$, then

$$B_z(m+1, d-m-1/2) = \int_0^z t^m (1-t)^{d-m-3/2}\, \mathrm{d}t.$$

Let's make the change of variables, assume $u = (1-t)^{1/2}$. The interval of integration with the new variable is $\left[(1-z)^{1/2}, 1\right]$ and $\frac{dt}{du} = -2u$. Now the integral can be written as

$$\begin{aligned} B_z(m+1, d-m-1/2) &= -\int_{(1-z)^{1/2}}^1 (1-u^2)^m\, u^{2d-2m-3}(-2u)\, \mathrm{d}u \\ &= 2\int_{(1-z)^{1/2}}^1 (1-u^2)^m\, u^{2d-2m-2}\, \mathrm{d}u. \end{aligned}$$

Using the binomial formula we can writte $(1-u^2)^m = \sum_{j=0}^m \binom{m}{j}(-1)^j u^{2j}$, therefore

$$\begin{aligned} B_z(m+1, d-m-1/2) &= 2\sum_{j=0}^{m} \binom{m}{j} (-1)^j \int_{(1-z)^{1/2}}^1 u^{2d-2m-2+2j}\, \mathrm{d}u \\ &= 2\sum_{j=0}^{m} \binom{m}{j} (-1)^j \left[\frac{u^{2d-2m-1+2j}}{2d-2m-1+2j}\right]_{u=(1-z)^{1/2}}^{u=1} \\ &= 2\sum_{j=0}^{m} \binom{m}{j} (-1)^j\, \frac{1-(1-z)^{d-m-1/2+j}}{2d-2m-1+2j} \end{aligned}$$

Now, if $m + 3/2 \in \mathbb{R}^+$ with $m = 0, 1, \ldots, d-2$, then

$$B_z(m + 3/2, d - m - 1) = \int_0^z t^{m+1/2}(1-t)^{d-m-2}\,\mathrm{d}t.$$

Using the binomial formula we have $(1-t)^{d-m-2} = \sum_{j=0}^{d-m-2} \binom{d-m-2}{j}(-1)^j\, t^j$, therefore

$$\begin{aligned} B_z(m + 3/2, d - m - 1) &= \sum_{j=0}^{d-m-2} \binom{d-m-2}{j}(-1)^j \int_0^z t^{m+1/2+j}\,\mathrm{d}t \\ &= \sum_{j=0}^{d-m-2} \binom{d-m-2}{j}(-1)^j \left[\frac{t^{m+3/2+j}}{m+3/2+j}\right]_{t=0}^{t=z} \\ &= \sum_{j=0}^{d-m-2} \binom{d-m-2}{j}(-1)^j \frac{z^{m+3/2+j}}{m+3/2+j} \qquad \square \end{aligned}$$

Therefore,

$$\mathbb{E}[w] = 2^{2-d} A^{-1/2}\,\Gamma(d/2)^{-2}\,\Gamma(d+1/2)\left[\sum_{k=0}^{d-1}\sum_{i=0}^{2d-2-k}(-1)^i \binom{d-1}{k}\binom{2d-2-k}{i} 2^{k+1} \times B_{\frac{1}{2}}\left(\frac{k+i}{2}+1, d-\frac{k+i+1}{2}\right)\right]$$

where

$$\begin{cases} B_{\frac{1}{2}}(m+1, d-m-1/2) = 2\displaystyle\sum_{j=0}^{m}\binom{m}{j}(-1)^j\,\frac{1-2^{-d+m+1/2-j}}{2d-2m-1+2j} \text{ if } m = 0, 1, \ldots, d-1 \\ B_{\frac{1}{2}}(m+3/2, d-m-1) = \displaystyle\sum_{j=0}^{d-m-2}\binom{d-m-2}{j}(-1)^j\,\frac{2^{-m-3/2-j}}{m+3/2+j} \text{ if } m = 0, 1, \ldots, d-2 \end{cases}$$

Recall that $A = d\,\Gamma(d/2)\,\pi^{d/2-1}\,(2\,\mathrm{tr}(\Sigma))^{-1}$, so we can write

$$\mathbb{E}[w] = 2^{2-d}\left(d\,\Gamma(d/2)\,\pi^{d/2-1}\,(2\,\mathrm{tr}(\Sigma))^{-1}\right)^{-1/2}\,\Gamma(d/2)^{-2}\,\Gamma(d+1/2)\left[\sum_{k=0}^{d-1}\sum_{i=0}^{2d-2-k}(-1)^i \times \binom{d-1}{k}\binom{2d-2-k}{i} 2^{k+1} B_{\frac{1}{2}}\left(\frac{k+i}{2}+1, d-\frac{k+i+1}{2}\right)\right]$$

and we want to find an expression for $\mathrm{tr}(\Sigma)$, therefore

$$\mathrm{tr}(\Sigma)^{1/2} = \mathbb{E}[w]\,2^{d-5/2}\,d^{1/2}\,\pi^{d/4-1/2}\,\Gamma(d/2)^{5/2}\,\Gamma(d+1/2)^{-1}\left[\sum_{k=0}^{d-1}\sum_{i=0}^{2d-2-k}(-1)^i \times \binom{d-1}{k}\binom{2d-2-k}{i} 2^{k+1} B_{\frac{1}{2}}\left(\frac{k+i}{2}+1, d-\frac{k+i+1}{2}\right)\right]^{-1}$$

C Subset of Submitted and Published Papers

[2] - (SIMBAD Technical Report n. 2010_26)

Bulò, S., Lourenço, A., Fred, A., Pelillo, M.: Pairwise probabilistic clustering using evidence accumulation. In Hancock, E., Wilson, R., Windeatt, T., Ulusoy, I., Escolano, F., eds.: Structural, Syntactic, and Statistical Pattern Recognition. Volume 6218 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2010) 395–404 10.1007/978-3-642-14980-1_38.

Pairwise Probabilistic Clustering Using Evidence Accumulation

Samuel Rota Bulò[1], André Lourenço[3], Ana Fred[2,3] and Marcello Pelillo[1]

[1] Dipartimento di Informatica - University of Venice - Italy
{srotabul,pelillo}@dsi.unive.it
[2] Instituto Superior Técnico - Lisbon - Portugal
[3] Instituto de Telecomunicações - Lisbon - Portugal
{arlourenco,afred}@lx.it.pt

Abstract. In this paper we propose a new approach for consensus clustering which is built upon the evidence accumulation framework. Our method takes the co-association matrix as the only input and produces a soft partition of the dataset, where each object is probabilistically assigned to a cluster, as output. Our method reduces the clustering problem to a polynomial optimization in probability domain, which is attacked by means of the Baum-Eagon inequality. Experiments on both synthetic and real benchmarks data, assess the effectiveness of our approach.

1 Introduction

There is a close connection between the concepts of pairwise similarity and probability in the context of unsupervised learning. It is a common assumption that, if two objects are similar, it is very likely that they are grouped together by some clustering algorithm, the higher the similarity, the higher the probability of co-occurrence in a cluster. Conversely, if two objects co-occur very often in the same cluster (high co-occurrence probability), then it is very likely that they are very similar. This duality and correspondence between pairwise similarity and pairwise probability within clusters forms the core idea of the clustering ensemble approach known as evidence accumulation clustering (EAC) [1].

Evidence accumulation clustering combines the results of multiple clusterings into a single data partition by viewing each clustering result as an independent evidence of pairwise data organization. Using a pairwise frequency count mechanism amongst a clustering committee, the method yields, as an intermediate result, a co-association matrix that summarizes the evidence taken from the several members in the clustering ensemble. This matrix corresponds to the maximum likelihood estimate of the probability of pairs of objects being in the same group, as assessed by the clustering committee. One of the main advantages of EAC is that it allows for a big diversification within the clustering committee. Indeed, no assumption is made about the algorithms used to produce the data partitions, it is robust to incomplete information, i.e., we may include partitions over sub-sampled versions of the original data set, and no restriction is made on the number of clusters of the partitions.

Once a co-association matrix is produced according to the EAC framework, a consensus clustering is obtained by applying a clustering algorithm, which typically induces a hard partition, to the co-association matrix. Although having crisp partitions as baseline for the accumulation of evidence of data organization is reasonable, this assumption is too restrictive in the phase of producing a consensus clustering. This is for instance the case for many important applications such as clustering micro-array gene expression data, text categorization, perceptual grouping, labeling of visual scenes and medical diagnosis. In fact, the importance of dealing with overlapping clusters has been recognized long ago [2] and recently, in the machine learning community, there has been a renewed interest around this problem [3, 4]. Moreover, by inducing hard partitions we loose important information like the level of uncertainty of each label assignment. It is also worth considering that the underlying clustering criteria of ad hoc algorithms do not take advantage of the probabilistic interpretation of the computed similarities, which is an intrinsic part of the EAC framework.

In this paper we propose a new approach for consensus clustering which is built upon the evidence accumulation framework. Our idea was inspired by a recent work due to Zass and Sashua [5]. Our method takes the co-association matrix as the only input and produces a soft partition of the data set, where each object is probabilistically assigned to a cluster, as output. In order to find the unknown cluster assignments, we fully exploit the fact that each entry of the co-association matrix is an estimation of the probability of two objects to be in a same cluster, which is derived from the ensemble of clusterings. Indeed, it is easy to see that under reasonable assumptions, the probability that two objects i and j will occur in the same cluster is a function of the unknown cluster assignments of i and j. By minimizing the divergence between the estimation derived from the co-association matrix and this function of the unknowns, we obtain the result of the clustering procedure. More specifically, our method reduces the clustering problem to a polynomial optimization in the probability domain, which is attacked by means of the Baum-Eagon inequality [6]. This inequality, indeed, provides us with a class of nonlinear transformations that serve our purpose. In order to assess the effectiveness of our findings we conducted experiments on both synthetic and real benchmark data sets.

2 A probabilistic model for clustering

Let $O = \{1, \ldots, n\}$ be a set of data objects (or simply objects) to cluster into K classes and let $\mathcal{E} = \{cl_i\}_{i=1}^{N}$ be an ensemble of N clusterings of O obtained by running different algorithms with different parameterizations on (possibly) sub-sampled versions of the original data set O. Data sub-sampling is herein put forward as a most general framework for the following reasons: it favors the diversification of the clustering ensemble; it models situations of distributed clustering where local clusterers have only partial access to the data; by using this type of data perturbation, the co-association matrix has an additional inter-

pretation of pairwise stability that can further be used for the purpose of cluster validation [7].

Each clustering in the ensemble $\mathcal{E}$ is a function $cl_i : O_i \to \{1, \ldots, K_i\}$ from the set of objects $O_i \subseteq O$ to a class label. For the afore-mentioned reasons, O_i is a subset of the original data set O and, moreover, each clustering may assume a different number of classes K_i. We denote by Ω_{ij} the indices of the clusterings where i and j have been classified, which is given by

$$\Omega_{ij} = \{p = 1 \ldots N : i, j \in O_p\} .$$

Consider also $N_{ij} = |\Omega_{ij}|$, where $|\cdot|$ provides the cardinality of the argument, which is the number of clusterings where i and j have been both classified.

The aim of our work is to learn, from the ensemble of clusterings $\mathcal{E}$, how to cluster the objects into K classes, without having, in principle, any other information about the objects we are going to cluster. To this end, we start from the assumption that objects can be softly assigned to clusters. Hence, the clustering problem consists in estimating, for each object $i \in O$, an unknown assignment $\mathbf{y}_i$, which is a probability distribution over the set of cluster labels $\{1, \ldots, K\}$, or, in other words, an element of the *standard simplex* Δ_K given by

$$\Delta_K = \{\boldsymbol{x} \in \mathrm{R}_+^K : \|\mathbf{x}\|_1 = 1\} ,$$

where R_+ is the set of nonnegative reals, and $\|\cdot\|_1$ is the ℓ^1-norm. The kth entry of $\mathbf{y}_i$ thus provides the probability of object i to be assigned to cluster k. Given the unknown cluster assignments $\mathbf{y}_i$ and $\mathbf{y}_j$ of objects i and j, respectively, and assuming independent cluster assignments, the probability of them to occur in a same cluster can be easily derived as $\mathbf{y}_i^\top \mathbf{y}_j$. Suppose now $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_n) \in \Delta_K^n$ to be the matrix formed by stacking the $\mathbf{y}_i$'s, which in turn form the columns of Y. Then, the $n \times n$ matrix $Y^\top Y$ provides the co-occurrence probability of any pair of objects in O.

For each pair of objects i and j, let X_{ij} be a Bernoulli distributed random variable (r.v.) indicating whether objects i and j occur in a same cluster. Note that, according to our model, the mean (and therefore the parameter) of X_{ij} is $\mathbf{y}_i^\top \mathbf{y}_j$, i.e., the probability of co-occurrence of i and j. For each pair of objects i and j, we collect from the clusterings ensemble N_{ij} independent realizations $x_{ij}^{(p)}$ of X_{ij}, which are given by:

$$x_{ij}^{(p)} = \begin{cases} 1 & \text{if } cl_p(i) = cl_p(j) , \\ 0 & \text{otherwise} . \end{cases}$$

for $p \in \Omega_{ij}$. By taking their mean, we obtain the empirical probability of co-occurrence c_{ij}, which is the fraction of times objects i and j have been assigned to a same cluster:

$$c_{ij} = \frac{1}{N_{ij}} \sum_{p \in \Omega_{ij}} x_{ij}^{(p)} .$$

The matrix $C = (c_{ij})$, derived from the empirical probabilities of co-occurrence of any pair of objects, is known as the *co-association matrix* within the evidence

accumulation-based framework for clustering [8, 1]. Since C is the maximum likelihood estimate of $Y^\top Y$ given the observations from the clustering ensemble $\mathcal{E}$, we will refer to the former as the *empirical co-association matrix*, and to the latter as the *true co-association matrix*.

At this point, by minimizing the divergence, in a least-square sense, of the true co-association matrix from the empirical one, with respect to Y, we find a solution Y^* of the clustering problem. This leads to the following optimization problem:

$$\begin{aligned} Y^* = \arg\min \quad & \|C - Y^\top Y\|_F^2 \\ \text{s.t.} \quad & Y \in \Delta_K^n \,. \end{aligned} \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm. Note that Y^* provides us with soft assignments of the objects to the K classes. Indeed, y^*_{ki} gives the probability of object i to be assigned to class k. If a hard partition is needed, this can be forced by assigning each object i to the highest probability class, which is given by: $\arg\max_{k=1\ldots K}\{y^*_{ki}\}$. Moreover, by computing the entropy of each $\mathbf{y}_i$, we can obtain an indication of the uncertainty of the cluster assignment for object i.

3 Related Work

In [5] a similar approach is proposed for pairwise clustering. First of all, a preprocessing on the similarity matrix W looks for its closest doubly-stochastic matrix F under ℓ_1 norm, or Frobenius norm, or relative entropy [9]. The k-clustering problem is then tackled by finding a completely-positive factorization of $F = (f_{ij})$ in the least-square sense, i.e., by solving the following optimization problem:

$$\begin{aligned} G^* = \arg\min \quad & \|F - G^\top G\|_F^2 \\ \text{s.t.} \quad & G \in \mathrm{R}_+^{k\times n} \,. \end{aligned} \tag{2}$$

Note that this leads to an optimization program, which resembles (1), but is inherently different. The elements g_{ri} of the resulting matrix G provide an indication of object i to be assigned to class r. However, unlike our formulation, these quantities are not explicit probabilities and it may happen for instance that $g_{ri} = 0$ for all $r = 1 \ldots k$, i.e., some objects may remain in principle unclassified.

The approach proposed to find a local solution of (2) consists in iterating the following updating rule:

$$g_{ri} \leftarrow \frac{g_{ri} \sum_{j\neq i}^n g_{rj} f_{ij}}{\sum_{s=1}^k g_{si} \sum_{j\neq i}^n g_{sj} g_{rj}} \,.$$

The computational complexity for updating all entries in G once (complete iteration) is $O(kn^2)$, while we expect to find a solution in $O(\gamma kn^2)$, where γ is the average number of complete iterations required to converge. A disadvantage of this iterative scheme is that updates must be sequential, i.e., we cannot update all entries of G in parallel.

4 The Baum-Eagon inequality

In the late 1960s, Baum and Eagon [6] introduced a class of nonlinear transformations in probability domain and proved a fundamental result which turns out to be very useful for the optimization task at hand. The next theorem introduces what is known as the Baum-Eagon inequality.

Theorem 1 (Baum-Eagon). *Let $X = (x_{ri}) \in \Delta_k^n$ and $Q(X)$ be a homogeneous polynomial in the variables x_{ri} with nonnegative coefficients. Define the mapping $Z = (z_{ri}) = \mathcal{M}(X)$ as follows:*

$$z_{ri} = x_{ri} \frac{\partial Q(X)}{\partial x_{ri}} \Big/ \sum_{s=1}^{k} x_{si} \frac{\partial Q(X)}{\partial x_{si}} \,, \tag{3}$$

for all $i = 1 \dots n$ and $r = 1 \dots k$. Then $Q(\mathcal{M}(X)) > Q(X)$, unless $\mathcal{M}(X) = X$. In other words $\mathcal{M}$ is a growth transformation for the polynomial Q.

This result applies to homogeneous polynomials, however in a subsequent paper, Baum and Sell [10] proved that Theorem 1 still holds in the case of arbitrary polynomials with nonnegative coefficients, and further extended the result by proving that $\mathcal{M}$ increases Q homotopically, which means that for all $0 \leq \eta \leq 1$, $Q(\eta\mathcal{M}(X) + (1-\eta)X) \geq Q(X)$ with equality if and only if $\mathcal{M}(X) = X$.

The Baum-Eagon inequality provides an effective iterative means for maximizing polynomial functions in probability domains, and in fact it has served as the basis for various statistical estimation techniques developed within the theory of probabilistic functions of Markov chains [11]. It is indeed not difficult to show that, by starting from the interior of the simplex, the fixed points of the Baum-Eagon dynamics satisfy the first-order Karush-Kuhn-Tucker necessary conditions for local maxima and that we have a strict local solution in correspondence to asymptotically stable point.

5 The algorithm

In order to use the Baum-Eagon theorem for optimizing (1) we need to meet the requirement of having a polynomial to maximize with nonnegative coefficients in the simplex-constrained variables. To this end, we consider the following optimization program, which is proved to be equivalent to (1):

$$\begin{aligned} \max \quad & 2Tr(CY^\top Y) + \|Y^\top E_K Y\|^2 - \|Y^\top Y\|^2 \\ \text{s.t.} \quad & Y \in \Delta_K^n \,, \end{aligned} \tag{4}$$

where E_K is the $K \times K$ matrix of all 1's, and $Tr(\cdot)$ is the matrix trace function.

Proposition 1. *The maximizers of* (4) *are minimizers of* (1) *and vice versa. Moreover, the objective function of* (4) *is a polynomial with nonnegative coefficients in the variables y_{ki}, which are elements of Y.*

Proof. Let $P(Y)$ and $Q(Y)$ be the objective functions of (1) and (4), respectively.

To prove the second part of the proposition note that trivially every term of the polynomial $\|Y^\top Y\|^2$ is also a term of $\|Y^\top E_K Y\|^2$. Hence, $Q(Y)$ is a polynomial with nonnegative coefficients in the variables y_{ki}.

As for the second part, by simple algebra, we can write $Q(Y)$ in terms of $P(Y)$ as follows:

$$\begin{aligned} Q(Y) &= \|C\|^2 - P(Y) + \|Y^\top E_K Y\|^2 \\ &= \|C\|^2 - P(Y) + 1\,, \end{aligned}$$

where we used the fact that $\|Y^\top E_K Y\| = 1$. Note that the removal of the constant terms from $Q(Y)$ leaves its maximizers over Δ_K^n unaffected. Therefore, maximizers of (4) are also maximizers of $-P(Y)$ over Δ_K^n and thus minimizers of (1). This concludes the proof.

By Proposition 1 we can use Theorem 1 to locally optimize (4). This allows us to find a solution of (1). Note that, in our case, the objective function is not a homogeneous polynomial but, as mentioned previously, this condition is not necessary [10]. By applying (3), we obtain the following updating rule for $Y = (y_{ki})$:

$$y_{ki}^{(t+1)} = y_{ki}^{(t)} \frac{n + [Y(C - Y^\top Y)]_{ki}}{n + \sum_k y_{ki}^{(t)} [Y(C - Y^\top Y)]_{ki}}\,, \tag{5}$$

where we abbreviated $Y^{(t)}$ with Y and any non-constant iteration of (5) strictly decreases the objective function of (1).

The computational complexity of the proposed dynamics is $O(\gamma k n^2)$, where γ is the average number of iterations required to converge (note that in our experiments we kept γ fixed). One remarkable advantage of this dynamics is that it can be easily parallelized in order to benefit from modern multi-core processors. Additionally, it can be easily implemented with few lines of Matlab code.

6 Experiments

We conducted experiments on different real data-sets from the UCI Machine Learning Repository: iris, house-votes, std-yeast-cell and breast-cancer. Additionally, we considered also the image-complex synthetic data-set, shown in figure 1. For each data-set, we produced the clustering ensemble $\mathcal{E}$ by running different clustering algorithms, with different parameters, on subsampled versions of the original data-set (the sampling rate was fixed to 0.9). The clustering algorithms used to produce the ensemble were the following [12]: Single Link (SL), Complete Link (CL), Average Link (AL) and K-means (KM).

Table 1 summarizes the experimental setting that has been considered. For each data-set, we report the optimal number of clusters K and the size n of the data-set, respectively. As for the ensemble, each algorithm was run several times in order to produce clusterings with different number of classes, K_i. For each

Data-Sets	K	n	Ensemble
			K_i
iris	3	150	3-10,15,20
house-votes	2	232	2-10,15,20
std-yeast-cell	5	384	5-10,15,20
breast-cancer	2	683	2-10,15,20
image-complex	8	1000	8-15,20,30, 37

Table 1. Benchmark data-sets and parameter values used with different clustering algorithms (see text for description).

clustering approach and each parametrization of the same we generated $N = 100$ different subsampled versions of the data-set.

Once all the clusterings have been generated, we grouped them by algorithm into several *base ensembles*, namely $\mathcal{E}_{\mathrm{SL}}$, $\mathcal{E}_{\mathrm{AL}}$, $\mathcal{E}_{\mathrm{CL}}$ and $\mathcal{E}_{\mathrm{KM}}$. Moreover, we created a large ensemble $\mathcal{E}_{\mathrm{All}}$ from the union of all of them. For each ensemble we created a corresponding co-association matrix, namely C_{SL}, C_{AL}, C_{CL}, C_{KM} and C_{All}. For each of these co-association matrices, we applied our Pairwise Probabilistic Clustering (PPC) approach, and compared it against the performances obtained with the same matrices by the agglomerative hierarchical algorithms SL, AL and CL. Each method was provided with the optimal number of classes as input parameter.

Figure 2 summarizes the results obtained over the benchmark data-sets. The performances are assessed in terms of accuracy, i.e., the percentage of correct labels. When we consider the base ensembles, i.e., $\mathcal{E}_{\mathrm{SL}}$, $\mathcal{E}_{\mathrm{AL}}$, $\mathcal{E}_{\mathrm{CL}}$ and $\mathcal{E}_{\mathrm{KM}}$, on average our approach achieves the best results, although other approaches, such as the AL, perform comparably well. Our algorithm, however, outperforms the competitors when we take the union $\mathcal{E}_{\mathrm{All}}$ of all the base ensembles into account. Interestingly, the results obtained by PPC on the combined ensemble are as good as the best one obtained in the base ensembles and, in some cases like the image-complex dataset, they are even better.

The different levels of performance obtained by the several algorithms over the different clustering ensembles, as shown in Figures 2(a) to 2(d), are illustra-

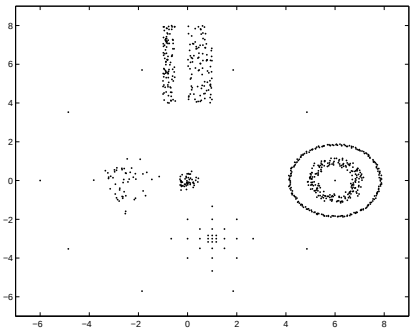
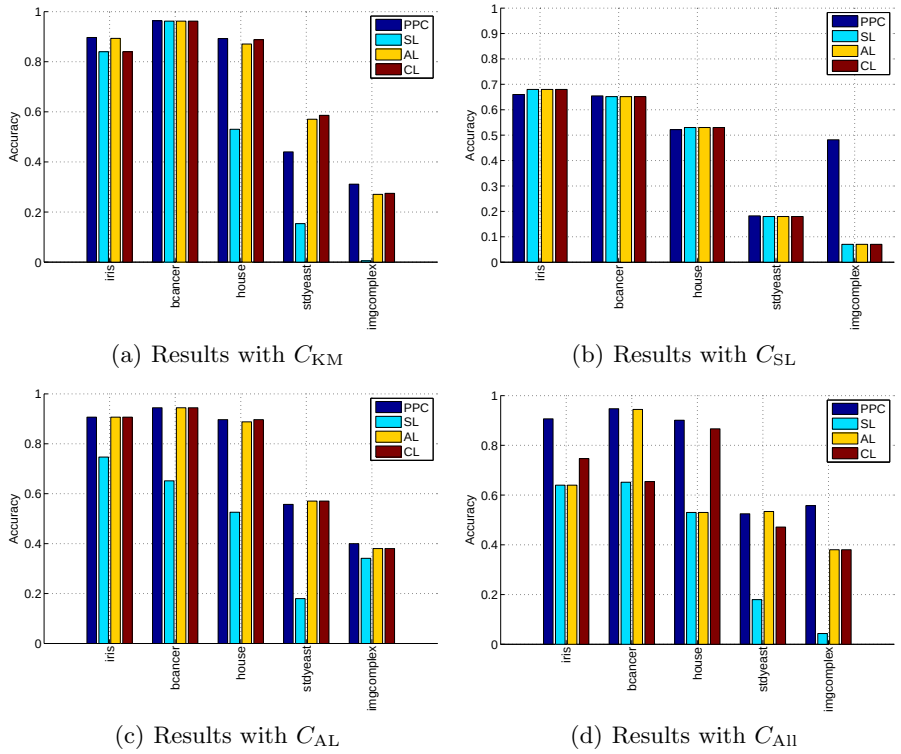


Fig. 1. Image Complex Synthetic data-set.



(a) Results with C_{KM}

(b) Results with C_{SL}

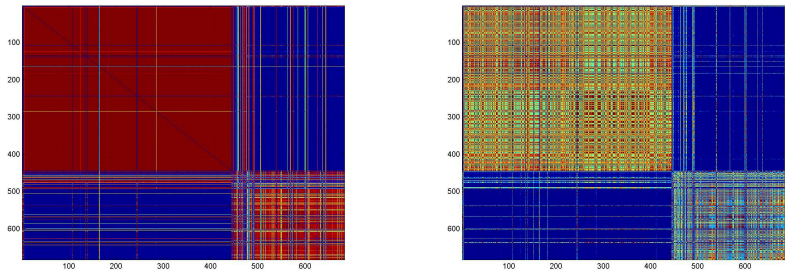
(c) Results with C_{AL}

(d) Results with C_{All}

Fig. 2. Experiments on benchmark data-sets.

tive of the distinctiveness between the underlying clustering ensembles, and the diversity of clustering solutions. It is then clear that the ensemble $\mathcal{E}_{\mathrm{All}}$ has the largest diversity when compared to the individual ensembles; this is quantitatively confirmed when computing average pairwise consistency values between partitions in the individual CEs and the one resulting by the merging of these. This higher diversity causes the appearance of noisy-like structure in the co-association matrices. This is illustrated in Figures 3(a) and 3(b) corresponding to the co-association matrices C_{AL} and C_{KM}, respectively, when compared to the C_{All} in Figure 4(a). The better performance of the PPC algorithm on the latter CE, can be attributed to a leveraging effect over these local noisy estimates, thus better unveiling the underlying structure of the data. This is illustrated next.

Figures 4(a) and 4(b) show the empirical co-association matrix C_{All} and the true one, respectively, for the breast-cancer data-set. While the block structure of two clusters is apparent in both figures, we can see that the true co-association turns out to be less noisy than the empirical one. In Figure 4(c) we plot the soft cluster assignments, Y. Here, object indices are on the x-axis, and probabilities are on the y-axis, each curve representing the profile of a cluster. As one can see from the cluster memberships, the two clusters can be clearly evinced, although there is a higher level of uncertainty in the assignments of objects belonging to



(a) C_{AL} (b) C_{KM}

Fig. 3. Co-association matrices with ensembles $\mathcal{E}_{AL}$ and $\mathcal{E}_{KM}$.

the smallest cluster. Indeed, this can also be seen in Figure 4(d), where we plot the uncertainty h_i in the cluster assignments, which is computed for each object i as the normalized entropy of $\mathbf{y}_i$, i.e.,

$$h_i = -\frac{\sum_{k=1}^{K} y_{ki} \log(y_{ki})}{\log(K)} \,.$$

7 Conclusion

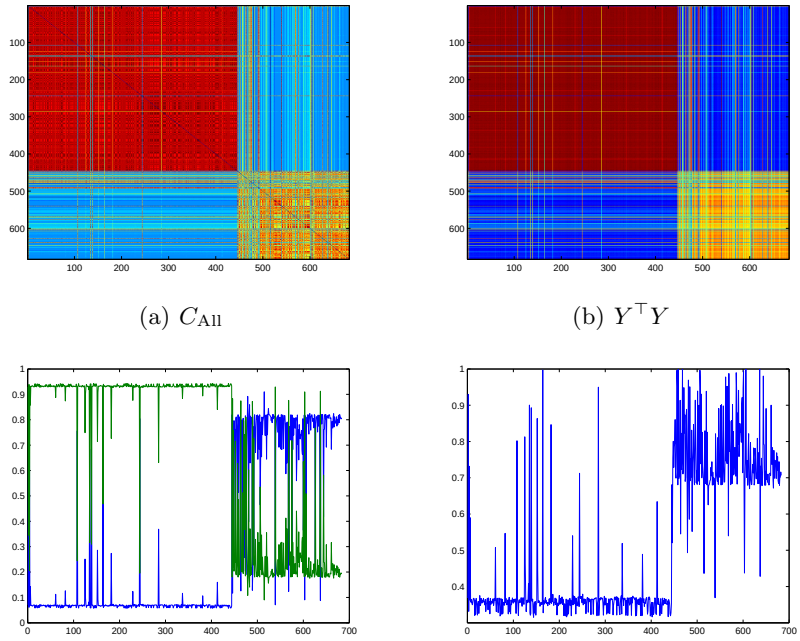
In this paper we introduced a new approach for consensus clustering. Taking advantage of the probabilistic interpretation of the computed similarities of the the co-association matrix, derived from the ensemble of clusterings, using the Evidence Accumulation Clustering, we propose a principled soft clustering method. Our method reduces the clustering problem to a polynomial optimization in probability domain, which is attacked by means of the Baum-Eagon inequality. Experiments on both synthetic and real benchmarks assess the effectiveness of our approach.

Acknowledgement

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). This work was partially supported by the Portuguese Foundation for Science and Technology (FCT), Portuguese Ministry of Science and Technology, under grant PTDC/ EIACCO/ 103230/ 2008.

References

1. Fred, A., Jain, A.K.: Combining multiple clustering using evidence accumulation. IEEE Trans. Pattern Anal. Machine Intell. **27**(6) (2005) 835–850
2. Jardine, N., Sibson, R.: The construction of hierarchic and non-hierarchic classifications. Computer J. **11** (1968) 177–184



(a) C_{All} (b) $Y^\top Y$

(c) Y (d) Uncertainty

Fig. 4. Results on the breast-cancer data-set.

3. Banerjee, A., Krumpelman, C., Basu, S., Mooney, R.J., Ghosh, J.: Model-based overlapping clustering. In: Int. Conf. on Knowledge Discovery and Data Mining. (2005) 532 – 537
4. Heller, K., Ghahramani, Z.: A nonparametric bayesian approach to modeling overlapping clusters. In: Int. Conf. AI and Statistics. (2007)
5. Zass, R., Shashua, A.: A unifying approach to hard and probabilistic clustering. In: Int. Conf. Comp. Vision (ICCV). Volume 1. (2005) 294–301
6. Baum, L.E., Eagon, J.A.: An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. Bull. Amer. Math. Soc. **73** (1967) 360–363
7. Fred, A., Jain, A.K.: Learning pairwise similarity for data clustering. In: Int. Conf. Patt. Recogn. (ICPR). (2006) 925–928
8. Fred, A., Jain, A.K.: Data clustering using evidence accumulation. In: Int. Conf. Patt. Recogn. (ICPR). (2002) 276–280
9. Zass, R., Shashua, A.: Doubly stochastic normalization for spectral clustering. Adv. in Neural Inform. Proces. Syst. (NIPS) **19** (2006) 1569–1576
10. Baum, L.E., Sell, G.R.: Growth transformations for functions on manifolds. Pacific J. Math. **27** (1968) 221–227
11. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann. Math. Statistics **41** (1970) 164–171
12. Jain, A.K., Dubes, R.C.: Algorithms for data clustering. Prentice-Hall (1988)

[3] - (SIMBAD Technical Report n. 2011_11)

Lourenço, A., Fred, A.L.N., Figueiredo, M.: A generative dyadic aspect model for evidence accumulation clustering. In: SIMBAD workshop. Lecture Notes in Computer Science. Springer (2011) To appear.

A Generative Dyadic Aspect Model for Evidence Accumulation Clustering

André Lourenço[†‡♭], Ana Fred[†‡], and Mário Figueiredo[†‡]

alourenco@deetc.isel.ipl.pt, {afred,mario.figueiredo}@lx.it.pt

[†]Instituto de Telecomunicações

[♭] Instituto Superior de Engenharia de Lisboa

[‡] Instituto Superior Técnico

Lisboa, Portugal

Abstract. Evidence accumulation clustering (EAC) is a clustering combination method in which a pair-wise similarity matrix (the so-called co-association matrix) is learnt from a clustering ensemble. This co-association matrix counts the co-occurrences (in the same cluster) of pairs of objects, thus avoiding the cluster correspondence problem faced by many other clustering combination approaches. Starting from the observation that co-occurrences are a special type of dyads, we propose to model co-association using a generative aspect model for dyadic data. Under the proposed model, the extraction of a consensus clustering corresponds to solving a maximum likelihood estimation problem, which we address using the expectation-maximization algorithm. We refer to the resulting method as *probabilistic ensemble clustering algorithm* (PEnCA). Moreover, the fact that the problem is placed in a probabilistic framework allows using model selection criteria to automatically choose the number of clusters. To compare our method with other combination techniques (also based on probabilistic modeling of the clustering ensemble problem), we performed experiments with synthetic and real benchmark data-sets, showing that the proposed approach leads to competitive results.

Keywords: Unsupervised learning, clustering, clustering Combination, generative models, model Selection

1 Introduction

Although clustering is one of the oldest and most studied problems in statistical data analysis, pattern recognition, and machine learning, it is still far from being considered solved and continues to stimulate a considerable amount of research. Given a set of unlabeled objects, the classical goal of clustering is to obtain a partition of these objects into a set of K classes/groups/clusters (where K itself

may be known or unknown). Numerous clustering algorithms having proposed in the past decades, but none can be considered of general applicability, mainly because each method is intimately attached to a particular answer to the key question that underlies clustering: "what is cluster?". For example, methods designed under the assumption that a cluster is a *compact* set of objects will fail to identify *connected* sets of objects [7].

Clustering combination techniques, which constitute a recent and promising research trend [1], [5], [8], [9], [17], [18], typically outperform stand-alone clustering algorithms and provide a higher degree of adaptability of the cluster structure to the data. The rationale behind clustering combination methods is that, in principle, a "better" and "more robust" partitioning of the data may be achieved by combining the information provided by an ensemble of clusterings than by using a single clustering (or clustering method).

Ensemble-based clustering techniques exploit the diversity of clustering solutions available in an ensemble of partitions, by proposing a consensus partition that leverages individual clustering results. One key aspect of this type of methods is that diversity can be created without any assumption about the data structure or underlying clustering algorithm(s). Moreover, ensemble methods are robust to incomplete information, since they may include partitions obtained from sub-sampled versions of the original dataset, from different data representations, from different clustering algorithms, and no assumptions need to be made about the number of clusters of each partition in the ensemble.

Evidence accumulation clustering (EAC), proposed by Fred and Jain [8], [9], is an ensemble-based method that seeks to find consistent data partitions by considering pair-wise relationships. The method can be decomposed into three major steps:

(i) construction of the clustering ensemble;
(ii) accumulation of the "clustering evidence" provided by ensemble;
(iii) extraction of the final consensus partition from the accumulated evidence.

In the combination/accumulation step (ii), the clustering ensemble is transformed into matrix, termed the *co-occurrence matrix*, where each entry counts the number of clusterings in the ensemble in which each pair of objects were placed in the same cluster. A key feature of EAC is that obtaining the co-occurrence matrix does not involve any type of cluster correspondence, a non-trivial problem with which many other clustering ensemble methods have to deal.

The theory of dyadic data analysis, as defined by Hofmann et al. [13], fits perfectly with the EAC approach. In dyadic data, each elementary observation is a dyad (a pair of objects), possibly complemented with a scalar value expressing strength of association [13]. As explained in detail in Section 2, the co-association matrix obtained in the EAC approach can be interpreted as an aggregation of the information provided by an observed set of pairs of objects, thus can be seen as a dyadic dataset.

Hofmann et al. [13] proposed a systematic, domain independent framework for learning from dyadic data using generative mixture models. In this paper, we

apply those ideas to the EAC formulation, yielding a generative model for the clustering ensemble. In the proposed approach, the consensus partition extraction step naturally consists in solving a *maximum likelihood estimation* (MLE) problem, which is addressed with the *expectation-maximization* (EM) algorithm [4]. We refer to the proposed method as *probabilistic ensemble clustering algorithm* (PEnCA).

One of the advantages of this MLE-based approach is the possibility of inclusion of a model selection criterion to estimate the number of cluster in the consensus partition. To that end, we can use a simple version of the *minimum description length* (MDL) criterion [15] or adaptation for mixtures [6] or even more recent and sophisticated methods [2].

This paper is organized as follows: in Section 2, we present the generative aspect model for the co-association matrix and a maximum likelihood estimation criterion for the consensus partition. Section 4 reviews some related work. Experimental results on both synthetic and real benchmark datasets are presented in Section 5. Finally, Section 6 concludes the paper by drawing some conclusions and giving pointers to future work.

2 Generative Model for Evidence Accumulation Clustering

2.1 Clustering Ensembles and Evidence Accumulation

The goal of evidence accumulation clustering (EAC) is to combine the results of an ensemble of clusterings into a single data partition, by viewing each clustering as an independent piece of evidence about the pairwise organization of the set of objects under study.

Consider a set of N objects $\mathcal{X} = \{1, \ldots, N\}$ to be clustered; without loss of generality[1], we simply index these objects with the integers from 1 to N. A *clustering ensemble* (CE), $\mathbb{P}$, is defined as a set of M different partitions of the set $\mathcal{X}$, that is, $\mathbb{P} = \{\mathcal{P}^1, \ldots, \mathcal{P}^M\}$, where each $\mathcal{P}^i$ is a partition with K_i clusters: $\mathcal{P}^i = \{\mathcal{C}^i_1, \ldots, \mathcal{C}^i_{K_i}\}$. This means that $\mathcal{C}^i_k \subseteq \mathcal{X}$, for any $i = 1, ..., M$ and $k = 1, ..., K_i$, and that the following constraints are satisfied:

$$(k \neq j) \Rightarrow \mathcal{C}^i_k \cap \mathcal{C}^i_j = \emptyset, \quad \text{for } i = 1, ..., M \tag{1}$$

and

$$\bigcup_{k=1}^{K_i} \mathcal{C}^i_k = \mathcal{X}, \quad \text{for } i = 1, ..., M. \tag{2}$$

Although higher-order information could be extracted from $\mathbb{P}$, the EAC approach focuses on the pair-wise information contained in $\mathbb{P}$, which is embodied

[1] Any characteristics of the objects themselves (feature vectors, distances, ...) are only relevant for the individual clusterings of the ensemble and are thus encapsulated under the clustering ensemble obtained and irrelevant for the subsequent steps.

in a sequence $\mathcal{S}$ of all the pairs of objects co-occurring in a common cluster of one of the partitions of the ensemble $\mathbb{P}$. Clearly, the number of pairs in $\mathcal{S}$ is

$$|\mathcal{S}| = \sum_{i=1}^{M} \sum_{k=1}^{K_i} |\mathcal{C}_k^i| \left(|\mathcal{C}_k^i| - 1\right), \tag{3}$$

where $|\mathcal{C}_k^i|$ denotes the number of objects in the k-th cluster of partition $\mathcal{P}^i$. Each element of $\mathcal{S}$ is a pair $(y_m, z_m) \in \mathcal{X} \times \mathcal{X}$, for $m = 1, ..., |\mathcal{S}|$, such that there exists one cluster in one of the partitions, say $\mathcal{C}_k^i$, for which $y_m \neq z_m$, $y_m \in \mathcal{C}_k^i$ and $z_m \in \mathcal{C}_k^i$.

The $(N \times N)$ co-association matrix $\mathbf{C} = [C_{y,z}]$, which is the central element in the EAC approach, collects a statistical summary of $\mathcal{S}$ by counting the number of clusterings in which each pair of objects falls in the same cluster; formally, the element (y, z) of matrix $\mathbf{C}$ is defined as

$$C_{y,z} = \sum_{m=1}^{|\mathcal{S}|} \mathbb{I}\big((y_m, z_m) = (y, z)\big), \quad \text{for } y, z \in \mathcal{X} \tag{4}$$

where $\mathbb{I}$ is the indicator function (equal to one if its argument is a true propostion, and equal to zero if it is a false proposition). Naturally, matrix $\mathbf{C}$ is symmetrical because if some pair $(a, b) \in \mathcal{S}$, then also $(b, a) \in \mathcal{S}$. Because the set $\mathcal{S}$ does not contain pairs with repeated elements (of the form (z, z)), the diagonal elements are all zero.

2.2 Generative Model

Inspired by [11], [12], [13], we adopt a generative model for $\mathcal{S}$, by interpreting it as samples of $|\mathcal{S}|$ independent and identically distributed pairs of random variables $(Y_m, Z_m) \in \mathcal{X} \times \mathcal{X}$, for $m = 1, ..., |\mathcal{S}|$. Associated with each pair (Y_m, Z_m), there is a set of $|\mathcal{S}|$ multinomial latent class variable $R_m \in \{1, ..., L\}$, also independent and identically distributed, conditioned on which the variables Y_m and Z_m themselves are mutually independent and identically distributed, that is

$$P(Y_m = y, Z_m = z | R_m = r) = P(Y_m = y | R_m = r)\, P(Z_m = z | R_m = r) \tag{5}$$

and

$$P(Z_m = z | R_m = r) = P(Y_m = z | R_m = r), \tag{6}$$

for any $r \in \{1, ..., L\}$, and $z \in \{1, ..., N\}$. The rationale supporting the adoption of this model for clustering is that if there is an underlying cluster structure revealed by the observations in $\mathcal{S}$, then this structure may be captured by the the different conditional probabilities. For example, if $L = 2$ and there are two clearly separated clusters, $\{1, ..., T\}$ and $\{T+1, ..., N\}$, then $P(Y_m = z | R_m = 1)$ will have values close to zero, for $z \in \{T+1, ..., N\}$, and relatively larger values for $z \in \{T+1, ..., N\}$, whereas $P(Y_m = z | R_m = 2)$ will have the reverse behavior.

The modeling assumptions in (5) and (6) correspond to a mixture model for (Y_m, Z_m) of the form

$$P(Y_m = y, Z_m = z) = \sum_{r=1}^{L} P(Y_m = y|R_m = r)\, P(Y_m = z|R_m = r)\, P(R_m = r), \tag{7}$$

which induces a natural mechanism for generating a random sample from (Y_m, Z_m): start by obtaining a sample r of the random variable R_m (with probability $P(R_m = r)$); then, obtain two independent samples y and z, with probabilities $P(Y_m = y|R_m = r)$ and $P(Y_m = z|R_m = r)$.

The model is parameterized by the (common) probability distribution of the latent variables R_m, $(P(R_m = 1), ..., P(R_m = L))$, and by the L conditional probability distributions $(P(Y_m = 1|R_m = r), ..., P(Y_m = N|R_m = r))$, for $r = 1, ..., L$. We write these distributions compactly as an L-vector $\mathbf{p} = (p_1, ..., p_L)$, where $p_r = P(R_m = r)$ (for any $m = 1, ..., |\mathcal{S}|$) and an $L \times N$ matrix $\mathbf{B} = [B_{r,j}]$, where $B_{r,j} = P(Y_m = j|R_m = r) = P(Z_m = j|R_m = r)$ (for any $m = 1, ..., |\mathcal{S}|$). With this notation, we can write

$$P(Y = y, Z = z, R = r) = p_r\, B_{r,y}\, B_{r,z}, \tag{8}$$

and

$$P(Y = y, Z = z) = \sum_{r=1}^{L} p_r\, B_{r,y}\, B_{r,z}. \tag{9}$$

With the generative model in hand, we can now write the probability distribution for the observed set of pairs $\mathcal{S} = \{(y_m, z_m),\ m = 1, ..., |\mathcal{S}|\}$, assumed to be independent and identically distributed samples of (Y, Z):

$$P(\mathcal{S}|\mathbf{p}, \mathbf{B}) = \prod_{m=1}^{|\mathcal{S}|} \sum_{r=1}^{L} p_r\ B_{r,y_m}\, B_{r,z_m}. \tag{10}$$

Consider now the so-called complete data, which, in addition to $\mathcal{S}$ (the samples (y_m, z_m) of (Y_m, Z_m), for $m = 1, ..., |\mathcal{S}|$), also contains the corresponding (missing/latent) samples of the random variables R_m, $\mathcal{R} = \{r_m,\ m = 1, ..., |\mathcal{S}|\}$. The so-called complete likelihood is then

$$P(\mathcal{S}, \mathcal{R}|\mathbf{p}, \mathbf{B}) = \prod_{m=1}^{|\mathcal{S}|} p_{r_m} B_{r_m,y_m} B_{r_m,z_m} \tag{11}$$

$$= \prod_{m=1}^{|\mathcal{S}|} \prod_{r=1}^{L} \left(p_r\, B_{r,y_m}\, B_{r,z_m}\right)^{\mathbb{I}(r_m=r)}. \tag{12}$$

Although it would be computationally very easy, it is not possible to obtain maximum likelihood estimates of $\mathbf{p}$ and $\mathbf{B}$ from (12), because $\mathcal{R}$ is not observed. Alternatively, we will resort to the EM algorithm, which will provide maximum marginal likelihood estimates of $\mathbf{p}$ and $\mathbf{B}$, by maximizing $P(\mathcal{S}|\mathbf{p}, \mathbf{B})$ with respect to these parameters.

3 The Expectation Maximization Algorithm

According to the generative model described in the previous section, each possible value of $R_m \in \{1, ..., L\}$ corresponds to one of the L clusters and each probability $B_{r,j} = P(Y_m = j | R_m = r)$ is the probability that cluster r "owns" object j, which can be seen as a soft assignment. Consequently, estimating matrix $\mathbf{B}$ will reveal the underlying (consensus) cluster structure. We pursue that goal by using the EM algorithm [4], where $\mathcal{R}$ is the missing data.

3.1 The E-step

The complete log-likelihood (the expectation of which is computed in the E-step) can be obtained from (12),

$$\log P(\mathcal{S}, \mathcal{R} | \mathbf{p}, \mathbf{B}) = \sum_{m=1}^{|\mathcal{S}|} \sum_{r=1}^{L} \mathbb{I}(r_m = r) \log\big(p_r \, B_{r,y_m} \, B_{r,z_m}\big). \tag{13}$$

The E-step consists in computing the conditional expectation of $\log P(\mathcal{S}, \mathcal{R} | \mathbf{p}, \mathbf{B})$ with respect to $\mathcal{R}$, conditioned on the current parameter estimates $\widehat{\mathbf{p}}$ and $\widehat{\mathbf{B}}$ and the observed $\mathcal{S}$, yielding the well-known Q-function. Since $\log P(\mathcal{S}, \mathcal{R} | \mathbf{p}, \mathbf{B})$ is a linear function of the (latent) binary indicator variables $\mathbb{I}(R_m = r)$,

$$Q(\mathbf{p}, \mathbf{B}; \widehat{\mathbf{p}}, \widehat{\mathbf{B}}) = \sum_{m=1}^{|\mathcal{S}|} \sum_{r=1}^{L} \mathbb{E}\left[\mathbb{I}(R_m = r) \,\middle|\, \mathcal{S}, \widehat{\mathbf{p}}, \widehat{\mathbf{B}}\right] \log\big(p_r \, B_{r,y_m} \, B_{r,z_m}\big) \tag{14}$$

$$= \sum_{m=1}^{|\mathcal{S}|} \sum_{r=1}^{L} \widehat{R}_{m,r} \log\big(p_r \, B_{r,y_m} \, B_{r,z_m}\big), \tag{15}$$

where

$$\widehat{R}_{m,r} \equiv \mathbb{E}\left[\mathbb{I}(R_m = r) \,\middle|\, \mathcal{S}, \widehat{\mathbf{p}}, \widehat{\mathbf{B}}\right] = P\left[R_m = r \,\middle|\, (y_m, z_m), \widehat{\mathbf{p}}, \widehat{\mathbf{B}}\right], \tag{16}$$

due to the independence assumption among the pairs and the fact that $\mathbb{I}(R_m = r)$ is a binary variable. The meaning of $\widehat{R}_{m,r}$ is clear: the conditional probability that the pair (y_m, z_m) was generated by cluster r. Finally, we can write

$$\widehat{R}_{m,r} = \frac{\widehat{p}_r \, \widehat{B}_{r,y_m} \, \widehat{B}_{r,z_m}}{\sum_{s=1}^{L} \widehat{p}_s \, \widehat{B}_{s,y_m} \, \widehat{B}_{s,z_m}}, \tag{17}$$

which is then plugged into (15).

3.2 The M-step

The M-step consists in maximizing, with respect to $\mathbf{p}$ and $\mathbf{B}$, the Q-function, which we now write as

$$Q(\mathbf{p},\mathbf{B};\widehat{\mathbf{p}},\widehat{\mathbf{B}}) = \sum_{m=1}^{|\mathcal{S}|}\sum_{r=1}^{L}\widehat{R}_{m,r}\log\big(p_r\big) + \sum_{m=1}^{|\mathcal{S}|}\sum_{r=1}^{L}\widehat{R}_{m,r}\log\big(B_{r,y_m}\,B_{r,z_m}\big), \quad (18)$$

showing that, as is common in EM for mixture models, the maximizations with respect to $\mathbf{p}$ and $\mathbf{B}$ can be carried out separately.

The maximization with respect to $\mathbf{p}$ (of course, under the constraints that $p_r \geq 0$, for $r = 1, ..., L$ and $\sum_{r=1}^{L} p_r = 1$) leads to the well-known

$$\widehat{p}_r^{\,\text{new}} = \frac{1}{|\mathcal{S}|}\sum_{m=1}^{|\mathcal{S}|}\widehat{R}_{m,r} \quad \text{for } r = 1, ..., L. \quad (19)$$

For the maximization with respect to $\mathbf{B}$, we begin by writing the relevant terms of (18) as

$$\sum_{m=1}^{|\mathcal{S}|}\sum_{r=1}^{L}\widehat{R}_{m,r}\log\big(B_{r,y_m}\,B_{r,z_m}\big) \;=\; \sum_{r=1}^{L}\sum_{y=1}^{N}\sum_{z=1}^{N}\widehat{C}_{y,z}^{\,r}\log\big(B_{r,y}\,B_{r,z}\big) \quad (20)$$

$$= \sum_{r=1}^{L}\sum_{y=1}^{N}\log\big(B_{r,y}\big)\sum_{z=1}^{N}\widehat{C}_{y,z}^{\,r} + \sum_{r=1}^{L}\sum_{z=1}^{N}\log\big(B_{r,z}\big)\sum_{y=1}^{N}\widehat{C}_{y,z}^{\,r} \quad (21)$$

$$= 2\sum_{r=1}^{L}\sum_{y=1}^{N}\log\big(B_{r,y}\big)\sum_{z=1}^{N}\widehat{C}_{y,z}^{\,r} \quad (22)$$

where

$$\widehat{C}_{y,z}^{\,r} = \sum_{m=1}^{|\mathcal{S}|}\widehat{R}_{m,r}\,\mathbb{I}((y_m,z_m)=(y,z)), \quad (23)$$

for $r = 1, ..., L$ and $y, z \in \{1, ..., N\}$ and the equality in (22) is due to the symmetry relation $\widehat{C}_{y,z}^{\,r} = \widehat{C}_{z,y}^{\,r}$. Comparison of (23) with (4) shows that $\widehat{C}_{y,z}^{\,r}$ is a weighted version of the co-association matrix; instead of simply counting how many times the pair (y, z) appeared in a common cluster in the clustering ensemble, each of these appearances is weighted by the probability that that particular co-occurrence was generated by cluster r. We thus have L weighted co-association matrices, $\widehat{\mathbf{C}}^1$, ..., $\widehat{\mathbf{C}}^L$, whose elements are given by (23).

Finally, maximization of (22) with respect to $B_{r,y}$, under the constraints $B_{r,y} \geq 0$, for all $r = 1, ..., L$ and $y = 1, ..., N$, and $\sum_{y=1}^{N} B_{r,y} = 1$, for all $r = 1, ..., L$, leads to

$$\widehat{B}_{r,y}^{\,\text{new}} = \frac{\sum_{z=1}^{N}\widehat{C}_{y,z}^{\,r}}{\sum_{t=1}^{N}\sum_{z=1}^{N}\widehat{C}_{t,z}^{\,r}}. \quad (24)$$

3.3 Summary of the Algorithm and Interpretation of the Estimates

In summary, the proposed EM algorithm, termed PEnCA (probabilistic ensemble clustering algorithm) works as follows:

1. Given the set of objects, obtain an ensemble $\mathbb{P}$ of clusterings and, from this ensemble, build the set $\mathcal{S}$ of co-occurring pairs (see Section 2.1).
2. Choose a number of clusters, L, and initialize the parameter estimates $\widehat{\mathbf{p}}$ and $\widehat{\mathbf{B}}$.
3. Perform the E-step, by computing $\widehat{R}_{m,r}$, for $m = 1, ..., |\mathcal{S}|$ and $r = 1, ..., L$ according to (17).
4. Compute the weighted co-association matrices $\widehat{\mathbf{C}}^1$, ..., $\widehat{\mathbf{C}}^L$, according to (23).
5. Update the parameter estimates according to (19) and (24).
6. If some stopping criterion is satisfied, stop; otherwise go back to step 3.

The parameter estimates returned by the algorithm have clear interpretations: $\widehat{p}_1, ..., \widehat{p}_L$ are the probabilities of the L clusters; each distribution $\widehat{B}_{r,1}, ..., \widehat{B}_{r,N}$ can be seen as the sequence of degrees of ownership of the N objects by cluster r. This is in contrast with the original EAC work [8, 9], where once a co-association matrix is obtained, a consensus clustering is sought by applying a some hard clustering algorithm. Notice that these soft ownerships are obtained even if all the clusterings in the ensemble are hard. It is also elementary to obtain an estimate of probability that object y belongs to cluster r (denoted as $\widehat{V}_{y,r}$), by applying Bayes law:

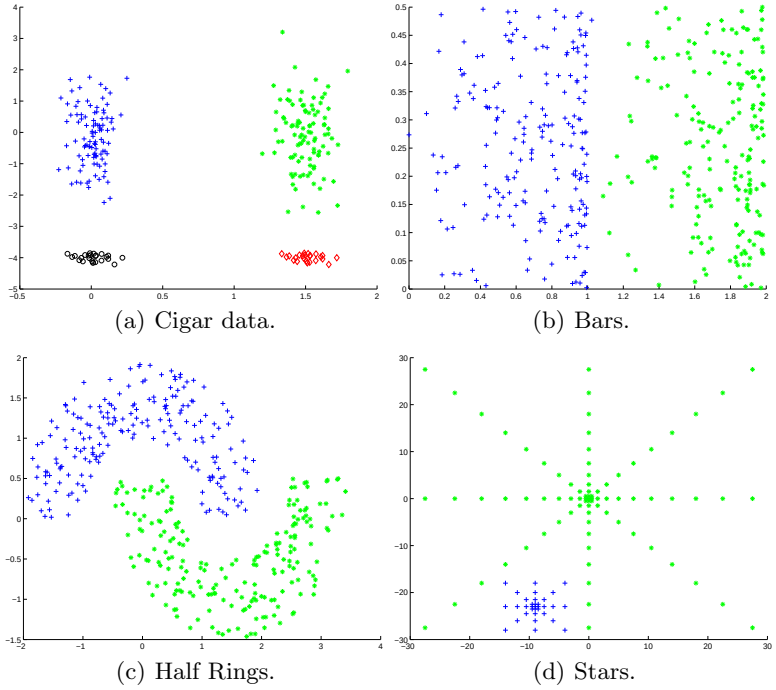
$$\widehat{V}_{y,r} = \hat{P}(R = r|Y = y) = \frac{\widehat{B}_{r,y}\,\widehat{p}_r}{\sum_{s=1}^{L} \widehat{B}_{s,y}\,\widehat{p}_s}. \tag{25}$$

4 Related Work

Topchy *et al.* introduced a combination method based on probabilistic model of the consensus partition, in the space of contributing clusters of the ensemble [18] [19]. As in present work, the consensus partition is found by solving a maximum likelihood estimation problem with respect to the parameters of a finite mixture distribution. Each mixture component is a multinomial distribution and corresponds to a cluster in the target consensus partition. As in this work, the maximum likelihood problem is solved using the EM algorithm. Our method differs from that of Topchy *et al.* in that it is based on co-association information.

Wang *et al.* extented the idea, with a model entitled *Bayesian cluster ensembles* (BCE) [20]. It is a mixed-membership model for learning cluster ensembles, assuming that they were generated by a graphical model. Although the posterior distribution cannot be calculated in closed, it is approximated using variational inference and Gibbs sampling. That work is very similar to the *latent Dirichlet allocation* (LDA) model [10], [16], but applied to a different input feature space.

Bulò *et al.* presented a method built upon the EAC framework where the co-association matrix was probabilisticly interpreted, and the extracted consensus solution consisted in a soft partition [3]. The method reduced the clustering problem to a polynomial optimization in the probability domain, solved using the Baum-Eagon inequality.



(a) Cigar data. (b) Bars. (c) Half Rings. (d) Stars.

Fig. 1. Synthetic two-dimensional datasets.

5 Experimental Results and Discussion

In this section, we present results for the evaluation of the proposed algorithm (which we refer to as PEnCA – *probabilistic ensemble clustering algorithm*) on several synthetic and real-world benchmark datasets from the well known UCI (University of California, Irvine) repository[2]. Figure 1 presents the four synthetic two-dimensional datasets used for this study.

To produce the clustering ensembles, we extend [14], where the classical K-means algorithm is used, and the several partitions in the ensembles are obtained

[2] `http://www.ics.uci.edu/~mlearn/MLRepository.html`

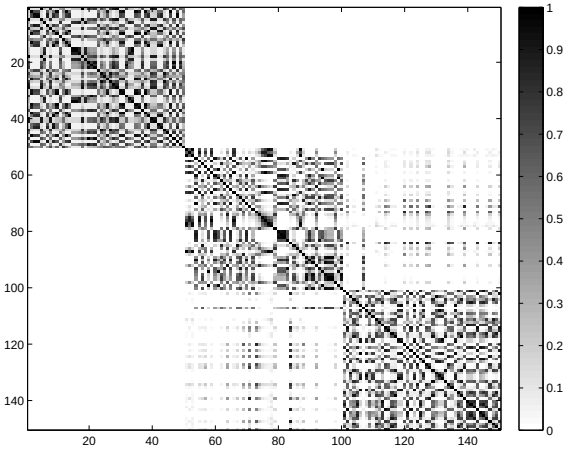


Fig. 2. Example of co-association matrix for the *Iris* dataset.

by varying the numbers of clusters and the initialization. The minimum and the maximum number of clusters varied as a function of the number of samples, according to the following rule:
$\{K_{min}, K_{max}\} = \left\{\max\left(\lceil\sqrt{N}/2\rceil, \lceil N/50\rceil\right), K_{min} + 20\right\}$,

Figure 2 shows a co-association matrix obtained for the *Iris* dataset, using an ensemble produced with the proposed rule.

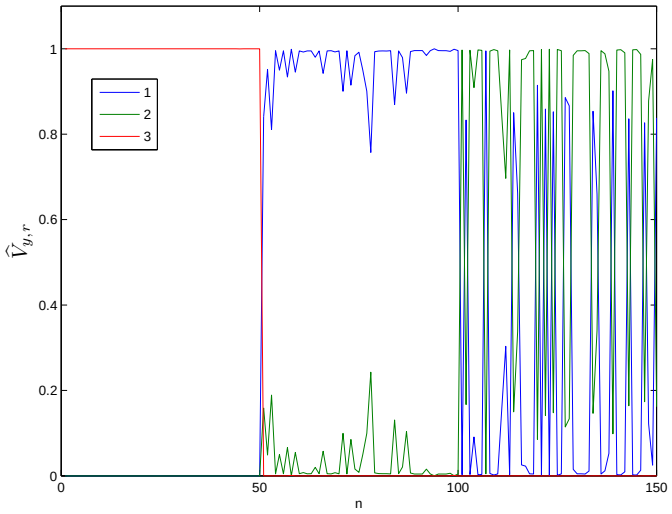


Fig. 3. Soft assignments obtained by PEnCA for the *Iris* dataset.

The color scheme of the representation ranges from white ($\mathcal{C}(y,z) = 0$) to black ($\mathcal{C}(y,z) = M$). Notice the evident block diagonal structure, and the clear separation the between the three clusters (Setosa, Versicolour, and Virginica).

Figure 3 presents the probabilistic assignment of each sample to each cluster, given by the posterior probabilities $\widehat{V}_{y,r}$ (see 25)) obtained after running the EM algorithm with $L = 3$. Notice that the assignments of the last fifty labels are noisier due to the not so clear separation of clusters 2 and 3, as can be seen in co-association matrix in Figure 2.

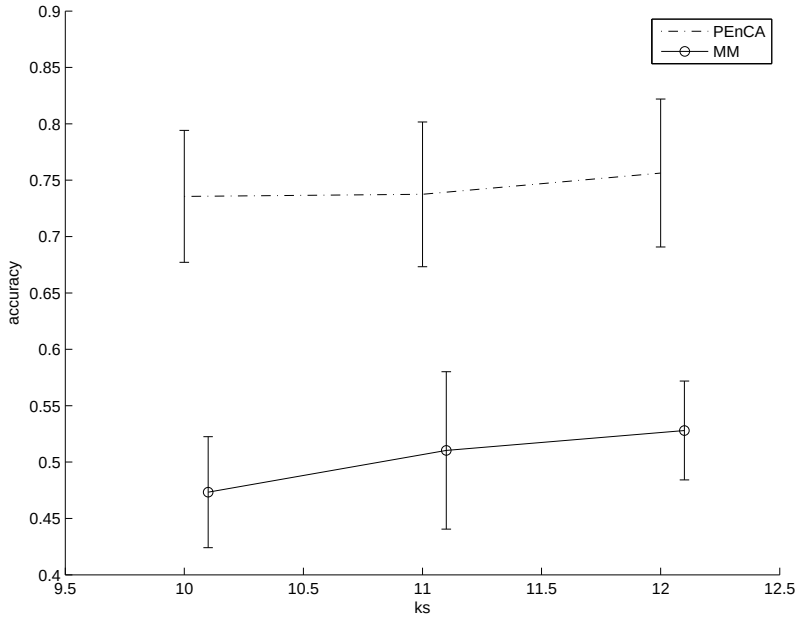


Fig. 4. Results on the *optdigits-r-tra-1000* dataset: accuracy (mean and standard deviation) over the several trials and for different number of aspects.

In a second set of experiments, we compared the results of PEnCA with those obtained with the approach from [18] (which we will refer to as MM – *mixture model*). The performance of the two methods was systematically assessed in terms of accuracy, by comparing the respective consensus partitions with ground truth clusterings. The accuracy is calculated using the *consistency index* (CI) [8] which provides percentages of correct labels. For each ensemble, we have repeated the extraction of the consensus partition 10 times, in order to test the variability of the result and the dependence of the initialization.

Figure 4 shows the results for the *optdigits-r-tra-1000* dataset, the variability in the accuracy over the several trials, and for different numbers of clusters. The dashed and solid lines represent, respectively, the PEnCA and the MM results;

blue and red represent results for ensemble (a) and (b). Notice that the variability in the accuracy on both models is of the same order of magnitude (in this example approximately 5% of the absolute value) and that PEnCA has always achieves higher accuracies than MM.

Finally, Table 1 reports the results obtained on several benchmark datasets (four synthetic and eight USI datasets).The best result for each dataset is shown in bold. These results show that PEnCA almost always achieves better accuracy than MM.

Table 1. Results obtained on the benchmark datasets (see text for details)

Data Set	N	K	PEnCA	MM
stars	114	2	**0.921**	0.737
cigar-data	250	4	0.712	**0.812**
bars	400	2	**0.985**	0.812
halfrings	400	2	**1.000**	0.797
iris-r	150	3	**0.920**	0.693
wine-normalized	178	3	**0.949**	0.590
house-votes-84-normalized	232	2	**0.905**	0.784
ionosphere	351	2	0.724	**0.829**
std-yeast-cellcycle	384	5	**0.729**	0.578
pima-normalized	768	2	**0.681**	0.615
Breast-cancers	683	2	**0.947**	0.764
optdigits-r-tra-1000	1000	10	**0.876**	0.581

6 Conclusions and Future Work

In this paper, we have proposed a probabilistic generative model for consensus clustering, based on a dyadic aspect model for evidence accumulation clustering framework.

Given an ensemble of clusterings, the consensus partition is extracted by solving a maximum likelihood estimation problem via the expectation-maximization (EM).

The output of the method is a probabilistic assignment of each sample to each cluster, which is an advantage over previous works using the evidence accumulation framework.

Experimental assessment of the performance of the proposed method has shown that it outperforms another recent probabilistic approach to ensemble clustering.

One of the advantages of this framework is the possibility of inclusion of a model selection criterion. We hope to address this issue in future.

On going work on different initialization schemes and strategies to escape from local solutions is being carried on.

7 Acknowledgements

This work was partially supported by Fundação para a Ciência e Tecnologia (FCT) under the grants SFRH/PROTEC/49512/2009 and PTDC/EIACCO/103230/2008 (Project EvaClue), and by the Future and Emerging Technologies Open Scheme (FET-Open) of the Seventh Framework Programme of the European Commission, under the SIMBAD project (contract 213250).

References

1. H. G. Ayad and M. S. Kamel. Cumulative voting consensus method for partitions with variable number of clusters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(1):160–173, 2008.
2. J. Buhmann. Information theoretic model validation for clustering. In *IEEE International Symposium on Information Theory*, 2010.
3. S. R. Bulò, A. Lourenço, A. Fred, and M. Pelillo. Pairwise probabilistic clustering using evidence accumulation. In *Proceedings of the 2010 joint IAPR international conference on Structural, syntactic, and statistical pattern recognition*, SSPR&;SPR'10, pages 395–404, Berlin, Heidelberg, 2010. Springer-Verlag.
4. A. Demspter, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society (B)*, 39:1–38, 1977.
5. X. Z. Fern and C. E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proc ICML '04*, 2004.
6. M. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):381–396, 2002.
7. B. Fischer, V. Roth, and J. Buhmann. Clustering with the connectivity kernel. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Neural Information Processing Systems – NIPS 16*. 2004.
8. A. Fred. Finding consistent clusters in data partitions. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems*, volume 2096, pages 309–318. Springer, 2001.
9. A. Fred and A. Jain. Combining multiple clustering using evidence accumulation. *IEEE Trans Pattern Analysis and Machine Intelligence*, 27(6):835–850, June 2005.
10. T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proc Natl Acad Sci U S A*, 101 Suppl 1:5228–5235, April 2004.
11. T. Hofmann. Unsupervised learning from dyadic data. pages 466–472. MIT Press, 1998.
12. T. Hofmann and J. Puzicha. Statistical models for co-occurrence data. Technical report, Cambridge, MA, USA, 1998.
13. T. Hofmann, J. Puzicha, and M. I. Jordan. Learning from dyadic data. In *Advances in Neural Information Processing Systems (NIPS) 11*, Cambridge MA, 1999. MIT Press.
14. A. Lourenço, A. Fred, and A. K. Jain. On the scalability of evidence accumulation clustering. In *ICPR*, Istanbul Turkey, August 23-26 2010.
15. J. Rissanen. *Stochastic COmplexity in Statistical Inquiry*. World Scientific, 1989.
16. M. Steyvers and T. Griffiths. *Probabilistic topic models*, chapter Latent Semantic Analysis: A Road to Meaning. Laurence Erlbaum, 2007.
17. A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *J. of Machine Learning Research 3*, 2002.

18. A. Topchy, A. Jain, and W. Punch. A mixture model of clustering ensembles. In *Proc. of the SIAM Conf. on Data Mining*, April 2004.
19. A. Topchy, A. K. Jain, and W. Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1866–1881, 2005.
20. H. Wang, H. Shan, and A. Banerjee. Bayesian cluster ensembles. In *9th SIAM International Conference on Data Mining*, 2009.

[4] - (SIMBAD Technical Report n. 2011_10)

Aidos, H., Fred, A.L.N.: A study of embedding methods under the evidence accumulation framework. In: SIMBAD workshop. Lecture Notes in Computer Science. Springer (2011) To appear.

A Study of Embedding Methods under the Evidence Accumulation Framework

Helena Aidos and Ana Fred

Instituto de Telecomunicações, Instituto Superior Técnico, Lisbon, Portugal
{haidos, afred}@lx.it.pt

Abstract. In this paper we address a voting mechanism to combine clustering ensembles leading to the so-called co-association matrix, under the Evidence Accumulation Clustering framework. Different clustering techniques can be applied to this matrix to obtain the combined data partition, and different clustering strategies may yield too different combination results. We propose to apply embedding methods over this matrix, in an attempt to reduce the sensitivity of the final partition to the clustering method, and still obtain competitive and consistent results. We present a study of several embedding methods over this matrix, interpreting it in two ways: (i) as a feature space and (ii) as a similarity space. In the first case we reduce the dimensionality of the feature space; in the second case we obtain a representation constrained to the similarity matrix. When applying several clustering techniques over these new representations, we evaluate the impact of these transformations in terms of performance and coherence of the obtained data partition. Experimental results, on synthetic and real benchmark datasets, show that extracting the relevant features through dimensionality reduction yields more consistent results than applying the clustering algorithms directly to the co-association matrix.

Keywords: clustering ensembles, co-association matrix, evidence accumulation clustering, embedding methods.

1 Introduction

Clustering is one of the central problems in Pattern Recognition and Machine Learning. Given a set of unlabeled data, its typical goal is to group objects into clusters, such that objects within a cluster are similar, and objects in distinct clusters are dissimilar. Assuming that clusters are disjoint, the clustering process leads to a data partition. Hundreds of clustering algorithms exist, handling differently issues such as cluster shape, density, noise. k-means is one of the most studied and used algorithms [9, 18].

Recently, taking advantage of the diversity of clustering solutions produced by clustering algorithms over the same dataset, an approach known as *Clustering Ensemble methods*, has been proposed and gained an increasing interest [4, 16, 10, 1]. Given a set of data partitions - a clustering ensemble (CE) - these methods propose a consensus partition based on a combination strategy, having in general a leveraging effect over the single data partitions in the CE.

We can generate clustering ensembles following two approaches: choice of data representation or choice of clustering algorithms or algorithmic parameters. In the first

case, we can get different representations of objects by applying different preprocessing mechanisms or feature extraction techniques, or just by sampling the data a number of times. We can also have clustering ensembles if we use several clustering algorithms or just the same algorithm with different parameter values.

Fred and Jain [5] proposed a clustering ensemble approach based on the combination of information provided by a set of different partitions of a given dataset, through the Evidence Accumulation method. To combine all the different partitions, Fred and Jain [5] proposed a voting scheme, which leads to a pairwise relationships matrix, called "co-association matrix". The final data partition is obtained by applying a clustering algorithm over the co-association matrix. One main advantage of this voting scheme is that it can deal with partitions having different number of clusters and different data representations.

The application of different clustering techniques to this matrix may yield different solutions. We propose to use embedding methods (also called dimensionality reduction (DR) methods) over this matrix, in an attempt to reduce the sensitivity of the combined data partition to the clustering method, and obtain better and more consensual results. We present a study of the performance and coherence of the solutions when different clustering techniques are applied to the resulting data representations. To obtain those representations we will follow two approaches: interpret the co-association matrix as a feature space, and as a similarity space.

The first approach is similar to the one proposed by Kuncheva *et al*. [11]: we will view the co-association matrix as a feature space, but instead of using the full feature space, we will reduce its dimension using several dimensionality reduction (DR) methods. These DR techniques aim to take a set of data points in a high-dimensional space and output a new set of data points in a lower-dimensional space, in a way that preserves the topology of the high-dimensional data. This new data representation is commonly called an *embedding* of the original dataset. We will empirically show that the use of DR methods to remove redundant features improves the quality and consistency of the final partition for different clustering techniques.

In the other approach we view the co-association matrix as a similarity space, as in [5]. However, instead of applying directly the clustering techniques to this matrix, we will first apply DR methods to it. Many DR methods take as input some distance measure between points (usually in a distance matrix whose (i, j) entry contains the distance between data points i and j). Therefore, if one converts the similarity measures in the co-association matrix into distance (or dissimilarity) measures, one can input this dissimilarity matrix into the DR methods directly. The resulting low-dimensional data points are then clustered with several clustering techniques. Again, we intend to study if there exists consistency and an improvement in the quality of the solutions.

The dimensionality reduction methods used have different characteristics such as: linear vs. nonlinear; preserving local structure vs. preserving global structure; preserve spatial distances vs. preserving graph distances. This means that different embedding strategies may influence differently the solutions; we intend to study if there exists a class of embedding methods suitable for certain types of datasets (well separate clusters, touching clusters).

This paper is organized as follows: Section 2 gives a brief explanation of the embedding algorithms used in the study. Section 3 explains the evidence accumulation approach, including the construction of the co-association matrix. Section 4 explains the new methodology proposed in this paper and the two interpretations we give to this matrix. Section 5 describes the datasets used in this study and the experimental results for the two interpretations of the co-association matrix: feature space (section 5.2) and similarity space (section 5.3). We summarize and discuss the main findings in Section 6. Conclusions are drawn in Section 7.

2 Embedding Methods

To perform embeddings we will use several unsupervised DR methods: Locality Preserving Projections (LPP) [7]. Neighborhood Preserving Projections (NPE) [6], Sammon's mapping [15], Curvilinear Component Analysis (CCA) [3], Isomap [17], Curvilinear Distance Analysis (CDA) [13], Locally Linear Embedding (LLE) [14] and Laplacian Eigenmap (LE) [2]. We now briefly introduce each of these algorithms.

2.1 Nonlinear Methods

The *Locally Linear Embedding* (LLE) [14] assumes that the data manifold is smooth and sampled densely enough such that each data point lies close to a locally linear subspace on the manifold. In other words, the manifold smoothness and sampling should be enough to locally approximate the manifold by a hyperplane. LLE makes a locally linear approximation of the whole data manifold; it first estimates a local coordinate system for each data point from its k-nearest neighbors. To produce the embedding, LLE finds low-dimensional coordinates that preserve the previously estimated local coordinate systems as well as possible. Technically, LLE first minimizes the reconstruction error $E(\mathbf{W}) = \sum_i \|\mathbf{x}_i - \sum_j W_{i,j}\mathbf{x}_j\|^2$ with respect to the coefficients $W_{i,j}$, under the constraints that $W_{i,j} = 0$ if i and j are not neighbors, and $\sum_j W_{i,j} = 1$. After finding these weights, the low-dimensional configuration of points is next found by minimizing $E(\mathbf{Y}) = \sum_i \|\mathbf{y}_i - \sum_j W_{i,j}\mathbf{y}_j\|^2$ with respect to the low-dimensional representation $\mathbf{y}_i$ of each data point.

The *Laplacian Eigenmap* (LE) [2] uses a graph embedding approach. An undirected k-nearest neighbor graph is formed, where each data point is a vertex. Points i and j are connected by an edge with weight $W_{i,j} = 1$ if j is among the k nearest neighbors of i, otherwise the edge weight is set to zero; this simple weighting method has been found to work well in practice [2]. To find a low-dimensional embedding of the graph, the algorithm tries to put points that are connected in the graph as close to each other as possible and does not care what happens to the other points. Technically, it minimizes $\frac{1}{2}\sum_{i,j} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{i,j} = \mathbf{y}^T\mathbf{L}\mathbf{y}$ with respect to the low-dimensional point locations $\mathbf{y}_i$, where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian and $\mathbf{D}$ is a diagonal matrix with elements $\mathrm{D}_{ii} = \sum_j W_{i,j}$. This cost function has an undesirable trivial solution: having all points in the same position would have a cost of zero, which would be a global minimum of the cost function. In practice, the low-dimensional configuration is found by solving

the generalized eigenvalue problem $\mathbf{Ly} = \lambda\mathbf{Dy}$ [2]. The smallest eigenvalue corresponds to the trivial solution, but the eigenvectors corresponding to the next smallest eigenvalues yield the desired LE solution.

Isomap [17] is a variant of Multidimensional Scaling (MDS) [12], which finds a configuration of output coordinates matching a given distance matrix. Isomap does not compute pairwise input-space distances as simple Euclidean distances but as *geodesic distances* along the manifold of the data (technically, along a graph formed by connecting all k-nearest neighbors). Given these geodesic distances the output coordinates are found by standard linear MDS. When output coordinates are found for such input distances, the manifold structure in the original data becomes unfolded; it has been shown that this algorithm is asymptotically able to recover certain types of manifolds.

Curvilinear component analysis (CCA) [3] is a variant of MDS [12] that tries to preserve only distances between points that are near each other in the embedding. This is achieved by weighting each term in the MDS cost function by a coefficient that depends on the corresponding pairwise distance in the embedding. In our case, this coefficient is simply 1 if the distance is below a predetermined threshold and 0 if it is larger. This approach is similar to Isomap but the determination of whether two points are neighbors is done in the output space in CCA, rather than in the input space as in Isomap.

Curvilinear distance analysis Curvilinear Distance Analysis (CDA) [13] is an extension of CCA. The idea is to replace in MDS the Euclidean distances in the original space with geodesic distances in the same manner as in the Isomap algorithm. Otherwise the algorithm is similar to CCA.

2.2 Linear Methods

Locality Preserving Projections (LPP) [7] is a linear dimensionality reduction method that preserves local neighborhood information. It shares many properties of nonlinear techniques such as Laplacian Eigenmaps or Locally Linear Embedding, since it is a linear approximation of the nonlinear Laplacian Eigenmaps.

Neighborhood Preserving Projections (NPE) [6] is a linear dimensionality reduction method that preserves the local structure of the data. It has similar properties to LPP, but it is a linear approximation of Locally Linear Embedding (LLE), which means that it has properties similar to that method.

3 Evidence Accumulation: The Co-association Matrix

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of n objects or samples represented in a feature space or some other data representation. A clustering algorithm takes X as input and groups the n patterns into k clusters, forming a partition P. A *clustering ensemble*, $\mathbb{P}$, is a set of N different partitions of the data X:

$$\mathbb{P} = \{P^1, P^2, \ldots, P^N\} \tag{1}$$

$$
\begin{aligned}
P^1 &= \left\{C_1^1, C_2^1, \ldots, C_{k_1}^1\right\} \\
&\vdots \\
P^N &= \left\{C_1^N, C_2^N, \ldots, C_{k_N}^N\right\},
\end{aligned}
$$

where C_j^i is the jth cluster in data partition P^i, which has k_i clusters and n_j^i is the cardinality of C_j^i, with $\sum_{j=1}^{k_i} n_j^i = n, i = 1, \ldots, N$.

The *evidence accumulation* approach, proposed by Fred and Jain [5], is a three-step cluster ensemble method: 1- build the clustering ensemble (CE); 2- combine evidence in the CE, mapping it into a co-association matrix; 3- extract the consensus partition by applying a clustering algorithm over the co-association matrix. The basic idea is that patterns belonging to a "natural" cluster are very likely to be assigned to the same cluster in different data partitions. Taking the co-occurrences of pairs of patterns in the same cluster as votes for their association, the N data partitions of n patterns yield a $n \times n$ co-association matrix:

$$
\mathcal{C}(i,j) = \frac{n_{ij}}{N}, \tag{2}
$$

where n_{ij} is the number of times the pattern pair (i, j) is assigned to the same cluster among the N partitions.

In its normalized form, as per expression (2), matrix $\mathcal{C}$ can be given different interpretations, either probabilistic or simply as pairwise similarity. Another issue is how to address and use this matrix for clustering purposes. In the following we propose a novel methodology by applying DR techniques.

4 Dimensionality Reduction in Evidence Accumulation Clustering

We propose a new methodology called Dimensionality Reduction in Evidence Accumulation Clustering (DR-EAC), which is based on the Evidence Accumulation Clustering (EAC) method described above. As said before, the evidence accumulation approach is a three-step cluster ensemble method; we now propose a four-step method. We build the clustering ensemble (step 1) and the co-association matrix (step 2) similarly to the evidence accumulation approach. However, instead of applying a clustering algorithm directly to the co-association matrix, we apply a DR technique to it (which is now step 3). As detailed below, we propose two ways to do this, depending on how one interprets the co-association matrix. This DR technique outputs a low-dimensional dataset, which is then fed into a clustering algorithm (which is now step 4). We now discuss each of these four steps in more detail.

1) Build the Clustering Ensemble. As referred before, there are several ways to produce a clustering ensemble. In this study we build a clustering ensemble by running the k-means algorithm to produce a total of $N = 200$ data partitions, each one with k clusters, k being an integer randomly drawn between $k_{min} = \max\{\sqrt{n}/2, n/50\}$ and $k_{max} = k_{min} + 20$, where n is the number of samples of the dataset.

2) Obtain the co-association matrix. We begin by computing the co-association matrix according to equation (2). Then, we interpret this matrix in one of two possible ways:

- *Co-associations viewed as Features:* One way to look at matrix $\mathcal{C}$ is to say that its i-th row represents a new set of features for the i-th data point, an idea originally proposed by Kuncheva *et al.* [11]. Thus, each pattern is now represented by how many times it was grouped together with all other patterns.
- *Co-association viewed as Similarities:* We can transform the co-association matrix $\mathcal{C}$, which is a similarity matrix, into a dissimilarity matrix (or distance matrix). Since many DR methods can take as input a matrix of pairwise distances (or dissimilarities), if we transform this matrix of similarities into a matrix of dissimilarities we can exploit this property. Since the elements of $\mathcal{C}$ lie between 0 and 1, we use a very simple transformation: the new dissimilarity matrix has the element (i, j) given by $1 - \mathcal{C}(i, j)$.

3) Apply Dimensionality Reduction techniques. We apply DR techniques to obtain a new representation of the data, preserving the topology of the original data. For the DR methods we need to choose a target dimension to reduce the data to and, in some cases, we also have to choose a parameter of the method (usually the number of nearest neighbors to consider). In all cases we let each algorithm choose the most suitable parameter and dimension by an intrinsic criterion. This intrinsic criterion can be the value of the cost function that each algorithm has to minimize, or the reconstruction error. For example, in Isomap we chose the parameter (which is the number of nearest neighbors used to construct a graph) which minimizes the residual variance [17]. It is beyond the scope of this paper to detail how these parameters should be chosen; the relevant information can be found in the references cited in Section 2.

4) Extract the consensus partition. After we get the embedded data, we apply eight well-known clustering algorithms: k-means, single-link, complete-link, average-link, Ward-link, centroid-link, median-link and weighted-link [9].

4.1 Quality measures

We use two quality measures to assess the results: consistency index (CI) and normalized mutual information (NMI).

The CI simply measures the fraction of patterns correctly grouped together compared to the ground-truth labeling. It takes values between 0 and 1, and it is a measure of the accuracy of the clustering.

The NMI [16] is a symmetric measure of the information shared between two partitions. Consider the partition P^a, which describes a labeling of the n patterns in the dataset X into k_a clusters. If one takes frequency counts as approximations for probabilities, the entropy of the data partition P^a is given by $H(P^a) = -\sum_{i=1}^{k_a} \frac{n_i^a}{n} \log\left(\frac{n_i^a}{n}\right)$, where n_i^a represents the number of patterns in cluster $C_i^a \in P^a$. The agreement between two partitions P^a and P^b is given by their mutual information:

$$I(P^a, P^b) = \sum_{i=1}^{k_a} \sum_{j=1}^{k_b} \frac{n_{ij}^{ab}}{n} \log\left(\frac{\frac{n_{ij}^{ab}}{n}}{\frac{n_i^a}{n} \cdot \frac{n_j^b}{n}}\right),$$

whith n_{ij}^{ab} the number of shared patterns between clusters $C_i^a \in P^a$ and $C_j^b \in P^b$.

The NMI is then defined by

$$NMI(P^a, P^b) = \frac{I(P^a, P^b)}{\sqrt{H(P^a)H(P^b)}}.$$

It is similar to the widely used mutual information, but normalized to be in the interval $[0, 1]$. For each DR method, we compute the NMI between all 28 pairs of clustering algorithms[1]. We then take the average of these 28 NMI values to obtain the average NMI for that DR method. This average NMI will measure how consistent the partitions are among the 8 clustering algorithms after applying that DR method.

5 Experimental Results

We will apply the new methodology described in section 4 to several datasets, in an attempt to improve the quality and robustness of the solutions, compared to the evidence accumulation approach. We will apply the clustering algorithms mentioned in section 4 to the co-association matrix directly (in both interpretations), an approach we will denote by EAC$_F$ (Evidence Accumulation Clustering in the feature space) and EAC (Evidence Accumulation Clustering in the sense presented by [5]). The idea is to verify empirically whether the use of embedding methods and subsequent clustering algorithms is advantageous relative to the application of clustering algorithms on the co-association matrix directly. Also, we will try to find some correspondence between pairs of embedding and clustering methods suitable for some types of data. In that sense, we will study synthetic data and real data, with the synthetic data divided in two broad meta-sets: datasets with separate clusters and datasets with touching clusters.

5.1 Data

We used 18 datasets: 10 synthetic datasets (5 well-separated and 5 with touching clusters), and 8 real datasets from the UCI Machine Learning Repository[2]. The synthetic datasets were chosen to take into account a wide variety of situations: well-separated and touching clusters; gaussian and non-gaussian clusters; arbitrary shapes; and diverse cluster densities. These synthetic datasets are shown in figure 1. The *Iris* dataset consists of three species of Iris plants (Setosa, Versicolor and Virginica). This dataset is characterized by four features and 50 samples in each cluster. *Std Yeast* is composed of 384 samples (genes) over two cell cycles of yeast cell data. This dataset is characterized by 17 features and consisting of five clusters corresponding to the five phases of the cell cycle. The *Pima* dataset is composed of 768 samples (genes) from National Institute of Diabetes and Digestive and Kidney Diseases, it has 8 features and two clusters. *Wine* consists of the results of a chemical analysis of wines grown in the same region in Italy

[1] 28 is the number of off-diagonal elements in the upper triangular part of the matrix containing the NMI between pairs of clustering algorithms, which is an 8-by-8 matrix.

[2] http://archive.ics.uci.edu/ml

divided into three clusters with 59, 71 and 48 patterns described by 13 features. *Optdigits* is a subset of Handwritten Digits dataset containing only the first 100 patterns of each digit, from a total of 1000 data samples characterized by 64 attributes. The *Wisconsin Breast-Cancer* dataset consists of 683 patterns represented by nine features and has two clusters. The *House Votes* dataset consists of votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the Congressional Quarterly Almanac. It is composed by two clusters and only the patterns without missing values were considered, for a total of 232 samples (125 democrats and 107 republicans). The *Crabs* dataset consists of 200 patterns represented by 5 features and has two classes. Pima, House Votes, Crabs and Wine were normalized to have unit variance.

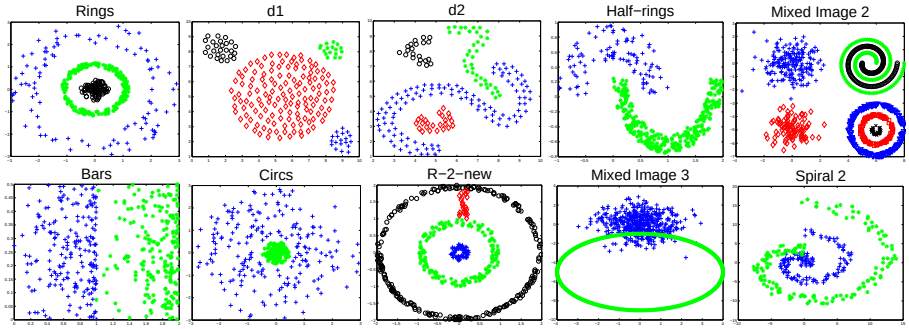


Fig. 1. Synthetic datasets.

5.2 Experiment 1: Feature Space

In this section we interpret the co-association matrix as a new feature space, as described in Section 4. The application of clustering algorithms directly to the co-association matrix viewed as a feature space, is here denoted by EAC_F.

Analyzing the average NMI in figure 2 over all clustering algorithms used to obtain the final partition, we notice that LE and LPP are the ones that produce more coherent solutions for the synthetic datasets with separate clusters (figure 2 top), which indicates that they are robust to the extraction algorithm. CCA and CDA are the algorithms with the most dispersion in the solutions for all datasets. Unlike for separate clusters, the NMI for datasets with touching clusters (figure 2 middle) shows that no DR algorithm is robust to the choice of the clustering algorithm. In the real datasets, LE is the most consistent DR algorithm in half of the datasets (Wine, Std Yeast, Optdigits and Iris).

Even if the NMI is high, it is not necessarily true that we have a high CI (i.e. that the results of the clustering algorithms are good), it only means that the clustering algorithms obtained similar final partitions. However, the use of that measure is a good indicator that the embedded space yields good clustering results regardless of the clustering algorithm. This is an advantage, since we do not know *a priori* which is the most suitable clustering algorithm for a certain kind of data.

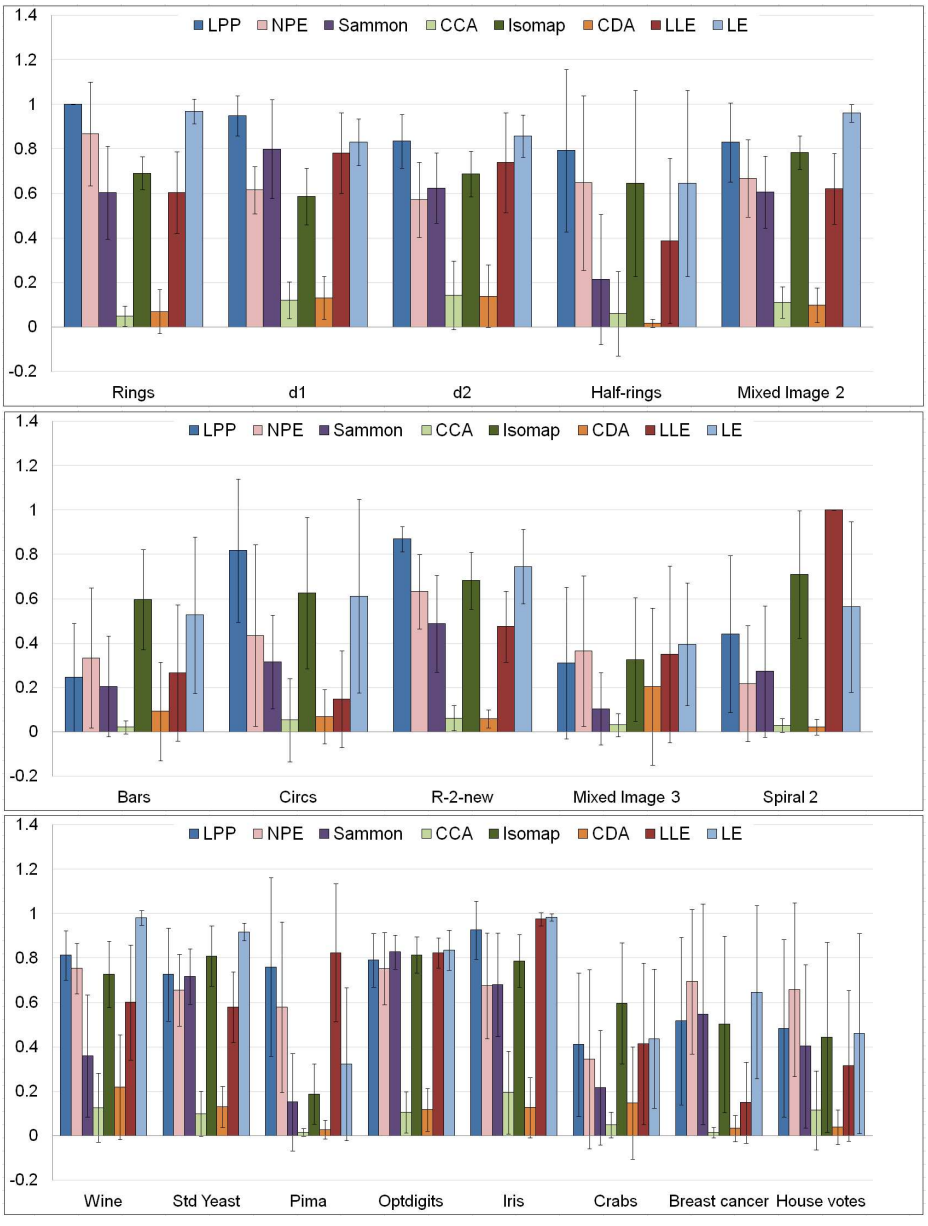


Fig. 2. Mean and standard deviation of Normalized Mutual Information over the clustering algorithms for each dataset and each embedding method. The co-association matrix was interpreted as features. *Top:* Synthetic datasets with separate clusters. *Middle:* Synthetic datasets with touching clusters. *Bottom:* Real datasets.

Table 1 contains the best CI values (first row of each dataset) and the corresponding clustering algorithm used for that solution; it also presents the average CI over all the clustering algorithms (second row of each dataset). Based on figure 2 we have claimed that LE and LPP are the ones that produce the most coherent solutions for the synthetic datasets with separate clusters; Table 1 corroborates these findings, since LE and LPP usually yield maximum CI for several clustering algorithms.

In synthetic datasets with separate clusters, LE and LPP, which are local algorithms, combine well with multiple hierarchical clustering algorithms. Isomap and Sammon, which are global and nonlinear, combine well with single-link, which is also the best clustering algorithm for EAC_F.

The analysis of the CI values for the synthetic datasets with touching clusters, shown in Table 1, shows that LPP, Isomap and LE are, on average values, better than EAC_F. In terms of maximum values, EAC_F outperforms the DR-based methods only in one dataset (R-2-new), and still by a very small margin; while it is outperformed in all remaining datasets.

The best DR-clustering algorithm pairs, for synthetic datasets with touching clusters, are LPP with k-means, Sammon with Ward-link and CDA with k-means. The overall best DR is Isomap, which is in first place in maximum CI for 4 out of 5 datasets.

The analysis of CI values for real datasets (see Table 1), shows that all DR methods do relatively well when compared to EAC_F, except for CCA and CDA. Isomap and Sammon are the two best DR algorithms when compared to the remaining DR techniques, especially in the Optdigits dataset. CCA and CDA are the worst overall methods, especially in the Std Yeast and Optdigits datasets.

These results show the advantage of performing DR over using EAC_F. In fact, from Table 1, using DR gives in general the best CI in all datasets, both in terms of maximum CI and of average CI.

Overall, for both synthetic and real datasets, there is no DR algorithm which is always robust in terms of NMI. However, LE and LPP (which is a linear version of LE), seem to have this property, especially in synthetic datasets with separate clusters. For the real datasets, LPP and LE present the best results, except in the Optdigits dataset, which yields better results with a global DR method (like Isomap and Sammon), instead of a local method.

5.3 Experiment 2: Similarity Space

In this section we interpret the entries of the co-association matrix as similarity values. We transform these into dissimilarity values, as described in Section 4. We plug-in this dissimilarity matrix into the embedding methods and will add "EA-" (from "Evidence Accumulation") before the acronyms of the DR methods to emphasize the dependency of this matrix.

The analysis of NMI values for the synthetic datasets with separate clusters, shown in Figure 3, shows that EA-LE and EA-LLE yield the most coherent clustering results, except for the Half-rings dataset. For the Mixed Image 2 dataset, local algorithms (EA-LPP, EA-NPE, EA-LLE and EA-LE) and global algorithms that preserve "geodesic" distances (EA-Isomap, EA-CDA) have very coherent results. However, the analysis of the CI values (Table 2) immediately shows that results are not good for that dataset.

Table 1. Consistency index (%) for co-association matrix interpreted as features. *(First row)* Best CI and clustering algorithm(s) which yield that CI value. Legend: (1) k-means, (2) single-link, (3) complete-link, (4) average-link, (5) Ward-link, (6) centroid-link, (7) median-link, (8) weighted-link. *(Second row)* Average CI (%) over all clustering methods. The gray cells correspond to the best NMI presented in figure 2 and the best average CI are shown in bold.

		EAC_F	LPP	NPE	Sammon	CCA	Isomap	CDA	LLE	LE
Synthetic data with separate clusters	Rings	100	100	61.25	100	50.00	100	52.00	85.50	100
		(2)	(1-8)	(1)	(2)	(2)	(2)	(4)	(5)	(2-8)
		65.28	**100**	55.13	64.78	43.00	73.56	45.75	66.06	99.47
	d1	100	100	82.00	100	70.00	100	70.50	72.50	100
		(2-8)	(2-8)	(6)	(2,4-8)	(2,6)	(2)	(2)	(2)	(2,8)
		98.44	**98.19**	65.25	91.75	54.75	65.31	50.13	69.69	85.88
	d2	100	93.50	51.00	100	59.00	69.00	60.50	50.50	67.00
		(2)	(7)	(7)	(2)	(6)	(2)	(4)	(2)	(3,4,6)
		76.31	**73.87**	42.56	73.00	49.87	59.56	49.25	44.88	61.69
	Half-rings	100	100	69.75	100	81.75	100	74.75	100	100
		(2)	(1,2,4-8)	(2,4-8)	(2)	(5)	(1,2)	(2,6,7)	(2)	(2,4-8)
		72.19	**93.31**	65.81	72.09	66.19	59.87	63.56	88.28	86.63
	Mixed Image 2	65.70	71.80	36.60	71.60	22.90	71.40	23.70	47.00	71.60
		(2)	(2)	(5)	(2)	(6)	(2)	(1)	(5)	(3)
		54.44	63.69	32.50	51.90	21.10	57.45	22.61	38.52	**70.66**
Synthetic data with touching clusters	Bars	99.25	99.25	79.25	99.25	59.50	99.25	73.75	76.00	96.00
		(5)	(4,5)	(1)	(5)	(1)	(2,3)	(1)	(7)	(1)
		68.19	75.25	64.78	68.19	53.09	**90.16**	58.78	62.84	76.84
	Circs	99.50	100	58.75	99.50	63.00	100	84.50	59.00	99.50
		(2,5,8)	(1-6,8)	(1)	(2,8)	(5)	(1,8)	(1)	(8)	(5)
		80.00	**96.16**	54.94	80.56	55.62	91.37	62.31	55.31	66.91
	R-2-new	90.20	77.40	44.40	89.20	50.40	82.80	51.20	57.20	78.60
		(4)	(1)	(2)	(4,5)	(6)	(4)	(7)	(5)	(1)
		66.60	**73.95**	40.55	70.52	39.52	71.22	42.57	51.67	71.52
	Mixed Image 3	84.90	71.90	66.80	74.60	54.80	89.50	74.80	55.30	83.80
		(5)	(1)	(3)	(5)	(8)	(3)	(1)	(3,5)	(4)
		61.52	67.10	58.16	61.59	52.15	73.42	55.59	53.17	**74.92**
	Spiral 2	77.67	77.67	64.33	77.67	58.67	85.00	51.67	85.00	85.00
		(2)	(2)	(8)	(2)	(1)	(2)	(1)	(1-8)	(2,5,7,8)
		63.50	70.96	56.54	61.12	52.50	82.54	50.75	**85.00**	81.33
Real data	Wine	96.07	98.31	90.45	96.07	72.47	96.63	84.27	61.24	96.63
		(8)	(3)	(3)	(5)	(1)	(5,6)	(1)	(5)	(1)
		75.91	94.03	77.18	71.07	46.49	88.48	58.43	47.68	**94.66**
	Std Yeast	60.94	63.80	58.07	61.20	37.24	61.20	35.94	60.16	71.35
		(4)	(1)	(7)	(8)	(4)	(3)	(6)	(5)	(3,5,7)
		54.88	58.36	50.10	54.10	33.33	57.19	32.49	51.14	**66.83**
	Pima	64.71	64.71	65.36	66.02	65.10	64.71	65.23	64.71	64.58
		(2,7)	(1,2,4,6-8)	(2)	(7)	(4)	(2)	(2,7)	(5)	(2)
		56.95	64.34	63.95	57.86	60.90	60.12	60.16	**64.49**	57.03
	Optdigits	87.90	49.60	52.00	85.40	22.50	84.10	17.60	46.30	55.90
		(8)	(5)	(5)	(1)	(5)	(3)	(1)	(3)	(5)
		69.75	31.06	39.34	**74.42**	17.46	71.18	14.53	43.91	38.61
	Iris	84.00	90.67	70.67	90.67	58.67	94.00	49.33	53.33	90.67
		(5,8)	(3)	(3)	(2,8)	(1)	(1)	(1)	(2,4-8)	(1-3,7,8)
		63.17	84.83	62.08	68.42	45.17	86.58	39.75	53.00	**90.42**
	Crabs	65.00	56.00	58.00	65.00	57.00	70.50	54.00	67.00	70.50
		(2)	(3)	(1,5)	(2)	(5)	(2)	(3)	(7)	(4,6)
		59.94	53.12	53.31	57.37	52.50	55.87	51.56	58.00	**62.81**
	Breast Cancer	62.96	68.81	58.13	64.86	64.86	94.58	74.23	75.55	68.67
		(2)	(5)	(2,3,7,8)	(2-4,6-8)	(2,6,7)	(4-8)	(1)	(8)	(1)
		56.81	61.11	56.44	61.68	60.65	**86.09**	64.81	67.84	60.45
	House Votes	89.22	88.36	81.90	87.93	81.47	87.07	61.21	64.66	74.14
		(1)	(1)	(5)	(1)	(1)	(3)	(5)	(1)	(1)
		74.52	71.28	63.31	**73.81**	59.37	69.34	54.69	57.81	62.88

This suggest that the co-association matrix might not be the best clustering ensemble approach for this dataset.

Similar to the feature space, the analysis of NMI values for synthetic datasets with touching clusters (figure 3 middle) suggests that no DR algorithm is robust to the choice of clustering algorithm; except the EA-Sammon in the Mixed Image 3. For the real datasets (figure 3 bottom) EA-LE is the DR algorithm with the most consistent results, except for the Pima, Crabs and Breast cancer datasets.

The best overall DR methods, for the synthetic datasets with separate clusters, are EA-LE and EA-LLE. EA-Isomap, EA-CCA, EA-CDA and EA-LE yield the best results with single-link. For the synthetic datasets with touching clusters, the best DR methods are EA-Isomap and EA-LE, when used with the appropriate clustering algorithm.

For the Std Yeast dataset the worst results correspond to nonlinear local DR methods (EA-LLE and EA-LE). For the Optdigits dataset, the worst results correspond to local methods (EA-LPP, EA-NPE, EA-LLE and EA-LE), while nonlinear global methods perform very well. In the House votes dataset, the best DR algorithms in average CI are linear methods (EA-LPP and EA-NPE) and nonlinear global methods that preserves "geodesic" distances (EA-Isomap and EA-CDA). These last two algorithms also have very good results for the Breast cancer dataset.

From Table 2, we notice that there exists at least one DR method that outperforms or equals EAC for each dataset, showing that there is an advantage in performing DR.

Like in the feature space, single-link is the best extraction method, except for real datasets. In real datasets, k-means and Ward link work better.

Overall, nonlinear methods are more suitable for this space, with local methods working better in synthetic data with separate clusters.

6 Discussion

There are some interesting findings to draw from all the above data. First, there is an advantage in using DR techniques on the co-association matrix to improve clustering results. However, care must be taken in choosing the right DR technique for each dataset.

Second, the use of DR techniques usually improves the average consistency index (CI) values over the co-association matrix. This suggests that using DR makes the clustering results less dependent on the choice of the specific clustering algorithm.

Although no DR algorithm consistently outperforms all the others, some algorithms do well in specific circumstances. Good results are obtained from datasets with separate clusters using LPP and LE (local DR methods). For datasets with touching clusters, Isomap and LE (nonlinear DR methods) yield the overall best results. Importantly, in real datasets no DR algorithm stood out from the others, and considerable variability was detected from dataset to dataset, again stressing out that the choice of the appropriate DR technique is crucial.

To further investigate this aspect, we have computed the measures N1[3] and silhouette for the real datasets studied in this paper. Those values are presented in table 3.

[3] As explained in [8] "This method constructs a class-blind minimum spanning tree over the entire dataset, and counts the number of points incident to an edge going across the two classes. The fraction of such points over all points in the dataset is used as the N1 measure."

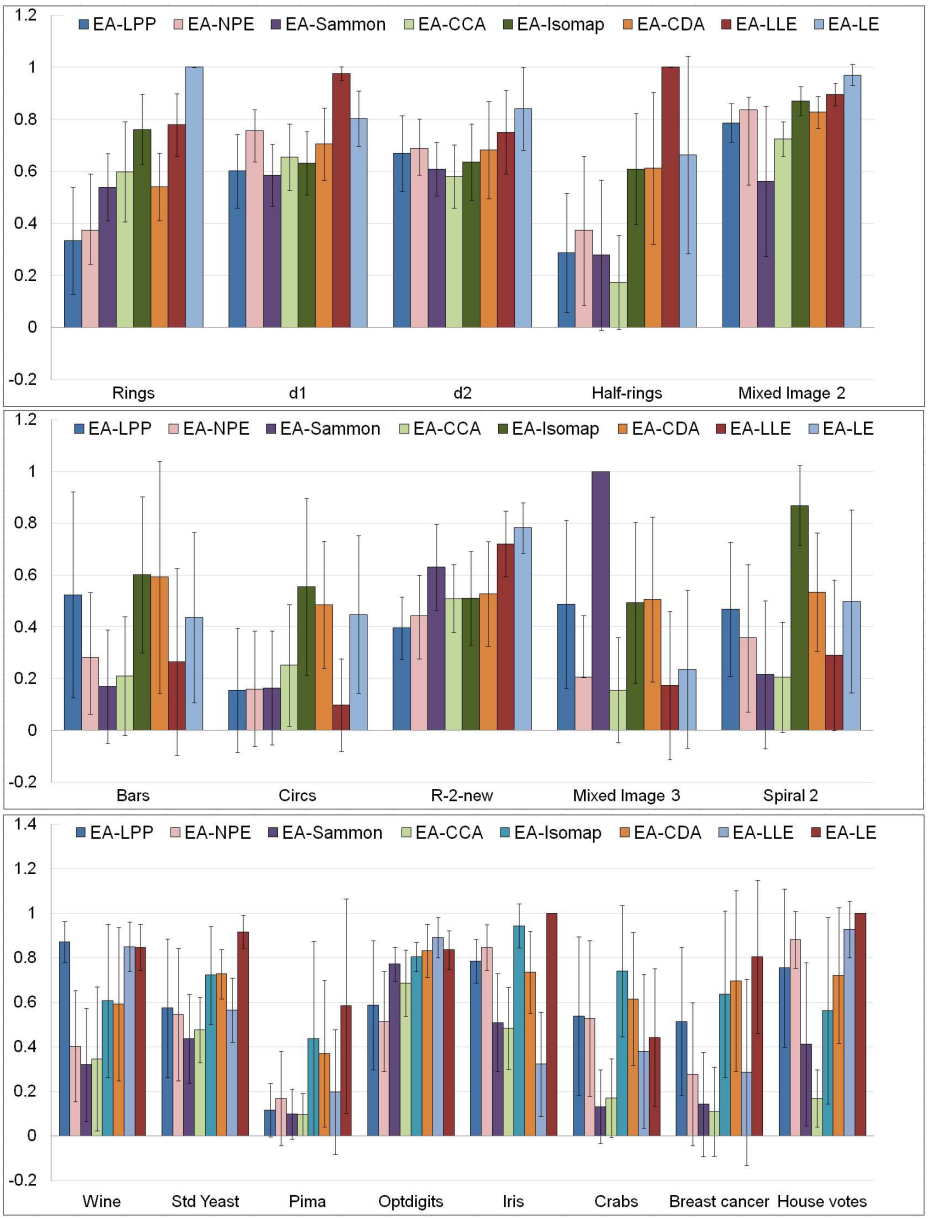


Fig. 3. Mean and standard deviation of Normalized Mutual Information over the clustering algorithms for each dataset and each embedding method. The co-association matrix was interpreted as similarities. *Top:* Synthetic datasets with separate clusters. *Middle:* Synthetic datasets with touching clusters. *Bottom:* Real datasets.

Table 2. Consistency index (%) for co-association matrix interpreted as similarities. *(First row)* Best CI and clustering algorithm(s) which yield that CI value. Legend: (1) k-means, (2) single-link, (3) complete-link, (4) average-link, (5) Ward-link, (6) centroid-link, (7) median-link, (8) weighted-link. *(Second row)* Average CI (%) over all clustering methods. The gray cells correspond to the best NMI presented in figure 3 and the best average CI are shown in bold.

		EAC	EA-LPP	EA-NPE	EA-Sammon	EA-CCA	EA-Isomap	EA-CDA	EA-LLE	EA-LE
Synthetic data with separate clusters	Rings	100	74.00	74.00	77.50	77.50	63.25	79.00	81.00	100
		(2,4,8)	(2)	(2)	(1)	(2)	(7,8)	(1)	(7,8)	(1-8)
		74.79	58.69	56.59	70.41	68.44	61.47	72.53	73.50	**100**
	d1	100	100	90.50	100	100	100	100	90.00	100
		(2,4-8)	(2,4)	(2)	(2)	(2)	(2)	(2,7)	(2)	(2)
		94.07	74.31	69.06	59.62	61.31	67.62	77.06	**87.81**	71.75
	d2	100	100	61.50	66.50	100	88.50	100	100	79.00
		(2)	(2)	(2)	(4)	(2)	(2)	(2)	(2)	(2)
		70.21	59.87	48.56	59.31	60.75	60.06	59.94	56.50	**64.50**
	Half-rings	100	94.75	88.00	81.75	93.25	100	100	100	100
		(2,4,8)	(4)	(8)	(2)	(6)	(2)	(2)	(1-8)	(2,4-8)
		82.86	81.06	80.12	64.59	68.69	79.62	72.41	**100**	90.84
	Mixed Image 2	72.40	67.50	67.70	60.00	70.80	71.00	70.80	66.90	68.10
		(8)	(6)	(2)	(1)	(2)	(2)	(2)	(2)	(2-4,6-8)
		53.34	60.10	61.46	50.72	60.31	63.45	62.81	64.09	**67.05**
Synthetic data with touching clusters	Bars	99.25	100	75.25	69.75	99.50	99.50	74.00	99.00	99.25
		(4)	(5,8)	(1)	(4,6)	(6)	(3,5)	(1)	(4)	(5)
		74.25	88.69	65.53	61.00	77.53	**90.28**	61.84	77.66	69.84
	Circs	99.50	81.00	78.75	82.25	71.00	99.50	99.50	78.75	99.50
		(2,4,5)	(3)	(5)	(1)	(3)	(1,4-6)	(2)	(5)	(2,5)
		76.54	63.50	62.47	66.22	61.37	**88.97**	73.78	63.37	76.47
	R-2-new	89.20	58.80	58.80	65.80	60.60	63.20	79.80	59.80	80.60
		(5)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(8)
		65.77	44.32	47.55	60.32	45.62	45.12	44.92	53.77	**67.80**
	Mixed Image 3	88.70	92.40	75.00	50.10	85.10	89.60	91.90	82.60	76.10
		(5)	(5)	(5)	(1-8)	(5)	(4)	(3)	(1)	(5)
		67.14	**82.00**	66.34	50.10	68.42	79.95	**82.00**	60.31	68.12
	Spiral 2	85.00	56.33	55.67	65.33	77.67	84.00	91.33	60.33	85.00
		(2)	(4,5,7)	(5,8)	(1)	(2)	(1,5)	(7,8)	(7)	(2,5,7,8)
		63.43	54.29	53.67	58.54	61.00	79.25	**81.92**	54.75	78.79
Real data	Wine	93.82	98.31	73.03	97.75	97.19	94.94	94.94	91.01	91.57
		(8)	(3)	(5)	(1)	(1)	(5)	(4,6)	(2)	(3-6)
		72.12	**92.84**	61.45	70.86	68.40	82.80	82.94	85.74	86.24
	Std Yeast	67.71	72.14	72.14	72.92	72.40	67.45	72.40	51.04	63.28
		(4)	(8)	(5)	(4)	(4)	(7)	(3)	(5)	(4-6,8)
		51.79	**63.38**	59.89	50.13	52.11	60.03	60.87	41.89	61.36
	Pima	65.10	71.35	65.63	64.71	68.49	64.71	64.71	65.76	64.71
		(6,7)	(7)	(6)	(2,4)	(4)	(2,3,6-8)	(2)	(7)	(2-4,6-8)
		62.91	**65.74**	61.95	60.81	60.03	62.04	58.41	64.13	63.49
	Optdigits	80.70	56.60	23.60	81.90	82.70	82.60	80.90	47.10	72.00
		(5)	(5)	(1)	(5)	(5)	(5)	(5)	(5)	(5)
		55.41	43.86	20.92	70.91	64.61	70.74	**72.30**	36.24	60.35
	Iris	90.67	90.00	95.33	89.33	90.67	94.67	90.67	79.33	90.67
		(2,4,5,8)	(4,6)	(1,3,8)	(5)	(2)	(1)	(2)	(1)	(1-8)
		75,62	83.92	88.75	70.75	67.75	**91.17**	71.83	57.25	90.67
	Crabs	71.00	54.00	88.00	70.50	71.00	71.00	71.00	66.00	74.50
		(2)	(1)	(1,4-6)	(5)	(2)	(2)	(2)	(3)	(5)
		57.56	52.06	**78.31**	56.13	56.87	56.87	56.44	62.12	63.44
	Breast Cancer	69.84	95.75	81.41	94.29	85.65	97.07	97.22	88.43	96.05
		(3)	(1,4)	(1)	(1)	(4)	(1)	(1)	(5)	(1)
		62.12	88.54	71.34	75.35	65.96	92.22	**92.90**	72.29	64.79
	House Votes	88.36	90.09	90.09	89.22	94.40	88.36	89.22	59.91	66.81
		(4)	(1)	(4-6)	(3,4)	(4)	(3)	(1)	(3)	(1-8)
		68.53	84.80	**88.79**	72.90	70.53	81.14	85.67	59.54	66.81

Datasets Std Yeast and Pima stand out for having high values of N1, and in those datasets local DR methods yield the best clustering results in terms of average CI. On the other hand, datasets Optdigits and Breast Cancer stand out for having low values of N1 and the best results in those datasets come from global DR methods. Also, Crabs and Std Yeast have low values of the silhouette index and local DR methods perform well with these datasets. Given the relatively small number of datasets and DR methods used in this paper, we present these associations not as proven rules, but rather as temporary guidelines. We will actively research these types of associations using more datasets and more DR methods in the future.

Table 3. N1 and Silhouette measures for the real datasets studied in this paper, and type of DR method that yields the best average CI for both types of spaces (feature and similarity spaces). The question mark (?) indicates datasets where the best DR type is different in the two spaces.

Real Datasets	N1	Silhouette	Best DR type
Wine	0.118	0.4368	local
Std Yeast	0.388	0.2274	local
Pima	0.438	0.1524	local
Optdigits	0.059	0.2892	global
Iris	0.100	0.6565	?
Crabs	0.160	0.0442	local
Breast Cancer	0.057	0.7178	global
House Votes	0.159	0.4471	?

There are some differences between using the co-association matrix as features or as similarities. For example, CCA and CDA perform poorly in the former case but considerably better in the latter. On the other hand, Sammon performs better in the feature space relative to the similarity space.

It is interesting to note that the DR algorithms which have the highest NMI values for each dataset are very often the ones which have also the highest average CI values. In other words, it seems that the DR algorithms which yield the most consistent partitions also yield the best partitions. Furthermore, for each dataset, the highest NMI between the feature space and the similarity space very often corresponds to the highest average CI as well. This suggests that NMI (a measure which does not need to know the true partition) can help predict the CI (which does use the true partition).

7 Conclusions

This study shows that the use of dimensionality reduction (DR) techniques in clustering ensembles presents interesting advantages in accuracy and robustness. Future work is needed to study the influence of different strategies to construct the clustering ensemble, and the influence of parameter choice for the DR and clustering algorithms.

We also reported some interesting associations between types of datasets and appropriate DR methods; however, further work is needed to draw conclusive information.

8 Acknowledgments

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). This work was partially supported by the Portuguese Foundation for Science and Technology (FCT), scholarship number SFRH/BD/39642/2007, and grant PTDC/EIA-CCO/103230/2008.

References

1. Ayad, H.G., Kamel, M.S.: Cluster-based cumulative ensembles. In: Proc. of the 6th Int. Workshop on Multiple Classifier Systems (MCS2005) (2005)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Advances in Neural Information Processing Systems (NIPS 2001). vol. 14, pp. 585–591 (2002)
3. Demartines, P., Hérault, J.: Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. IEEE Trans. on Neural Networks 8(1), 148–154 (1997)
4. Fred, A.: Finding consistent clusters in data partitions. In: Proc. of the 2nd Int. Workshop on Multiple Classifier Systems (MCS 2001). pp. 309–318 (2001)
5. Fred, A., Jain, A.: Combining multiple clusterings using evidence accumulation. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(6), 835–850 (2005)
6. He, X., Cai, D., Yan, S., Zhang, H.J.: Neighborhood preserving embedding. In: Proc. of the 10th Int. Conf. on Computer Vision (ICCV 2005). vol. 2, pp. 1208–1213 (2005)
7. He, X., Niyogi, P.: Locality preserving projections. In: Advances in Neural Information Processing Systems (NIPS 2003). vol. 16 (2004)
8. Ho, T.K., Basu, M., Law, M.H.C.: Data Complexity in Pattern Recognition, Advanced Information and Knowledge Processing, vol. 16, chap. Measures of Geometrical Complexity in Classification Problems, pp. 3–23. Springer, 1st edn. (2006)
9. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3), 264–323 (1999)
10. Kuncheva, L.I., Hadjitodorov, S.T.: Using diversity in cluster ensembles. In: Proc. of the Int. Conf. on Systems, Man and Cybernetics. vol. 2, pp. 1214–1219 (2004)
11. Kuncheva, L.I., Hadjitodorov, S.T., Todorova, L.P.: Experimental comparison of cluster ensemble methods. In: Proc. of the 9th Int. Conf. on Information Fusion (FUSION 2006) (2006)
12. Lee, J.A., Verleysen, M.: Nonlinear Dimensionality Reduction. Information Science and Statistics, Springer (2007)
13. Lee, J.A., Lendasse, A., Verleysen, M.: Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. Neurocomputing 57, 49–76 (2004)
14. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
15. Sammon, J.W.: A nonlinear mapping for data structure analysis. IEEE Trans. on Computers 18(5), 401–409 (1969)
16. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3, 583–617 (2002)
17. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
18. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Elsevier Academic Press (2003)

[5] - (no technical report)

Duarte, J.M.M., Fred, A.L.N., Duarte, J.F.: Combining data clusterings with instance level constraints. In Fred, A., ed.: Intl. Workshop on Pattern Recognition in Information Systems, Milan, Itally, INSTICC Press (2009) 49–60

Combining Data Clusterings with Instance Level Constraints

João M. M. Duarte[1,2], Ana L. N. Fred[2], and F. Jorge F. Duarte[1]

[1] GECAD - Knowledge Engineering and Decision Support Group,
Instituto Superior de Engenharia do Porto, Instituto Superior Politécnico, Porto, Portugal
{jmmd, fjd}@isep.ipp.pt
[2] Instituto de Telecomunicações,
Instituto Superior Técnico, Lisboa, Portugal
afred@lx.it.pt

Abstract. Recent work has focused the incorporation of *a priori* knowledge into the data clustering process, in the form of pairwise constraints, aiming to improve clustering quality and find appropriate clustering solutions to specific tasks or interests. In this work, we integrate must-link and cannot-link constraints into the cluster ensemble framework. Two algorithms for combining multiple data partitions with instance level constraints are proposed. The first one consists of a modification to Evidence Accumulation Clustering and the second one maximizes both the similarity between the cluster ensemble and the target consensus partition, and constraint satisfaction using a genetic algorithm. Experimental results shown that the proposed constrained clustering combination methods performances are superior to the unconstrained Evidence Accumulation Clustering.

1 Introduction

Data clustering is an unsupervised technique that aims to partition a given data set into groups or clusters, based on a notion of similarity or proximity between data patterns. Similar data patterns are grouped together while heterogeneous data patterns are grouped into different clusters. Data clustering techniques can be used in several applications including exploratory pattern-analysis, decision-making, data mining, document retrieval, image segmentation and pattern classification [1]. Despite a large number of clustering algorithms have been proposed, none can discover all sorts of cluster shapes and structures.

In the last decade, cluster ensembles approaches have been introduced based on the idea of combining information from multiple clusterings results to improve data clustering robustness [2], reuse clustering solutions [3] and cluster data in a distributed way. The main proposals to solve the cluster ensemble problem are based in: co-associations between pairs of patterns [2, 4, 5], graphs [6], hyper-graphs [3], mixture models [7] and the search for a median partition that summarizes the cluster ensemble [8].

A recent and very promising area is constrained data clustering [9], allowing the incorporation of *a priori* knowledge about the data set into the clustering process. This knowledge is mapped as constraints to express preferences, limitations and/or conditions to be imposed in data clustering, making it more useful and appropriate to specific

tasks or interests. The constraints can be set on a more general level using rules that are applied to the entire data set, such as data clustering with obstacles [10], at an intermediate level, where they are applied to data features [11] or to groups' characteristics, such as, the minimum and maximum capacity [12], or at a more specific level, where the constraints are applied to data patterns, using labels on some data [13] or the relations between pairs of patterns [11]. Relations between pairs of patterns (must-link and cannot-link constraints) have been the most studied due to their versatility, because many constraints on more general levels can also be represented by relations between pairs of patterns. Several constrained data clustering algorithms were proposed concerning various perspectives: inviolable constraints [11], distance editing [14], partial label data [13], constraints violation penalty [15] and modification of the generation model [13].

In this paper we propose to integrate pairwise constraints into the clustering ensemble framework. We build on previous work on Evidence Accumulation Clustering and propose a new approach based on maximizing the Average Cluster Consistency and Constraint Satisfaction measures using a genetic algorithm.

The rest of this paper is organized as follows. Section 2 presents the cluster ensemble problem formulation and describes the Evidence Accumulation Clustering. We propose an extension to Evidence Accumulation Clustering Approach in Section 3. Section 4 presents a new approach to constrained clustering combination using a genetic algorithm. We describe the experimental setup used to assess the performance of the proposed approaches in Section 5 and the results are shown in Section 6. Finally, Section 7 concludes this paper.

2 Background

2.1 Problem Formulation

Let $\mathcal{X} = \{x_1, \cdots, x_n\}$ be a set of n data patterns and let $P = \{C_1, \cdots, C_K\}$ be a partition of $\mathcal{X}$ into K clusters. A cluster ensemble $\mathcal{P}$ is defined as a set of N data partitions P^l of $\mathcal{X}$:

$$\mathcal{P} = \{P^1, \cdots, P^N\}, P^l = \{C_1^l, \cdots, C_{K^l}^l\}, \tag{1}$$

where C_k^l is the k^{th} cluster in data partition P^l, which contains K^l clusters, with $\sum_{k=1}^{K^l} |C_k^l| = n, \forall l \in \{1, \cdots, N\}$.

There are two fundamental phases in combining multiple data partitions: the partition generation mechanism and the consensus function, that is, the method that combines the N data partitions in $\mathcal{P}$. There are several ways to generate a cluster ensemble $\mathcal{P}$, such as, producing partitions of $\mathcal{X}$ using different clustering algorithms, changing parameters initialization for the same clustering algorithm, using different subsets of data features or patterns, projecting $\mathcal{X}$ to subspaces and combinations of these. A consensus function f maps a cluster ensemble $\mathcal{P}$ into a consensus partition P^*, $f : \mathcal{P} \rightarrow P^*$, such that P^* should be consistent with $\mathcal{P}$ and robust to small variations in $\mathcal{P}$.

In this work we focus on combining multiple data partitions into a more robust consensus partition using *a priori* information in terms of pairwise relations. These relations between pair of patterns are represented by two sets of constraints: must-link ($\mathcal{C}_{=}$) and cannot-link ($\mathcal{C}_{\neq}$) constraint sets. A must-link constraint between x_i and x_j data patterns, i.e. $(x_i, x_j) \in \mathcal{C}_{=}$, indicates that x_i and x_j should belong to the same cluster in the clustering solution and a cannot-link constraint, i.e. $(x_i, x_j) \in \mathcal{C}_{\neq}$, points that x_i should not be placed in the cluster of x_j. These instance level constraints can be seen as hard or soft constraints. When $\mathcal{C}_{=}$ and $\mathcal{C}_{\neq}$ are defined as hard constraint sets, if $(x_i, x_j) \in \mathcal{C}_{=}$ then both data patterns *must* belong to the same cluster in the clustering solution and if $(x_i, x_j) \in \mathcal{C}_{\neq}$ these patterns *cannot* be grouped into the same cluster. When $\mathcal{C}_{=}$ and $\mathcal{C}_{\neq}$ are defined as soft constraint sets, must-link and cannot-link constraints can be thought as preferences of grouping (x_i, x_j) into the same cluster or into different clusters, but not an obligation. Is this work we explore both types of constraints.

2.2 Evidence Accumulation Clustering

Evidence Accumulation Clustering (EAC) [2] considers each data partition $P^l \in \mathcal{P}$ as an independent evidence of data organization. The underlying assumption of EAC is that two patterns belonging to the same *natural* cluster will be frequently grouped together. A vote is given to a pair of patterns every time they co-occur in the same cluster. Pairwise votes are stored in a $n \times n$ co-association matrix and are normalized by the total number of data partitions to combine:

$$co\_assoc_{ij} = \frac{\sum_{l=1}^{N} vote_{ij}^{l}}{N}, \tag{2}$$

where $vote_{ij}^{l} = 1$ if x_i and x_j belong to the same cluster C_k^l in the l^{th} data partition P^l, otherwise $vote_{ij}^{l} = 0$. This voting mechanism avoids the need of making the correspondence between clusters in different partitions because only relation between pairs of patterns are considered. The resulting co-association matrix corresponds to a non-linear transformation of the original feature space of $\mathcal{X}$ into a new representation defined in co_assoc, which can be viewed as new inter-pattern similarity measure. In order to produce the consensus partition one can apply any clustering algorithm over the co-association matrix co_assoc.

3 Constrained Evidence Accumulation Clustering

Our first approach for combining multiple data clusterings using must-link and cannot-link constraints consists of a simple extension of EAC, hereafter referred as Constrained Evidence Accumulation (CEAC). As seen in subsection 2.2, the consensus partition is obtained by applying a data clustering algorithm to co_assoc. The EAC extension requires that this clustering algorithm supports the incorporation of instance level constraints (in this paper, in the form of must-link and cannot-link constraints).

We used two (hard) constrained data clustering algorithms to extract the consensus partition from co_assoc. The first one, Constrained Complete-Link (CCL) [14], is a

constrained agglomerative clustering algorithm that modifies a $(n \times n)$ dissimilarity matrix, D, to reflect the pairwise constraints and then applies the well-known complete-link algorithm to the modified distance matrix to obtain the data partition. The modified distance matrix is computed in three steps: set all must-linked data patterns distances to 0, $\forall (x_i, x_j) \in \mathcal{C}_{=} : D_{i,j} = D_{j,i} = 0$; compute shortest paths between data patterns with D; impose cannot-link constraints, $\forall (x_i, x_j) \in \mathcal{C}_{\neq} : D_{i,j} = D_{j,i} = \max(D) + 1$. Cannot-link constraints are implicitly propagated by the complete-link algorithm. In order to use the CCL in the CEAC, each entry of the input dissimilarity matrix D is computed as $D_{ij} = 1 - co\_assoc_{ij}$ since the co_assoc is a similarity matrix with values in the interval $[0, 1]$.

The second data clustering algorithm used to extract the consensus partition is a modification of the single-link algorithm: at the beginning all must-linked patterns are grouped into the same clusters and then, iteratively, the closest pair of clusters (C_a, C_b) such that $\nexists (x_i, x_j), x_i \in C_a, x_j \in C_b$ and $(x_i, x_j) \in \mathcal{C}_{\neq}$ is merged. From now on this algorithm is referred as Constrained Single-Link (CSL). Algorithm 1 summarizes the Constrained Evidence Accumulation Clustering.

Algorithm 1 Constrained Evidence Accumulation

procedure CEAC($\mathcal{P}$, $\mathcal{C}_{=}$, $\mathcal{C}_{\neq}$, N, n) ▷ Where $\mathcal{P} = \{P^1, \cdots, P^N\}$, N is the number of clusterings to combine and n is the number of data patterns
  Set co_assoc as a $n \times n$ null matrix ▷ Co-association matrix initialization
  for $l \leftarrow 1, N$ **do**
    for all $C_k^l \in P^l$ **do** ▷ Update co-association matrix
      for all $(x_i, x_j) \in C_k^l$ **do**
        $co\_assoc_{ij} \leftarrow co\_assoc_{ij} + 1$
      end for
    end for
    for $i = 1 : n$ **do** ▷ Normalize co-association matrix
      $co\_assoc_{ij} \leftarrow \frac{co\_assoc_{ij}}{N}$
    end for
  end for
  $P^* \leftarrow$ CONSTRAINEDCLUSTERER(co_assoc, $\mathcal{C}_{=}$, $\mathcal{C}_{\neq}$) ▷ Produce consensus partition
  return P^*
end procedure

4 Average Cluster Consistency and Constraint Satisfaction (ACCCS approach)

Our second proposal to combine multiple data clusterings consists of maximizing an objective-function $\jmath_{ACCCS}$ based on Average Cluster Consistency (ACC) [16] and Constraints Satisfaction (CS) measures using a genetic algorithm. These are described in the next subsections.

4.1 Average Cluster Consistency

Average Cluster Consistency index measures the average similarity between each data partition in the cluster ensemble ($P^l \in \mathcal{P}$) and a target consensus partition P^*, assuming that the number of clusters of each partition in $\mathcal{P}$ is equal or greater than the number of clusters in P^*. The notion of similarity between two partitions P^* and P^l is based on the following idea: P^l is similar to P^* if each cluster $C_k^l \in P^l$ is contained by a cluster $C_m^* \in P^*$. Taking this notion in mind, we define the similarity between two partitions as:

$$sim(P^*, P^j) = \frac{\sum_{m=1}^{K^j} \max_{1 \leq k \leq K^*}(|Inters_{k,m}|) \times (1 - \frac{|C_k^*|}{n})}{n}, K^j \geq K^*, \quad (3)$$

where $|Inters_{k,m}|$ is the cardinality of the set of patterns common to the k^{th} and m^{th} clusters of P^* and P^j, respectively ($Inters_{k,m} = \{x_a | x_a \in C_k^* \wedge x_a \in C_m^j\}$). Note that in Eq. 3, $|Inters_{k,m}|$ is weighted by $(1 - \frac{|C_k^*|}{n})$ in order to prevent cases were P^* have clusters with almost all data patterns to have a high value of similarity. The Average Cluster Consistency between $\mathcal{P} = \{P^1, \cdots, P^N\}$ and P^* is then defined as

$$ACC(P^*, \mathcal{P}) = \frac{\sum_{i=1}^{N} sim(P^i, P^*)}{N}. \quad (4)$$

4.2 Algorithm Description

In addition to optimize ACC (Eq. 4) we also consider the consensus partition Constraints Satisfaction $CS(P^*, \mathcal{C}_=, \mathcal{C}_\neq)$ defined as the fraction of constrains satisfied by the consensus partition P^*:

$$CS(P^*, \mathcal{C}_=, \mathcal{C}_\neq) = \frac{\sum_{(x_i,x_j) \in \mathcal{C}_=} I(c_i = c_j) + \sum_{(x_i,x_j) \in \mathcal{C}_\neq} I(c_i \neq c_j)}{|\mathcal{C}_=| + |\mathcal{C}_\neq|} \quad (5)$$

where $|\mathcal{C}_=|$ and $|\mathcal{C}_\neq|$ are, respectively, the number of must-link and cannot-link constrains, $I(\cdot)$ takes value 1 if its expression is true, taking value 0 otherwise, and $c_i = C_k^*, x_i \in C_k^*$.

We define our objective-function $\jmath_{ACCCS}$ as the weighted mean of ACC and CS and it is formally defined as:

$$\jmath_{ACCCS}(P^*, \mathcal{P}, \mathcal{C}_=, \mathcal{C}_\neq) = (1 - \beta) ACC(P^*, \mathcal{P}) + \beta CS(P^*, \mathcal{C}_=, \mathcal{C}_\neq), \quad (6)$$

where $0 \leq \beta \leq 1$ is weighting coefficient that controls the importance of satisfying must-link and cannot-link constraints. Note that in this approach constraint sets are thought as soft constrains.

In order to produce the consensus function P^*, we propose the maximization of Eq. 6 using a genetic algorithm (GA). GA is a search technique inspired by evolutionary biology used to find approximate best solutions of optimization problems. Candidate solutions are represented by a population of individuals that are recombined and possibly mutated to create new individuals (candidate solutions). The fittest individuals

(based on a fitness or objective function) are selected to belong to next generation until a stopping criterium is reached. Our fitness function is $\jmath_{ACCCS}$ (Eq. 6). Our genetic algorithm is described next. First, the initial population $\mathcal{B}^0$, i.e. a set of $PopSize$ data partitions $\mathcal{B}^0 = \{b_1^0, \cdots, b_{PopSize}^0\}$, is generated. Initial population individuals can be randomly generated, but we used the K-means algorithm to generate it, in order to start the solution search (probably) closer to the optimal solution. After $\mathcal{B}^0$ is built, the algorithm iterates the following 4 steps until a specified maximal number of generations $MaxGen$ is reached.

Selection $PopSize$ individuals b_j^t are selected from $\mathcal{B}^t$. Individual selection probability is proportional to its fitness function value $\jmath_{ACCCS}$ and is defined as

$$Pr_{sel}(b_j^t) = \frac{\jmath_{ACCCS}(\mathcal{P}, b_j^t, \mathcal{C}_=, \mathcal{C}_{\neq})}{\sum_{i=1}^{PopSize} \jmath_{ACCCS}(\mathcal{P}, b_i^t, \mathcal{C}_=, \mathcal{C}_{\neq})}. \tag{7}$$

Note that an individual b_j^t can be selected several times.

Crossover Previously selected individuals (parents) are grouped in pairs and are randomly split and merged producing new individuals (children). This process is done by cutting the pair of data partitions that represents the individuals at a randomly chosen vector position $CrossoverPoint \in \{1, \cdots, n\}$ and then swap the two tails of the vectors, as shown in Fig. 1. Note that it is necessary to match the clusters of the data partitions before this step occurs.

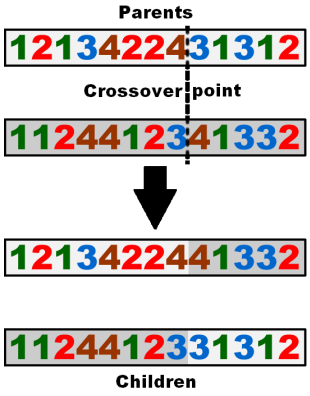


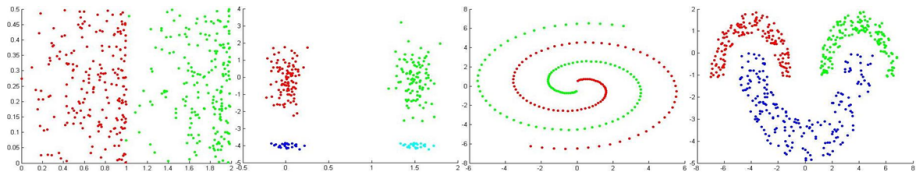
Fig. 1: Crossover example.

Mutation In this step, pattern labels in each clustering solution (individual) can be changed (mutated). The mutation probability $MutationProb$ is usually very low, to prevent the algorithm search from being random.

Sampling Finally, $PopSize$ individuals with best fitness (i.e. highest $\jmath_{ACCCS}$ value) are selected for the next generation $\mathcal{B}^{t+1}$.

5 Experimental Setup

We used 4 synthetic and 8 real data sets to assess the quality of the cluster ensemble methods on a wide variety of situations, such as data sets with different cardinality and dimensionality, arbitrary shaped clusters, well separated and touching clusters and distinct cluster densities. A brief description for each data set is given below.



(a) Bars (b) Cigar (c) Spiral (d) Half Rings

Fig. 2: Synthetic data sets.

Synthetic Data Sets Fig. 2 presents the 2-dimensional synthetic data sets used in our experiments. Bars data set is composed by two clusters very close together, each with 200 patterns, with increasingly density from left to right. Cigar data set consists of four clusters, two of them having 100 patterns each and the other two groups 25 patterns each. Spiral data set contains two spiral shaped clusters with 100 data patterns each. Half Rings data set is composed by three clusters, two of them have 150 patterns and the third one 200.

Real Data Sets The 8 real data sets used in our experiments are available at UCI repository (http://mlearn.ics.uci.edu/MLRepository.html). The first one is Iris and consists of 50 patterns from each of three species of Iris flowers (setosa, virginica and versicolor) characterized by four features. One of the clusters is well separated from the other two overlapping clusters. Breast Cancer data set is composed of 683 data patterns characterized by nine features and divided into two clusters: benign and malignant. Yeast Cell data set consists of 384 patterns described by 17 attributes, split into five clusters concerning five phases of the cell cycle. There are two versions of this dataset, the first one is called Log Yeast and uses the logarithm of the expression level and the other is called Std Yeast and is a "standardized" version of the same data set, with mean 0 and variance 1. Optdigits is a subset of Handwritten Digits data set containing only the first 100 objects of each digit, from a total of 3823 data patterns characterized by 64 attributes. Glass data set is composed of 214 data patterns, concerning to 6 types of glass six types of glass, characterized by their chemical composition on 9 attributes. Wine data set consists of tree clusters (with 59, 71 and 48 data patterns) of wines grown in the same region in Italy but derived from three different cultivars. Its features are the quantities of 13 constituents found in each type of wine. Finally, Image Segmentation data set consists of 2310 data patterns with 19 features, where each pattern is a 3×3 pixels image segment randomly obtained from seven outdoor images.

We artificially built several constraint sets of must-link and cannot-link constraints. For each data set, $NumConstr \in \{10, 20, 50, 100, 200\}$ pairs of patterns (x_i, x_j), $x_i \neq x_j$ were randomly chosen. If x_i and x_j belonged to the same cluster in the *real* data partition, P^0, the pair was added to the must-link constraint set, i.e. $\mathcal{C}_= = \mathcal{C}_= \cup \{(x_i, x_j)\}$. Otherwise the pair of patterns was added to the cannot-link constraint set ($\mathcal{C}_\neq = \mathcal{C}_\neq \cup \{(x_i, x_j)\}$).

For each possible combination of data set, clustering combination method and constraint set we built 20 cluster ensembles. Each cluster ensemble was composed by $N = 50$ data partitions obtained using K-means clustering algorithm and randomly choosing the number of clusters K to be an integer number in the set $K \in \{10, \cdots, 30\}$ in order to create diversity.

The number of clusters K^* of the consensus partition P^*, for all clustering combination methods, was defined as the *real* number of clusters K^0. In EAC, the well-known Single-Link (SL) and Complete-Link (CL) algorithms were used to extract P^* from co_assoc. We used constrained versions of SL and CL to produce P^* in the CEAC approach, as described in Section 3. For j_{ACCCS} maximization using the genetic algorithm approach we set the stopping criterium to 100 generations, population size to 20, crossover probability to 80%, mutation probability to 1% and $\beta = \frac{1}{2}$. The initial population was obtained using K-means algorithm.

In order to evaluate the quality of the proposed clustering combination methods we used the Consistency index (Ci) [2]. Ci measures the fraction of shared data patterns in matching clusters of the consensus partition (P^*) and the *real* data partition (P^0) obtained from known labeling of data. Formally, the Consistency index is defined as

$$Ci(P^*, P^0) = \frac{1}{n} \sum_{k=1}^{\min\{K^*, K^0\}} |C_k^* \cap C_k^0| \tag{8}$$

where $|C_k^* \cap C_k^0|$ is the cardinality of the P^* and P^0 k^{th} matching clusters data patterns intersection.

6 Results

Table 1 shows the results of the experiments concerning the clustering combination algorithms evaluation, described in Section 5. The first column indicates the data set, second column the number of constraints used for the constrained clustering combination algorithms and columns 3-7 the clustering combination algorithms. Rows in columns 3-7 show average and maxima (shown between parentheses) consistency index values in percentage, $Ci(P^*, P^0) \times 100$.

From the analysis of Bars results we see that the constrained clustering combination methods usually have higher average Ci than both EAC (using SL and CL algorithms to produce consensus partition) methods. ACCCS approach achieved the highest average Ci value for each constraint set but the absolute higher Ci value was obtained by CEAC using both CSL and CCL to extract from co_assoc the consensus partition. In Cigar data set we highlight the perfect EAC (using SL) and CEAC (using CSL with 200 constraints) average results. The ACCCS approach never achieved 100% and its

Table 1: Average and maxima consistency index values in percentage, $Ci(P^*, P^0) \times 100$ for EAC, CEAC and ACCCS approaches.

Data set	Number of constraints	EAC		CEAC				ACCCS	
		SL	CL	CSL		CCL			
Bars	10	76.45 (99.50)	53.93 (60.50)	94.09	(99.50)	64.85	(85.00)	98.70	(99.25)
	20			96.15	(99.50)	70.65	(94.00)	98.61	(99.25)
	50			95.40	(99.50)	69.71	(99.50)	98.31	(99.25)
	100			92.34	(100.0)	71.32	(99.50)	98.40	(99.50)
	200			92.69	(100.0)	85.43	(100.0)	98.72	(99.25)
Cigar	10	100.0 (100.0)	43.3 (62.40)	83.50	(90.00)	50.90	(62.80)	82.94	(98.40)
	20			90.50	(100.0)	52.80	(67.20)	80.06	(98.00)
	50			96.00	(100.0)	66.08	(83.20)	80.80	(98.40)
	100			99.00	(100.0)	85.00	(100.0)	88.06	(98.40)
	200			100.0	(100.0)	96.40	(100.0)	87.98	(99.20)
Spiral	10	75.11 (100.0)	53.05 (65.50)	94.83	(100.0)	55.75	(68.50)	55.52	(64.50)
	20			96.48	(100.0)	56.38	(67.00)	57.15	(68.00)
	50			98.00	(100.0)	59.80	(75.50)	58.30	(69.00)
	100			100.0	(100.0)	62.85	(88.50)	59.27	(65.00)
	200			100.0	(100.0)	77.48	(100.0)	63.37	(73.50)
Half Rings	10	97.26 (99.80)	45.68 (53.60)	88.54	(99.80)	55.99	(71.80)	78.01	(80.00)
	20			97.09	(99.80)	63.71	(83.00)	76.76	(80.40)
	50			98.21	(99.80)	74.47	(100.0)	75.58	(78.00)
	100			99.03	(100.0)	91.38	(100.0)	74.37	(80.80)
	200			98.91	(100.0)	94.59	(100.0)	77.79	(83.40)
Iris	10	69.87 (74.67)	59.72 (84.00)	79.27	(96.00)	66.60	(84.67)	87.63	(93.33)
	20			84.67	(96.00)	73.77	(94.67)	89.30	(91.33)
	50			89.17	(98.00)	74.00	(97.33)	89.80	(96.67)
	100			92.30	(98.67)	73.30	(98.67)	91.90	(96.67)
	200			96.63	(100.0)	79.27	(99.33)	95.87	(99.33)
Breast Cancer	10	83.88 (95.17)	62.75 (71.74)	85.69	(97.36)	64.24	(92.97)	90.41	(92.24)
	20			87.75	(97.07)	74.52	(97.07)	90.43	(92.24)
	50			91.76	(97.51)	69.16	(97.07)	89.99	(92.09)
	100			89.71	(97.36)	75.42	(96.34)	89.42	(93.70)
	200			94.14	(97.95)	73.79	(97.51)	90.52	(93.56)
Log Yeast	10	40.27 (45.31)	38.54 (47.14)	38.53	(45.31)	35.98	(42.19)	30.42	(33.33)
	20			42.68	(52.60)	37.97	(49.22)	29.61	(32.29)
	50			43.19	(56.51)	35.69	(45.05)	29.36	(31.25)
	100			44.92	(56.77)	39.13	(53.13)	29.90	(32.29)
	200			43.33	(55.21)	37.97	(47.40)	30.21	(34.90)
Std Yeast	10	48.95 (60.42)	46.59 (60.16)	50.56	(60.94)	39.74	(49.22)	63.61	(73.70)
	20			50.90	(61.72)	42.17	(54.95)	62.89	(73.18)
	50			54.31	(63.02)	39.32	(49.48)	62.60	(71.09)
	100			52.21	(64.58)	42.37	(57.55)	64.53	(69.79)
	200			50.39	(70.05)	40.79	(51.04)	66.17	(71.61)
Optdigits	10	54.62 (75.20)	56.81 (71.10)	30.20	(39.10)	61.58	(73.60)	78.27	(83.80)
	20			38.34	(49.20)	63.20	(73.50)	78.11	(83.20)
	50			51.13	(59.10)	61.63	(70.60)	77.21	(82.70)
	100			63.90	(75.40)	66.14	(77.00)	77.70	(82.20)
	200			79.40	(90.30)	70.24	(78.50)	78.64	(83.90)
Glass	10	43.94 (51.40)	39.42 (47.20)	46.17	(59.81)	39.56	(42.99)	46.14	(52.80)
	20			50.68	(62.15)	41.50	(53.74)	44.60	(48.60)
	50			53.86	(65.89)	45.07	(55.61)	43.36	(51.40)
	100			54.74	(64.02)	45.56	(55.14)	41.87	(45.79)
	200			60.07	(76.17)	44.98	(56.07)	42.66	(48.13)
Wine	10	70.64 (72.47)	51.03 (53.37)	63.85	(72.47)	49.55	(61.80)	65.48	(71.35)
	20			61.49	(70.79)	48.23	(62.36)	64.66	(71.91)
	50			53.57	(65.73)	50.51	(59.55)	68.54	(73.03)
	100			50.31	(64.04)	51.54	(65.73)	68.51	(73.03)
	200			61.80	(73.60)	53.85	(69.66)	72.92	(76.97)
Image Segmentation	10	27.68 (29.26)	42.41 (52.81)	42.21	(42.86)	50.52	(52.51)	49.55	(56.28)
	20			46.36	(51.65)	38.72	(40.52)	57.45	(58.66)
	50			51.95	(55.71)	45.69	(46.02)	52.58	(54.42)
	100			57.62	(65.28)	50.76	(54.29)	51.04	(54.42)
	200			66.75	(67.49)	51.97	(52.68)	52.16	(52.90)

best results was 99.2% of accuracy with 200 constraints. CEAC using CSL algorithm also obtained 100% of average accuracy in Spiral data set while the other combination algorithms never reached 80% and only EAC using SL and CEAC using CCL achieved also 100% as maximum result. In Half Rings data set, CEAC using CSL obtained the highest average Ci value (99.03%) closely followed by EAC using SL (97.26%). Only CEAC, using both CSL and CCL to produce the consensus partition, obtained maxima values of 100%. CEAC using CSL achieved again the best average (96.63%) and maximum (100%) results for Iris. In this data set, the constrained clustering combination algorithms obtained almost always better average and maxima Ci values than EAC. In Breast Cancer data set ACCCS achieved about 90% of average accuracy for every constraint set but the best average (94.14%) and maximum (97.95%) results were obtained by CEAC using CSL with 200 constraints. The other methods best average result was obtained by EAC using SL with 83.88% of average accuracy. The results for Log Yeast data set were generally poor. The best average and maximum Ci values were achieved again by CEAC using CSL with 44.92% and 56.77% of accuracy, respectively. In the "standardized" version of the same data, the results were a little better. ACCCS achieved the best average results for each constraint set with accuracies superior to 62% and also the maximum Ci value (73.70%). In Optdigits data set, EAC obtained 54.62% and 56.82% average results using, respectively, SL and CL algorithms to produce the consensus partition. These results were outperformed by all constrained clustering combination methods. ACCCS obtained average accuracies superior to 77% with all constraint sets, and the better average and absolute results were achieved by CEAC using CSL with 79.40% and 90.3% of accuracy. In Glass data set, all clustering combination methods obtained average accuracy values between 39% and 47%, with the CEAC using CSL exception that achieved in average 60.07% of accuracy and 76.17% as best result with 200 constraints. In Wine data set, EAC using SL algorithm achieved 70.64% of average accuracy and had generally better performance that the constrained methods. The exception was ACCCS with 200 constraints that obtained 72.92% in average and the highest Ci value (76.97%). Finally, in Image Segmentation data set the constrained clustering combination methods usually outperformed EAC (27.68% and 42.41% of average accuracy using SL and CL, respectively). We highlight again CEAC CSL performance using 200 constraints that achieved in average 66.75% of correctly clustered data patterns, according to P^0, and the the maximum Ci value with 67.49%.

Despite none of the clustering combination methods produced always the best average or maximum results, the CEAC method using CSL algorithm stands out by achieving the best average Ci values in 9 out of the 12 data sets, followed by ACCCS method with 3 best average results. EAC only equaled one best result (in Cigar data set) and the methods that used CL or CCL to produce the consensus partitions never obtained a best average result. It can also be seen that with the increase of the number of constraints the quality of the consensus partitions is improved, specially in CEAC clustering combination method. In ACCCS this relation is not as evident, probably due to $\mathcal{C}_=$ and $\mathcal{C}_{\neq}$ being thought as soft constraints.

7 Conclusions

We proposed an extension to Evidence Accumulation Clustering (CEAC) and a novel algorithm (ACCCS) to solve the cluster ensemble problem using data pattern pairwise constraints in order to improve data clustering quality. The extension to Evidence Accumulation Clustering consists of requiring the clustering algorithm that produces the consensus partition, using pairwise pattern similarities defined in the co-association matrix, to support the incorporation of must-link and cannot-link constraints. The ACCCS approach comprises the maximization of both the similarity between cluster ensemble data partitions and a target consensus partition, and the constraint satisfaction. Experimental results using 4 synthetic and 8 real data sets shown that constrained clustering combination methods usually improve clustering quality.

In this work, we assumed that the constraint sets are noise free. In future work, the proposed constrained clustering combination algorithms should also be tested with noisy constraint sets.

Acknowledgments

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250).

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys **31** (1999) 264–323
2. Fred, A.L.N.: Finding consistent clusters in data partitions. In: MCS '01: Proceedings of the Second International Workshop on Multiple Classifier Systems, London, UK, Springer-Verlag (2001) 309–318
3. Strehl, A., Ghosh, J.: Cluster ensembles — a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3** (2003) 583–617
4. Fred, A.L.N., Jain, A.K.: Combining multiple clusterings using evidence accumulation. IEEE Trans. Pattern Anal. Mach. Intell. **27** (2005) 835–850
5. Duarte, F.J., Fred, A.L.N., Rodrigues, M.F.C., Duarte, J.: Weighted evidence accumulation clustering using subsampling. In: Sixth International Workshop on Pattern Recognition in Information Systems. (2006)
6. Fern, X., Brodley, C.: Solving cluster ensemble problems by bipartite graph partitioning. In: ICML '04: Proceedings of the twenty-first international conference on Machine learning, New York, NY, USA, ACM (2004) 36
7. Topchy, A.P., Jain, A.K., Punch, W.F.: A mixture model for clustering ensembles. In Berry, M.W., Dayal, U., Kamath, C., Skillicorn, D.B., eds.: SDM, SIAM (2004)
8. Jouve, P., Nicoloyannis, N.: A new method for combining partitions, applications for distributed clustering. In: International Workshop on Paralell and Distributed Machine Learning and Data Mining (ECML/PKDD03). (2003) 35–46
9. Basu, S., Davidson, I., Wagstaff, K.: Constrained Clustering: Advances in Algorithms, Theory, and Applications. Chapman & Hall/CRC (2008)

10. Tung, A.K.H., Hou, J., Han, J.: Coe: Clustering with obstacles entities. a preliminary study. In: PADKK '00: Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications, London, UK, Springer-Verlag (2000) 165–168
11. Wagstaff, K.L.: Intelligent clustering with instance-level constraints. PhD thesis, Ithaca, NY, USA (2002) Chair-Claire Cardie.
12. Ge, R., Ester, M., Jin, W., Davidson, I.: Constraint-driven clustering. In: KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2007) 320–329
13. Basu, S.: Semi-supervised clustering: probabilistic models, algorithms and experiments. PhD thesis, Austin, TX, USA (2005) Supervisor-Mooney, Raymond J.
14. Klein, D., Kamvar, S.D., Manning, C.D.: From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In: ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2002) 307–314
15. Davidson, I., Ravi, S.: Clustering with constraints feasibility issues and the k-means algorithm. In: 2005 SIAM International Conference on Data Mining (SDM'05), Newport Beach,CA (2005) 138–149
16. Duarte, F.J.: Optimização da Combinação de Agrupamentos Baseado na Acumulação de Provas Pesadas por índices de Validação e com Uso de Amostragem. PhD thesis, Universidade de Trás-os-Montes e Alto Douro (2008)

[6] - (no technical report)

Duarte, F.J., Duarte, J.M.M., Fred, A.L.N., Rodrigues, M.F.: Average cluster consistency for cluster ensemble selection. In Fred, A., Dietz, J.L.G., Liu, K., Filipe, J., eds.: Knowledge Discovery, Knowlege Engineering and Knowledge Management. Volume 128 of Communications in Computer and Information Science. Springer (2011) 133–148 First International Joint Conference, IC3K 2009, Funchal, Madeira, Portugal, October 6-8, 2009, Revised Selected Papers.

Average Cluster Consistency for Cluster Ensemble Selection

F. Jorge F. Duarte[1], João M. M. Duarte[1,2], Ana L. N. Fred[2], and M. Fátima C. Rodrigues[1]

[1] GECAD - Knowledge Engineering and Decision Support Group,
Instituto Superior de Engenharia do Porto,
R. Dr. António Bernardino de Almeida, 431, P-4200-072 Porto, Portugal,
{fjd,jod,fr}@isep.ipp.pt,
www.gecad.isep.ipp.pt

[2] Instituto de Telecomunicações, Instituto Superior Técnico,
Av. Rovisco Pais, 1, P-1049-001, Lisboa, Portugal
{jduarte,afred}@lx.it.pt
www.it.pt

Abstract. Various approaches to produce cluster ensembles and several consensus functions to combine data partitions have been proposed in order to obtain a more robust partition of the data. However, the existence of many approaches leads to another problem which consists in knowing which of these approaches to produce the cluster ensembles' data and to combine these partitions best fits a given data set. In this paper, we propose a new measure to select the best consensus data partition, among a variety of consensus partitions, based on the concept of average cluster consistency between each data partition that belongs to the cluster ensemble and a given consensus partition. The experimental results obtained by comparing this measure with other measures for cluster ensemble selection in 9 data sets, showed that the partitions selected by our measure generally were of superior quality in comparison with the consensus partitions selected by other measures.

1 Introduction

Data clustering goal consists of partitioning a data set into clusters, based on a concept of similarity between data so that similar data patterns are grouped together, and unlike patterns are separated into different clusters. Several clustering algorithms have been proposed in the literature but none can discover all kinds of cluster structures and shapes.

In order to improve data clustering robustness and quality [1], reuse clustering solutions [2] and cluster data in a distributed way, various cluster ensemble approaches have been proposed based on the idea of combining multiple data clustering results into a more robust and better quality consensus partition. The principal proposals to solve the cluster ensemble problem are based on: co-associations between pairs of patterns [3, 4], mapping the cluster ensemble into

graph [5], hyper-graph [2] or mixture model [6] formulations, and searching for a median partition that summarizes the cluster ensemble [7].

A cluster ensemble can be built by using different clustering algorithms [4], using distinct parameters and/or initializations to the same algorithm [3], sampling the original data set [8] and using different feature sets to produce each individual partition [9].

One can also apply different consensus functions to the same cluster ensemble. These variations in the cluster ensemble problem leads to a question: "*Which cluster ensemble construction method and which consensus function should one select for a given data set?*". This paper addresses the implicit problem in the previous question by selecting the best consensus partition based on the concept of *average cluster consistency* between the consensus partition and the respective cluster ensemble.

The rest of this paper is organized as follows. In section 2, the cluster ensemble problem formulation (subsection 2.1), background work about cluster ensemble selection (subsection 2.2) and the clustering combination methods used in our experiments (subsection 2.3) are presented. Section 3 presents a new approach for cluster ensemble selection, based on the notion of average cluster consistency. The experimental setup used to assess the performance of our proposal is described in section 4 and the respective results are presented in section 5. Finally, the conclusions appear in section 6.

2 Background

2.1 Cluster Ensemble Formulation

Let $\mathcal{X} = \{x_1, \cdots, x_n\}$ be a set of n data patterns and let $P = \{C_1, \cdots, C_K\}$ be a partition of $\mathcal{X}$ into K clusters. A cluster ensemble $\mathcal{P}$ is defined as a set of N data partitions P^l of $\mathcal{X}$:

$$\mathcal{P} = \{P^1, \cdots, P^N\}, P^l = \{C_1^l, \cdots, C_{K^l}^l\}, \tag{1}$$

where C_k^l is the k^{th} cluster in data partition P^l, which contains K^l clusters, and $\sum_{k=1}^{K^l} |C_k^l| = n$, $\forall l \in \{1, \cdots, N\}$.

There are two fundamental phases in combining multiple data partitions: the partition generation mechanism and the consensus function, that is, the method that combines the N data partitions in $\mathcal{P}$. As introduced before, there are several ways to generate a cluster ensemble $\mathcal{P}$, such as, producing partitions of $\mathcal{X}$ using different clustering algorithms, changing parameters and/or initializations for the same clustering algorithm, using different subsets of data features or patterns, projecting $\mathcal{X}$ to subspaces and combinations of these. A consensus function f maps a cluster ensemble $\mathcal{P}$ into a consensus partition P^*, $f : \mathcal{P} \rightarrow P^*$, such that P^* should be robust and consistent with $\mathcal{P}$, i.e., the consensus partition should not change (significantly) when small variations are introduced in the cluster ensemble and the consensus partition should reveal the underlying structure of $\mathcal{P}$.

2.2 Cluster Ensemble Selection

As previously referred, the combination of multiple data partitions can be carried out in various ways, which may lead to very different consensus partitions. This diversity causes the problem of picking the best consensus data partition from all the produced ones.

In [10] work, a study was conducted on the diversity of the cluster ensemble and its relation to the consensus partition quality. Four measures were defined in order to assess the diversity of a cluster ensemble, by comparing each data partition $P^l \in \mathcal{P}$ with the final data partition P^*. The adjusted Rand index [11] was used to assess the agreement between pairs of data clusterings ($Rand(P^l, P^*) \in [0, 1]$). Values close to 1 mean that the clusterings are similar.

The first measure, $Div_1(P^*, \mathcal{P})$, is defined as the average diversity between each clustering $P^l \in \mathcal{P}$ and the consensus partition P^*. The diversity between P^l and P^* is defined as $1 - Rand(P^l, P^*)$. Formally, the average diversity between P^* and $\mathcal{P}$ is defined as:

$$Div_1(P^*, \mathcal{P}) = \frac{1}{N} \sum_{l=1}^{N} 1 - Rand(P^l, P^*). \tag{2}$$

Previous work [12] showed that the cluster ensembles that exhibit higher individual variation of diversity generally obtained better consensus partitions.

The second measure, $Div_2(P^*, \mathcal{P})$, is based on this idea and is defined as the standard deviation of cluster ensemble individual diversity:

$$Div_2(P^*, \mathcal{P}) = \sqrt{\frac{1}{N-1} \sum_{l=1}^{N} \left(1 - Rand\left(P^l, P^*\right) - Div_1\right)^2}, \tag{3}$$

where Div_1 is $Div_1(P^*, \mathcal{P})$.

The third diversity measure, $Div_3(P^*, \mathcal{P})$ is based on the intuition that the consensus partition, P^*, is similar to the *real* structure of the data set. So, if the clusterings $P^l \in \mathcal{P}$ are similar to P^*, i.e., $1 - Div_1$ is close to 1, P^* is expected to be a high quality consensus partition. Nevertheless, as it is assumed that cluster ensembles with high individual diversity variance are likely to produce good consensus partitions, the third measure also includes a component associated to $Div_2(P^*, \mathcal{P})$. It is formally defined as:

$$Div_3(P^*, \mathcal{P}) = \frac{1}{2}(1 - Div_1 + Div_2), \tag{4}$$

where Div_2 corresponds to $Div_2(P^*, \mathcal{P})$.

The forth measure, $Div_4(P^*, \mathcal{P})$, simply consists of a ratio between the standard deviation of the cluster ensemble individual diversity and the average diversity between P^* and $\mathcal{P}$, as shown in equation 5.

$$Div_4(P^*, \mathcal{P}) = \frac{Div_2(P^*, \mathcal{P})}{Div_1(P^*, \mathcal{P})} \tag{5}$$

The four previously referred measures were compared in [10] and the authors concluded that only $Div_1(P^*, \mathcal{P})$ and, specially, $Div_3(P^*, \mathcal{P})$ measures showed some correlation with the quality of the consensus partition. Despite that, in some data sets the quality of the final data partitions increased as $Div_1(P^*, \mathcal{P})$ and $Div_3(P^*, \mathcal{P})$ also increased, in several other data sets it did not occur. The authors recommended that one should select the cluster ensembles with the median values of $Div_1(P^*, \mathcal{P})$ or $Div_3(P^*, \mathcal{P})$ to choose a good consensus partition.

In other work [2], the best consensus partition P^B is thought as the consensus partition P^* that maximizes the Normalized Mutual Information (NMI) between each data partition $P^l \in \mathcal{P}$ and P^*, i.e., $P^B = \arg\max_{P^*} \sum_l^N NMI(P^*, P^l)$. $NMI(P^*, P^l)$ is defined as:

$$NMI(P^*, P^l) = \frac{MI(P^*, P^l)}{\sqrt{H(P^*)H(P^l))}}, \tag{6}$$

where $MI(P^*, P^l)$ is the mutual information between P^* and P^l (eq. 7) and $H(P)$ is the entropy of P (eq. 8). The mutual information between two data partitions, P^* and P^l, is defined as:

$$MI(P^*, P^l) = \sum_i^{K^*} \sum_j^{K^l} \frac{Prob(i,j)}{Prob(i)Prob(j)}, \tag{7}$$

with $Prob(k) = \frac{n_k}{n}$, where n_k is the number of patterns in the k^{th} cluster of P, and $Prob(i,j) = \frac{1}{n}|C_i^* \cap C_j^l|$.

The entropy of a data partition P is given by:

$$H(P) = -\sum_{k=1}^{K} Prob(k) \log Prob(k). \tag{8}$$

Therefore, the Average Normalized Mutual Information ($ANMI(P^*, \mathcal{P})$) between the cluster ensemble and a consensus partition, defined in eq. 9, can be used to select the best consensus partition. Higher values of $ANMI(P^*, \mathcal{P})$ suggest better quality consensus partitions.

$$ANMI(P^*, \mathcal{P}) = \frac{1}{N} \sum_{l=1}^{N} NMI(P^*, P^l). \tag{9}$$

2.3 WEACS

The Weighted Evidence Accumulation Clustering using Subsampling (WEACS) [4] approach is an extension to Evidence Accumulation Clustering (EAC) [1]. EAC considers each data partition $P^l \in \mathcal{P}$ as an independent evidence of data organization. The underlying assumption of EAC is that two patterns belonging to the same *natural* cluster will be frequently grouped together. A vote is given

to a pair of patterns every time they co-occur in the same cluster. Pairwise votes are stored in a $n \times n$ co-association matrix and are normalized by the total number of combining data partitions:

$$co\_assoc_{ij} = \frac{\sum_{l=1}^{N} vote_{ij}^{l}}{N}, \tag{10}$$

where $vote_{ij}^{l} = 1$ if x_i and x_j belong to the same cluster C_k^l in the data partition P^l, otherwise $vote_{ij}^{l} = 0$. In order to produce the consensus partition, one can apply any clustering algorithm over the co-association matrix *co_assoc*.

WEACS extends EAC by weighting each pattern pairwise vote based on the quality of each data partition P^l and by using subsampling in the construction of the cluster ensemble. The idea consists of perturbing the data set and assigning higher relevance to better data partitions in order to produce better combination results. To weight each $vote_{ij}^{l}$ in a weighted co-association matrix, *w_co_assoc*, one or several internal clustering validity indices are used to measure the quality of each data partition P^l, and the corresponding normalized index value, IV^l, corresponds to the weight factor. Note that the internal validity indices assess the clustering results in terms of quantities that involve only the features of the data set, so no *a priori* information is provided. Formally, *w_co_assoc* is defined as

$$w\_co\_assoc_{ij} = \frac{\sum_{l=1}^{N} IV^l \times vote_{ij}^{l}}{S_{ij}}, \tag{11}$$

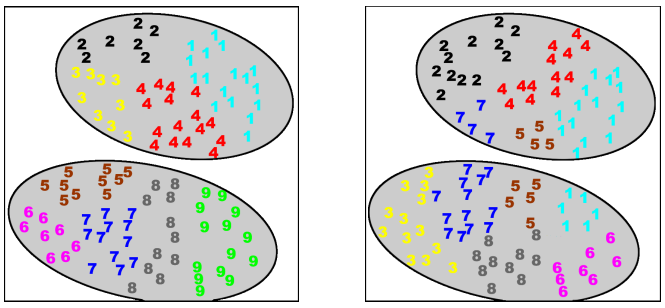
where S is a $n \times n$ matrix with S_{ij} equal to the number of data partitions where both x_i and x_j are simultaneously selected to belong to the same data subsample.

There are two versions of WEACS that correspond to two different ways for computing the weight factor IV^l. The first one, Single WEACS (SWEACS), uses the result of only one clustering validity index to assess the quality of P^l, i.e., $IV^l = norm\_validity(P^l)$, where $norm\_validity(\cdot)$ corresponds to a normalized validity index function that returns a value in the interval $[0, 1]$. Higher values correspond to better data partitions. In the second version, Joint WEACS (JWEACS), IV^l is defined as the average of the output values of $NumInd$ normalized validity index functions, $norm\_validity_m(\cdot)$, applied to P^l, i.e., $IV^l = \sum_{m=1}^{NumInd} norm\_validity_m(P^l)/NumInd$.

In the WEACS approach, one can use different cluster ensemble construction methods, different clustering algorithms to obtain the consensus partition, and, particularly in the SWEACS version, one can even use different cluster validity indices to weight pattern pairwise votes. These constitute variations of the approach, taking each of the possible modifications as a configuration parameter of the method. As shown in section 4, although the WEACS leads in general to good results, no individual tested configuration led consistently to the best result in all data sets. We used a complementary step to the WEACS approach which consists of combining all the final data partitions obtained in the WEACS approach within a cluster ensemble construction method using EAC. The interested reader is encouraged to read [4] for a detailed description of WEACS.

3 Average Cluster Consistency (ACC)

The idea behind Average Cluster Consistency (ACC) measure [13] is that if the similarity between the multiple data partitions in the cluster ensemble and the consensus partition is high, the quality of the consensus partition will also be high. Some clustering combination methods, such as the EAC and WEACS methods presented in subsection 2.3, usually produce better quality consensus data partitions when combining data partitions with more clusters than the expected *real* number of clusters K^0. This difference in the number of clusters usually leads to low similarity scores when comparing two data partitions. For this reason, a new concept for comparing data partitions was defined. In this new similarity measure between two data partitions, P^l and P^0 with $K^l >> K^0$, if each of the K^l clusters $C_k^l \in P^l$ is a subset of a cluster $C_m^0 \in P^0$, i.e. $C_k^l \subseteqq C_m^0$,then the partitions P^l and P^0 have the maximum degree of similarity. If the data patterns belonging to each cluster in P^l are split into different clusters in P^0, the data partitions P^l and P^0 are dissimilar. Figure 1 shows an example of the previously described situations. The figure includes two consensus partitions (one in figure 1 (a) and another in figure 1 (b)) each with $K^0 = 2$ clusters (shaded areas). Inside each consensus partition's clusters, there are several patterns represented by numbers, which indicate the cluster labels assigned to the data patterns in a partition P^l belonging to the cluster ensemble. Note that the number of clusters of the partition P^l is higher than the number of clusters of the consensus partition P^0 ($K^l >> K^0$). On the left figure, a perfect similarity between P^0 and P^l is presented as all data patters of each cluster C_k^l belong to the same cluster in P^0. On the right figure, two dissimilar partitions are presented as the data patterns belonging to clusters 1, 5 and 7 in P^l are divided in the two clusters of P^0.



(a) *similar* partitions (b) *dissimilar* partitions

Fig. 1: Example of Average Cluster Consistency motivation.

Our similarity measure between two partitions, P^* and P^l, is then defined as

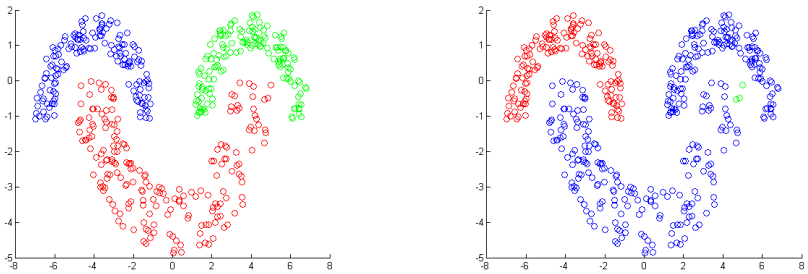
$$sim(P^*, P^l) = \frac{\sum_{m=1}^{K^l} \max_{1 \leq k \leq K^*} |Inters_{km}|(1 - \frac{|C_k^*|}{n})}{n}, \quad (12)$$

where $K^l \geq K^*$, $|Inters_{km}|$ is the cardinality of the set of patterns common to the k^{th} and m^{th} clusters of P^* and P^l, respectively ($Inters_{km} = \{x_a | x_a \in C_k^* \wedge x_a \in C_m^l\}$). Note that in Eq. 12, $|Inters_{km}|$ is weighted by $(1 - \frac{|C_k^*|}{n})$ in order to prevent cases where P^* has clusters with almost all data patterns and would obtain a high value of similarity.

The Average Cluster Consistency measures the average similarity between each data partition in the cluster ensemble ($P^l \in \mathcal{P}$) and a target consensus partition P^*, using the previously explained notion of similarity. It is formally defined by

$$ACC(P^*, \mathcal{P}) = \frac{\sum_{i=1}^{N} sim(P^i, P^*)}{N}. \quad (13)$$

From a set of possible choices, the *best* consensus partition is the one that achieves the highest $ACC(P^*, \mathcal{P})$ value. Note that by the fact of using subsampling, the ACC measure only uses the data patterns of the consensus partition P^* that appear in the combining data partition $P^l \in \mathcal{P}$.



(a) high quality partition (b) low quality partition

Fig. 2: Two data partitions of the Half Rings data set.

In order to justify the use of the weighting factor $\left(1 - \frac{|C_k^*|}{n}\right)$ in our similarity measure between two data partitions (equation 12) used in the ACC measure (equation 13), we present the example shown in figure 2. This figure shows two consensus partitions of a synthetic data set used in our experiments, the Half Rings data set (presented in section 4). Both consensus partitions have 3 clusters and were obtained using two different clustering algorithms (Single-Link and K-means) to extract the consensus partition in the WEACS approach.

The first consensus partition is perfect since it correctly identifies the three existing groups in the data set, while the second consensus partition is of poor

quality because it contains a large cluster (represented in blue) that almost encompasses two real clusters of the data set.

Table 1: ACC values obtained for the data partitions shown in figure 2 with and without the use of the weighting factor.

Data set partitions	Partition a	Partition b
ACC not using the weighting factor	1.0000	1.0000
ACC using the weighting factor	0.6595	0.4374

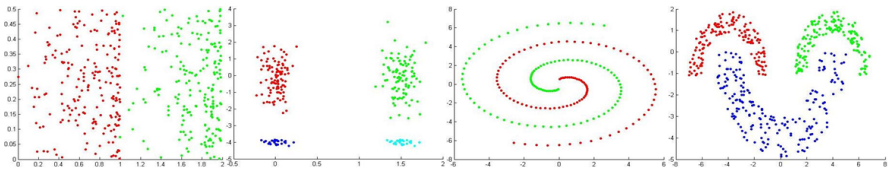
Table 1 shows the values obtained by ACC measure without (second line) and with (third line) the use of the weighting factor for both data partitions. The ACC measure without the use of the weighting factor obtained the value 1 for both data partitions, while the ACC measure using the weighting factor obtained the value 0.6595 for the "optimal" partition (figure 2 a) and 0.4374 for the other partition (figure 2 b). As can be seen by this example, the use of the weighting factor in our similarity measure between two data partition (equation 12) prevents cases where the consensus partitions have clusters with almost all data patterns and would obtain a high value of similarity.

At the first glance, this measure may seem to contradict the observations by [10] and [12] which point out that the clustering quality is improved with the increase of diversity in the cluster ensemble. However, imagine that each data partition belonging to a cluster ensemble is obtained by random guess. The resulting cluster ensemble is very diverse but does not provide useful information about the structure of the data set, so, it is expected to produce a low quality consensus partition. For this reason, one should distinguish the "good" diversity from the "bad" diversity. Our definition of similarity between data partitions (Eq. 12) considers that two apparently different data partitions (for instance, partitions with different number of clusters) may be similar if they have a common structure, as shown in the figure 1 (a) example, and the outcome is the selection of cluster ensembles with "good" diversity rather than the ones with "bad" diversity.

4 Experimental Setup

We used 4 synthetic and 5 real data sets to assess the quality of the cluster ensemble methods on a wide variety of situations, such as data sets with different cardinality and dimensionality, arbitrary shaped clusters, well separated and touching clusters and distinct cluster densities. A brief description for each data set is given below.

Synthetic Data Sets. Fig. 3 presents the 2-dimensional synthetic data sets used in our experiments. Bars data set is composed by two clusters very close together, each with 200 patterns, with increasingly density from left to right. Cigar data set consists of four clusters, two of them having 100 patterns each



(a) Bars (b) Cigar (c) Spiral (d) Half Rings

Fig. 3: Synthetic data sets.

and the other two groups 25 patterns each. Spiral data set contains two spiral shaped clusters with 100 data patterns each. Half Rings data set is composed by three clusters, two of them have 150 patterns and the third one 200.

Real Data Sets. The 5 real data sets used in our experiments are available at UCI repository (http://mlearn.ics.uci.edu/MLRepository.html). The first one is Iris and consists of 50 patterns from each of three species of Iris flowers (setosa, virginica and versicolor) characterized by four features. One of the clusters is well separated from the other two overlapping clusters. Breast Cancer data set is composed of 683 data patterns characterized by nine features and divided into two clusters: benign and malignant. Yeast Cell data set consists of 384 patterns described by 17 attributes, split into five clusters concerning five phases of the cell cycle. There are two versions of this dataset, the first one is called Log Yeast and uses the logarithm of the expression level and the other is called Std Yeast and is a "standardized" version of the same data set, with mean 0 and variance 1. Finally, Optdigits is a subset of Handwritten Digits data set containing only the first 100 objects of each digit, from a total of 3823 data patterns characterized by 64 attributes.

In order to produce the cluster ensembles, we applied the Single-Link (SL) [14], Average-Link (AL) [14], Complete-Link (CL) [15], K-means (KM) [16], CLARANS (CLR) [17], Chameleon (CHM) [18], CLIQUE [19], CURE [20], DBSCAN [21] and STING [22] clustering algorithms to each data set to generate 50 cluster ensembles for each clustering algorithm. Each cluster ensemble has 100 data partitions with the number of clusters, K, randomly chosen in the set $K \in \{10, \cdots, 30\}$.

After all cluster ensembles have been produced, we applied the EAC, SWEACS and JWEACS approaches using the KM, SL, AL and Ward-Link (WR) [23] clustering algorithms to produce the consensus partitions. The number of clusters of the combined data partitions were set to be the *real* number of clusters of each data set. We also defined other two cluster ensembles: ALL5 and ALL10. The cluster ensemble referred as ALL5 is composed by the data partitions of SL, AL, CL, KM and CLR algorithms ($N = 500$) and the cluster ensemble ALL10 is composed by the data partitions produced by all data clustering algorithms ($N = 1000$).

To evaluate the quality of the consensus partitions we used the Consistency index (Ci) [1]. Ci measures the fraction of shared data patterns in matching clusters of the consensus partition (P^*) and of the *real* data partition (P^0). Formally, the Consistency index is defined as

$$Ci(P^*, P^0) = \frac{1}{n} \sum_{k=1}^{\min\{K^*, K^0\}} |C_k^* \cap C_k^0| \tag{14}$$

where $|C_k^* \cap C_k^0|$ is the cardinality of the P^* and P^0 k^{th} matching clusters data patterns intersection.

As an example, table 2 shows the results of the cluster combination approaches for the Optdigits data set, averaged over the 50 runs. In this table, rows are grouped by cluster ensemble construction method. Inside each cluster ensemble construction method appears the 4 clustering algorithms used to extract the final data partition (KM, SL, CL and WR). The last column (C. Step) shows the results of the complementary step of WEACS. As it can be seen, the results vary from a very poor result obtained by SWEACS, combining data partitions produced by SL algorithm and using the K-means algorithm to extract the consensus partitions (10% of accuracy), to good results obtained by all clustering combination approaches, when combining data partitions produced by CHM and using the WR algorithm to extract the consensus partition. For this configuration, EAC achieved 87.54% of accuracy, JWEAC 87.74%, SWEAC 87.91% using PS validity index to weight each vote in w_co_assoc, and 88.03% using the complementary step. Due to space restrictions and to the fact that this is not the main topic of this paper, we do not present the results for the other data sets used in our experiments.

Table 3 shows the average and best $C_i(P^*, P^0)$ percentage values obtained by each clustering combination method for each data set. We present this table to remark that the average quality of the consensus partitions produced by each clustering combination method is substantially different from the best ones. As an example, SWEACS approach achieved 90.89% as the best result for Std Yeast data set while the average accuracy was only of 54.00%.

The results presented in tables 2 and 3 show that different cluster ensemble construction methods and consensus functions can produce consensus partitions with very different quality. This reason emphasizes the importance of selecting the best consensus partition from a variety of possible consensus data partitions.

5 Results

In order to assess the quality of Average Cluster Consistency (ACC) measure (Eq. 13), we compared its performance against three other measures: the Average Normalized Mutual Information (ANMI) measure (Eq. 9), the Div_1 measure (Eq. 2) and the Div_3 measure (Eq. 4). For each data set, the four measures were calculated for each consensus clustering produced by the clustering combination

Table 2: Average $C_i(P^*, P^0)$ percentage values obtained by EAC, JWEACS and SWEACS for Optdigits data set.

CE	Ext.Alg.	EAC	JWEAC	HubN	Dunn	S_Dbw	CH	S	I	XB	DB	SD	PS	C.Step
SL	KM	39.75	34.47	36.89	36.66	38.14	35.29	10.00	39.16	38.03	33.84	42.09	33.55	34.19
	SL	10.60	10.60	10.60	10.60	10.60	10.60	10.10	10.60	10.60	10.60	10.60	10.60	11.19
	AL	10.60	10.60	10.60	10.60	10.60	10.60	10.10	10.60	10.60	10.60	10.60	10.60	20.21
	WR	40.31	40.31	40.53	40.30	40.40	40.31	10.10	40.30	40.31	40.40	40.49	40.31	44.28
AL	KM	70.33	69.84	71.09	68.83	70.40	71.47	70.42	72.19	69.59	67.68	69.49	68.83	73.93
	SL	60.14	60.21	60.14	60.14	51.48	60.37	60.14	60.37	60.14	60.14	60.14	60.14	67.65
	AL	67.29	67.28	67.29	67.29	67.29	67.30	67.29	69.42	67.28	67.29	67.29	67.29	67.28
	WR	82.10	82.06	82.10	82.10	83.57	84.31	82.10	84.31	82.10	82.10	82.10	82.09	84.32
CL	KM	62.77	62.39	64.20	63.05	62.28	64.97	64.82	66.30	62.97	63.78	68.95	62.92	64.25
	SL	53.76	52.54	53.80	53.80	53.80	58.45	58.57	58.25	52.72	53.80	52.47	52.52	58.15
	AL	69.28	70.97	70.94	70.94	69.28	70.89	71.21	63.50	69.28	70.94	70.94	70.94	70.53
	WR	76.27	76.34	76.35	76.27	76.27	71.16	76.35	71.14	76.34	76.26	76.35	76.35	71.25
KM	KM	68.77	69.43	72.56	69.97	73.75	73.43	69.52	70.94	69.57	69.29	71.81	74.39	67.86
	SL	30.59	30.60	30.21	30.60	30.78	30.21	30.78	30.69	30.78	30.60	30.60	30.60	59.50
	AL	79.78	79.43	79.42	79.51	79.32	77.49	79.41	77.54	79.41	79.78	79.41	79.60	79.35
	WR	79.51	79.67	79.49	79.85	79.71	77.11	78.85	77.00	78.74	78.97	78.87	79.75	78.05
CLR	KM	63.96	63.61	65.60	65.24	65.39	67.14	64.58	65.13	62.32	65.69	62.28	65.38	62.81
	SL	20.31	20.11	20.31	20.51	20.51	19.81	20.31	19.81	20.40	20.31	20.31	20.31	42.67
	AL	82.73	82.37	82.24	82.78	82.48	75.53	81.11	75.32	82.60	82.21	82.85	79.34	76.15
	WR	78.85	78.66	79.27	79.25	77.54	78.58	79.37	78.81	79.06	78.86	77.12	79.27	77.37
ALL5	KM	71.49	69.85	69.52	69.93	69.43	71.31	69.67	70.70	75.98	70.57	69.11	67.77	64.77
	SL	39.50	30.30	49.24	30.30	20.81	40.40	49.83	40.39	30.39	20.60	30.30	30.30	51.23
	AL	65.57	65.22	73.21	51.24	30.50	71.14	80.44	65.62	60.11	30.41	30.60	30.79	65.32
	WR	80.86	80.88	80.51	80.89	80.76	80.95	80.54	80.98	80.53	80.31	80.69	80.51	80.85
CHM	KM	71.97	72.12	73.11	71.40	73.74	72.17	72.69	72.77	73.20	70.48	72.26	73.10	68.74
	SL	62.44	62.24	62.06	62.43	62.62	62.63	62.63	61.66	62.61	62.44	62.24	62.24	78.34
	AL	87.14	86.88	86.53	87.28	86.46	87.28	87.31	86.76	86.26	86.75	86.82	86.50	84.78
	WR	**87.54**	**87.74**	87.61	87.51	87.53	87.78	87.52	87.72	87.56	87.68	87.76	**87.91**	**88.03**
CLIQUE	KM	59.41	60.29	61.33	59.84	59.95	60.69	63.27	61.28	61.90	60.50	60.41	60.30	64.19
	SL	10.50	10.47	10.50	10.48	10.48	10.50	10.47	10.49	10.50	10.48	10.48	10.50	18.76
	AL	61.03	63.30	64.89	62.20	62.13	63.67	65.71	64.12	66.02	63.65	63.29	64.54	62.85
	WR	67.00	68.23	69.11	67.65	67.68	68.77	73.19	71.02	71.36	69.30	68.67	69.03	70.69
CURE	KM	58.84	57.03	62.75	58.15	45.17	66.12	23.81	51.28	50.60	55.22	52.17	46.88	63.06
	SL	10.63	10.63	10.63	10.63	10.62	10.62	16.61	10.64	10.63	10.63	10.63	10.63	11.00
	AL	10.60	10.60	10.58	10.60	10.61	10.63	18.39	10.61	10.60	10.61	10.61	10.60	26.81
	WR	67.09	67.04	75.55	68.00	62.29	77.48	26.16	71.46	63.41	65.81	63.82	63.56	71.25
DBSCAN	KM	68.81	69.61	70.18	67.85	66.97	69.71	68.68	68.51	69.42	69.04	69.51	70.00	71.10
	SL	62.87	62.56	63.01	63.15	62.72	64.40	62.52	65.09	63.88	63.16	62.86	63.20	75.86
	AL	77.21	77.16	77.07	77.11	76.76	76.90	77.16	77.25	76.69	77.20	76.85	76.88	77.32
	WR	80.98	79.84	80.02	80.36	81.06	79.13	80.78	78.82	78.83	80.61	79.96	79.36	81.19
STING	KM	60.60	59.77	59.00	59.49	60.27	60.09	58.60	59.01	58.70	59.17	59.47	58.55	62.07
	SL	22.03	22.03	22.17	22.05	21.99	22.59	19.59	23.71	22.50	22.01	22.01	22.02	34.97
	AL	37.89	38.01	37.86	38.07	36.32	39.97	46.09	42.06	37.97	36.72	37.60	37.60	48.40
	WR	57.65	57.74	57.90	57.60	57.66	57.69	66.12	57.77	57.72	57.64	57.70	57.63	58.35
ALL10	KM	72.36	72.05	72.50	72.64	72.04	71.40	72.33	72.36	72.62	73.39	72.96	73.67	66.39
	SL	42.66	38.14	53.57	32.91	20.63	55.39	55.24	49.65	30.82	20.47	30.20	30.21	59.59
	AL	74.22	70.63	74.95	61.66	22.04	76.03	83.09	75.23	62.20	30.59	30.23	31.40	73.58
	WR	83.24	83.87	83.65	83.80	83.83	83.14	83.78	82.89	84.14	83.54	84.19	83.69	83.10

Table 3: Average and best $C_i(P^*, P^0)$ percentage values obtained by EAC, JWEACS and SWEACS for all data sets.

Approach		Bars	Breast	Cigar	Half Rings	Iris	Log Yeast	Optical	Std Yeast	Spiral
EAC	Average	86.80	80.96	85.57	84.13	73.88	34.14	58.33	53.23	67.22
	Best	99.50	97.07	100.00	100.00	97.37	40.93	87.54	88.50	100.00
SWEACS	Average	84.65	80.58	84.23	83.10	74.30	33.97	57.25	54.00	65.83
	Best	99.50	97.08	100.00	100.00	97.19	41.57	87.74	90.89	100.00
JWEACS	Average	86.98	80.38	84.66	83.96	74.59	34.16	57.83	53.80	66.57
	Best	99.50	97.20	100.00	100.00	97.29	41.58	87.91	92.64	100.00

methods. These values were ploted (figures 4-12) against the respective clustering quality values of each consensus partition ($C_i(P^*, P^0)$). Dots represent the consensus partitions, their positions in the horizontal axis represent the obtained values for the cluster ensemble selection measures and the corresponding positions in the vertical axis indicate the C_i values. The lines shown in the plots were obtained by polynomial interpolation of degree 2.

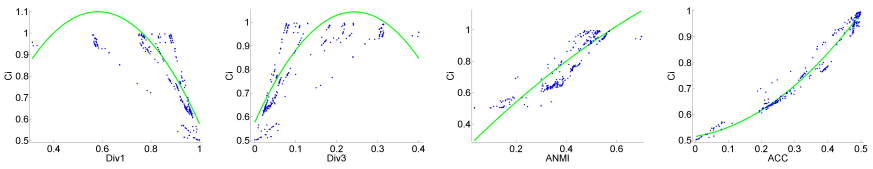


Fig. 4: Ci vs each cluster ensemble selection measures for Bars data set.

Figure 4 presents the results obtained by the cluster ensemble selection measures for Bars data set. Div_1 values decrease with the increment of the quality of the consensus partitions, while the values of Div_3 increase as the quality of the consensus partitions is improved. However, the correlations between Div_1 with C_i and Div_3 with C_i are not clearly evident. In the ANMI and ACC plots, one can easily see that as the values of these measures increase, the quality of the consensus partitions are improved.

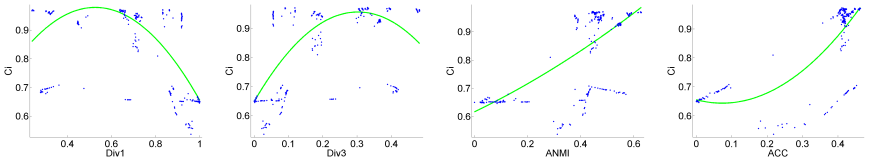


Fig. 5: Ci vs each cluster ensemble selection measures for Breast Cancer data set.

The results achieved for Breast Cancer data set are shown in figure 5. It can be seen that Div_1 and Div_3 measures are not correlated with the quality (C_i values) of the consensus partitions. However, in ANMI and ACC cluster ensemble selection measures there is a tendency of quality improvement as the values of these measures augment.

In the results obtained for Cigar data set, all the four measures showed some correlation with the Consistency index values (figure 6). For Div_1 measure, the quality of the consensus partitions are improved as Div_1 values decrease.

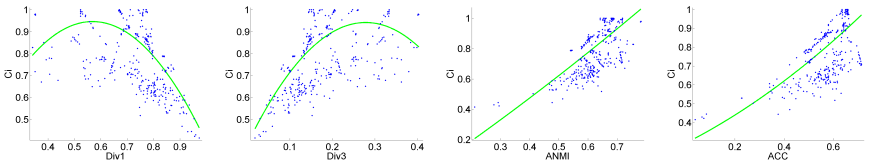


Fig. 6: Ci vs each cluster ensemble selection measures for Cigar data set.

For the remaining measures, the increasing of their values are followed by the improvement of the consensus partitions. Note that the dispersion of the dots in Div_1 and Div_3 plots are clearly higher than the dispersion presented in ANMI and ACC plots, showing that the correlations with C_i of the latter two measures are much stronger.

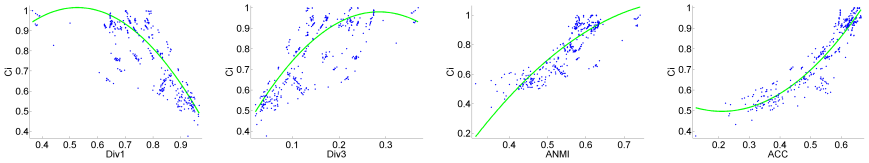


Fig. 7: Ci vs each cluster ensemble selection measures for Half Rings data set.

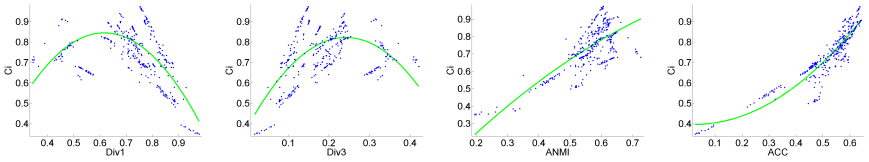


Fig. 8: Ci vs each cluster ensemble selection measures for Iris data set.

Figures 7 and 8 present the plots obtained for the selection of the best consensus partition for Half Rings and Iris data sets. The behavior of the measures are similar in both data sets and they are all correlated with the quality of the consensus partition. Again, one can see that as the values of Div_3, ANMI and ACC measures increase, the quality of the consensus partition is improved, while there is an inverse tendency for Div_1 measure. In both data sets, the ACC measure is the one that better correlates its values with C_i as it is the one with the lowest dispersion of the dots in the plot.

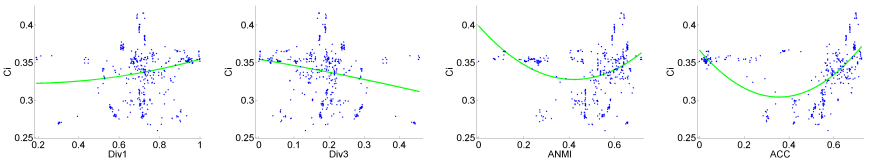


Fig. 9: Ci vs each cluster ensemble selection measures for Log Yeast data set.

The results for the Log Yeast data set are presented in figure 9. The Div_1 and Div_3 measures show no correlations with the quality of the consensus partitions. The ANMI and ACC measures also do not show a clear correlation with C_i. However, in both plots, one can see a cloud of dots that indicates some correlation between the measures and the Consistency index, specially in the ACC plot.

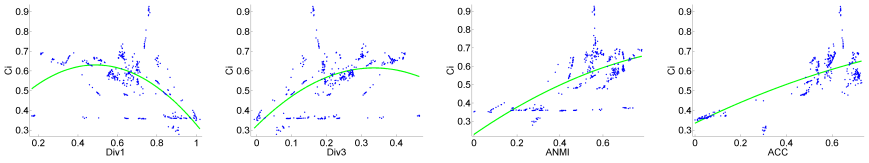


Fig. 10: Ci vs each cluster ensemble selection measures for Std Yeast data set.

In figure 10, the results of the cluster ensemble selection methods for Std Yeast data set are presented. Once again, there is no clear correlation between Div_1 and Div_3 measures and the C_i values. The ANMI and ACC measures also do not present such correlation. However, there is a weak tendency of clustering quality improvement as these measures values increase.

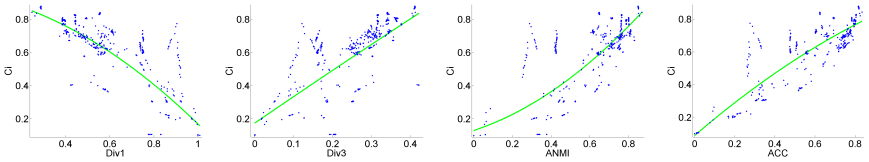


Fig. 11: Ci vs each cluster ensemble selection measures for Optdigits data set.

In the Optdigits data set, all measures are correlated with the quality of the consensus partitions. This correlation is stronger in ACC measure, as it can

be seen in figure 11. The values of Div_1 decrease as the clustering quality is improved while the quality of the consensus partitions is improved as the values of Div_3, ANMI and ACC measures increase.

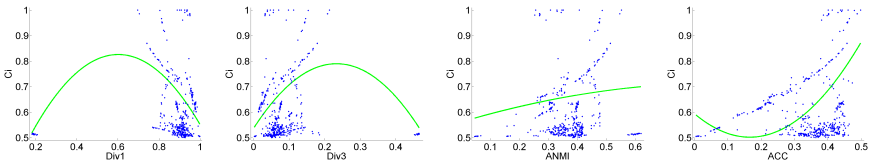


Fig. 12: Ci vs each cluster ensemble selection measure for Spiral data set.

The plots for the last data set, Spiral, are presented in figure 12. The Div_1 and Div_3 measures do not present correlation with C_i values, while the ANMI and ACC measures show weak tendencies of clustering improvement with the increasing of their values, specially in ACC cluster ensemble selection measure.

Table 4 shows the correlation coefficients between the Consistency index and the consensus partition selection measures. Values close to 1 (-1) suggest that there is a positive (negative) linear relationship between Ci and the selection measure, while values close to 0 indicate that there is no such linear relationship. In 6 out of the 9 data sets used in the experiments, the ACC measure obtained the highest linear relationship with the clustering quality (measured using the Consistency index). In the other 3 data sets, the highest linear relationships were obtained by the ANMI measure in the Bars (0.8635 against 0.8480 achieved by ACC) and Cigar (0.6293 against 0.6154 achieved by ACC) data sets, and by the Div_3 measure in the Log Yeast data set which achieved -0.2820, a counterintuitive correlation coefficient when observing the positive coefficients obtained by Div_3 for all the other data sets. In average, the ACC measure presents the highest linear relationship with Ci (0.6928), followed by the ANMI (0.5980), Div_3 (0.4082) and Div_1 (-0.3979) measures.

Table 4: Correlation coefficients between the Consistency index (Ci) and the consensus partition selection measures (Div_1, Div_3, ANMI and ACC measures) for each data set.

Measure	Bars	Breast C.	Cigar	Half Rings	Iris	Log Yeast	Std Yeast	Optdigits	Spiral	Average
Div1	-0.5712	-0.6006	-0.3855	-0.6444	-0.3010	0.2448	-0.5356	-0.7922	0.0044	-0.3979
Div3	0.6266	0.6487	0.4367	0.6838	0.2578	**-0.2820**	0.5450	0.7123	0.0450	0.4082
ANMI	**0.8635**	0.7979	**0.6293**	0.8480	0.6856	-0.0444	0.7141	0.7785	0.1095	0.5980
ACC	0.8480	**0.8684**	0.6154	**0.9308**	**0.8785**	-0.0897	**0.8505**	**0.9149**	**0.4187**	**0.6928**

Table 5 presents the Consistency index values achieved by the consensus partitions selected by the cluster ensemble selection measures (Div_1, Div_3, ANMI and ACC) for each data set, the maximum C_i value of all the produced consensus

partitions and the average C_i values for each best consensus partition selection measure. The consensus partitions for Div_1 and Div_3 measures were selected choosing the consensus partition corresponding to the median of their values, as mentioned in [10]. For the ANMI and ACC measures, the best consensus partition was selected to be the one that maximizes the respective measures.

The quality of the consensus partitions selected by ACC measure was in 6 out of 9 data sets superior or equal to the quality of the consensus partitions selected by the other measures, specifically, in Bars (99.50%), Breast Cancer (97.07%), Iris (90.67%), Log Yeast (35.61%), Optdigits (84.31%) and Spiral (100%) data sets. In Cigar data set, the best consensus partition was selected using Div_3 measure (100%), and the same happened in Half Rings data set together with ANMI. In Std Yeast data set, none of the four measures selected a consensus partition with similar quality to the best produced consensus partition (92.64%). The closed selected consensus partition was selected using ANMI (69.09%). Concerning the average quality of the partitions chosen by the four measures, the ACC measure stands out again, achieving 80.81% of accuracy, followed by ANMI with 77.67%. The Div_3 and Div_1 measures obtained the worst performance with 74.54% and 73.35%, respectively.

Table 5: Ci values for the consensus partition selected by Div_1, Div_3, ANMI and ACC measures, and the maximum C_i value obtained, for each data set.

Measure	Bars	Breast C.	Cigar	Half Rings	Iris	Log Yeast	Std Yeast	Optdigits	Spiral	Average
Div1	95.47	95.11	97.93	99.90	87.35	26.96	57.97	58.55	51.68	74.54
Div3	**99.50**	95.38	**100.0**	**100.0**	85.12	29.92	67.66	30.60	51.94	73.35
ANMI	95.75	96.92	97.85	**100.0**	68.04	35.42	**69.09**	**84.31**	51.63	77.67
ACC	**99.50**	**97.07**	70.97	95.20	**90.67**	**35.61**	53.99	**84.31**	**100.0**	**80.81**
Max Ci	99.50	97.20	100.0	100.0	97.37	41.57	92.64	88.03	100.0	90.70

6 Conclusions

With the aim of combining multiple data partitions into a better consensus partition, several approaches to produce the cluster ensemble and several consensus functions have been developed. With this diversity, very different consensus partitions with very dissimilar qualities can be obtained. This diversity of consensus partitions was exemplified using the Evidence Accumulation Clustering and the Weighted Evidence Accumulation Clustering using Subsampling combination approaches. This paper deals with the question of choose the best consensus partition from a set of consensus partitions, that best fits a given data set. With this purpose, we proposed the Average Cluster Consistency (ACC) measure, based on a new similarity conception between each data partition belonging to the cluster ensemble and a given consensus partition. We compared the performance of the proposed measure with three other measures for cluster ensemble selection, using 9 data sets with arbitrary shaped clusters, well separated and touching clusters, and different cardinality, dimensionality and cluster densities.

The experimental results showed that the consensus partitions selected by ACC measure, usually were of better quality in comparison with the consensus partitions selected by other measures used in our experiments. Therefore, we can say that our approach is a good option for selecting a high quality consensus partition from a set of consensus partitions.

Acknowledgments

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). This work was partially supported by the Portuguese Foundation for Science and Technology (FCT), Portuguese Ministry of Science and Technology, under grant PTDC/EIACCO/103230/2008.

References

1. Fred, A.L.N.: Finding consistent clusters in data partitions. In: MCS '01: Proceedings of the Second International Workshop on Multiple Classifier Systems, London, UK, Springer-Verlag (2001) 309–318
2. Strehl, A., Ghosh, J.: Cluster ensembles — a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3** (2003) 583–617
3. Fred, A.L.N., Jain, A.K.: Combining multiple clusterings using evidence accumulation. IEEE Trans. Pattern Anal. Mach. Intell. **27**(6) (2005) 835–850
4. Duarte, F.J., Fred, A.L.N., Rodrigues, M.F.C., Duarte, J.: Weighted evidence accumulation clustering using subsampling. In: Sixth International Workshop on Pattern Recognition in Information Systems. (2006)
5. Fern, X., Brodley, C.: Solving cluster ensemble problems by bipartite graph partitioning. In: ICML '04: Proceedings of the twenty-first international conference on Machine learning, New York, NY, USA, ACM (2004) 36
6. Topchy, A.P., Jain, A.K., Punch, W.F.: A mixture model for clustering ensembles. In Berry, M.W., Dayal, U., Kamath, C., Skillicorn, D.B., eds.: SDM, SIAM (2004)
7. Jouve, P., Nicoloyannis, N.: A new method for combining partitions, applications for distributed clustering. In: International Workshop on Paralell and Distributed Machine Learning and Data Mining (ECML/PKDD03). (2003) 35–46
8. Topchy, A., Minaei-Bidgoli, B., Jain, A.K., Punch, W.F.: Adaptive clustering ensembles. In: ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1, Washington, DC, USA, IEEE Computer Society (2004) 272–275
9. Topchy, A., Jain, A.K., Punch, W.: Combining multiple weak clusterings. (2003) 331–338
10. Hadjitodorov, S.T., Kuncheva, L.I., Todorova, L.P.: Moderate diversity for better cluster ensembles. Inf. Fusion **7**(3) (2006) 264–275
11. Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification (1985)
12. Kuncheva, L., Hadjitodorov, S.: Using diversity in cluster ensembles. Volume 2. (Oct. 2004) 1214–1219 vol.2
13. Duarte, F., Duarte, J., Fred, A., Rodrigues, F.: Cluster ensemble selection - using average cluster consistency. In: International Conference on Discovery and Information Retrieval (KDIR 2009), Funchal (6-8 October 2009) 85–95

14. Sneath, P., Sokal, R.: Numerical taxonomy (1973) Freeman, London, UK.
15. King, B.: Step-wise clustering procedures. Journal of the American Statistical Association (69) (1973) 86–101
16. Macqueen, J.B.: Some methods of classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathemtical Statistics and Probability. (1967) 281–297
17. Ng, R.T., Han, J.: Clarans: A method for clustering objects for spatial data mining. IEEE Trans. on Knowl. and Data Eng. **14**(5) (2002) 1003–1016
18. Karypis, G., Eui, News, V.K.: Chameleon: Hierarchical clustering using dynamic modeling. Computer **32**(8) (1999) 68–75
19. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD Rec. **27**(2) (1998) 94–105
20. Guha, S., Rastogi, R., Shim, K.: Cure: an efficient clustering algorithm for large databases. In: SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data, New York, NY, USA, ACM (1998) 73–84
21. Ester, M., Kriegel, H.P., Jörg, S., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise (1996)
22. Wang, W., Yang, J., Muntz, R.R.: Sting: A statistical information grid approach to spatial data mining. In: VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1997) 186–195
23. Ward, J.H.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association **58**(301) (1963) 236–244

[7] - (SIMBAD Technical Report n. 2010_27)

Duarte, J., Fred, A., Lourenço, A., Duarte, F.: On consensus clustering validation. In Hancock, E., Wilson, R., Windeatt, T., Ulusoy, I., Escolano, F., eds.: Structural, Syntactic, and Statistical Pattern Recognition. Volume 6218 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2010) 385–394 10.1007/978-3-642-14980-1_37.

On Consensus Clustering Validation

João M. M. Duarte[1,2], Ana L. N. Fred[1], André Lourenço[1], and F. Jorge F. Duarte[2]

[1] Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa, Portugal
{jduarte, arlourenco, afred}@lx.it.pt
[2] GECAD - Knowledge Engineering and Decision Support Group,
Instituto Superior de Engenharia do Porto, Porto, Portugal
{jmmd, fjd}@isep.ipp.pt

Abstract. Work on clustering combination has shown that clustering combination methods typically outperform single runs of clustering algorithms. While there is much work reported in the literature on validating data partitions produced by the traditional clustering algorithms, little has been done in order to validate data partitions produced by clustering combination methods. We propose to assess the quality of a consensus partition using a pattern pairwise similarity induced from the set of data partitions that constitutes the clustering ensemble. A new validity index based on the likelihood of the data set given a data partition, and three modified versions of well-known clustering validity indices are proposed. The validity measures on the original, clustering ensemble, and similarity spaces are analysed and compared based on experimental results on several synthetic and real data sets.

1 Introduction

Clustering ensemble approaches have been proposed aiming to improve data clustering robustness and quality [1], reuse clustering solutions [2], and cluster data in a distributed way. Schematically, these methods can be split into two main phases: the construction of the clustering ensemble (CE); and the combination of information extracted from the CE into a consensus partition. The Evidence Accumulation Clustering method (EAC) [1] additionally produces, as an intermediate result, a learned pairwise similarity between patterns, summarized in a co-association matrix. In the literature on this topic, one can find many alternative ways of building the clustering ensemble, defining the combination strategy and extraction algorithm, and choosing the final number of clusters. All these lead to a myriad of alternative clustering solutions. Hence, we are faced with the following problem: "*for a given data set, which clustering solution should be selected?*".

While there is much work reported in the literature on validating data partitions produced by the traditional clustering algorithms [3], little has been done in order to validate data partitions produced by clustering combination methods. Most of the reported works use measures of consistency between consensus solutions and the clustering ensemble, such as Average Normalized Mutual Information [2] and Average Cluster Consistency [4]. The classical validity indices may also be used to assess the quality of the consensus partition. This requires the original data representation to be available,

which may not always be possible. Also, not considering clustering ensemble information should be a drawback, since the clustering structure, used by the clustering combination methods to produce the consensus partitions, is not used.

In this paper we propose the validation of clustering combination results at three levels:

- *original data representation* – measure the consistency of clustering solutions with the structure of the data, perceived from the original representation (either feature-based or similarity-based);
- *clustering ensemble level* – measure the consistency of consensus partitions with the clustering ensemble;
- *learned pairwise similarity* – measure the coherence between clustering solutions and the co-association matrix induced by the clustering ensemble.

Additionally to the methodology of evaluation at these distinct levels, we propose a new criterion based on likelihood estimates, and adaptation of "classical" cluster validity measures to pairwise similarity representations.

The remaining of the paper is organized as follows. Section 2 formulates the clustering ensemble problem, and describes the EAC method, that will be used in our experiments. The methodology for the validation of consensus partitions is presented in section 3. In section 4, a new validity index based on pairwise similarities is proposed. Experiments comparing all the validation measures are presented in section 5. Finally, the conclusions appear in section 6.

2 Clustering Combination

Let $\mathcal{X} = \{x_1, \cdots, x_n\}$ be a data set with n data patterns. Different partitions of $\mathcal{X}$ can be obtained by using different clustering algorithms, changing parameters and/or initializations for the same clustering algorithm, using different subsets of data features or patterns, projecting $\mathcal{X}$ to subspaces, and combinations of these. A clustering ensemble, $\mathcal{P}$, is defined as a set of N data partitions of $\mathcal{X}$:

$$\mathcal{P} = \{P^1, \cdots, P^N\}, \;\; P^l = \{C_1^l, \cdots, C_{K^l}^l\}, \tag{1}$$

where C_k^l is the k^{th} cluster in data partition P^l, which contains K^l clusters. Different partitions capture different views of the structure of the data. Clustering ensemble methods use a consensus function f which maps a clustering ensemble $\mathcal{P}$ into a consensus partition $P^* = f(\mathcal{P})$.

The Evidence Accumulation Clustering method (EAC) [1] considers each data partition $P^l \in \mathcal{P}$ as an independent evidence of data organization. The underlying assumption of EAC is that two patterns belonging to the same "natural" cluster will be frequently grouped together. A vote is given to a pair of patterns every time they co-occur in the same cluster. Pairwise votes are stored in a $n \times n$ co-association matrix, $\mathbf{C}$, normalized by the total number of combined data partitions, i.e., $\mathbf{C}_{ij} = \frac{\sum_{l=1}^{N} vote_{ij}^l}{N}$ where $vote_{ij}^l = 1$ if x_i and x_j co-occur in a cluster of data partition P^l; otherwise $vote_{ij}^l = 0$. The consensus partition is obtained by applying some clustering algorithm over the co-association matrix, $\mathbf{C}$.

3 Consensus Partition Validation

We herein propose the assessment of the quality of a consensus partition, P^*, by measuring its consistency at three levels: the original representation space; the clustering ensemble; and the learned pairwise similarity.

3.1 Validity Measures on the Original Data Space

Validity measures on the original data space are the most common approaches to perform clustering validation. The basic idea consists of evaluating a data partition using a utility or cost function, and comparing it with other partitions of the same data set. The utitlity/cost function usually measures the intra-cluster compactness and inter-cluster separation of a given data partition. Many different validity measures on the original data representation space have been proposed in the literature [3]. In this paper we will focus on three of them: the Silhouette, Dunn's and Davies-Bouldin indices.

Let $\mathcal{X} = \{x_1, \cdots, x_n\}$ be the data set, $P = \{C_1, \cdots, C_K\}$ its partition into K clusters, and $|C_l|$ the number of elements in the l-th cluster. Let $d(x_i, x_j)$ be the dissimilarity (distance) between data patterns x_i and x_j.

The Silhouette index [5] is formally defined as follows. Let a_i denote the average distance between $x_i \in C_l$ and the other patterns in the same cluster, and b_i the minimum average distance between x_i and all patterns grouped in another cluster:

$$a_i = \frac{1}{|C_l| - 1} \sum_{\substack{x_j \in C_l \\ j \neq i}} d(x_i, x_j), \quad b_i = \min_{k \neq l} \frac{1}{|C_k|} \sum_{x_j \in C_k} d(x_i, x_j). \tag{2}$$

The silhouette width, s_i, for each x_i, produces a score in the range $[-1, 1]$ indicating how well x_i fits in its own cluster when compared to other clusters; the global Silhouette index, S, is given by the average silhouette width computed over all samples in the data set:

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}, \qquad S = \frac{1}{n} \sum_{i=1}^{n} s_i \tag{3}$$

Dunn's index, quantifying how well a set of clusters represent compact and separated clusters [6], is defined as:

$$D = \frac{\min\limits_{1 \leq q \leq K} \; \min\limits_{1 \leq r \leq K, r \neq q} \operatorname{dist}(C_q, C_r)}{\max\limits_{1 \leq p \leq K} \operatorname{diam}(C_p)} \tag{4}$$

where $\operatorname{dist}(C_q, C_r)$ represents the distance between clusters C_q and C_r, and $\operatorname{diam}(C_p)$ is the p^{th} cluster diameter:

$$\operatorname{dist}(C_q, C_r) = \min_{x_i \in C_q,\, x_j \in C_r} d(x_i, x_j), \quad \operatorname{diam}(C_p) = \max_{x_i,\, x_j \in C_p} d(x_i, x_j). \tag{5}$$

The best partition is the one that maximizes the index value, D.

Davies-Bouldin index [7], is defined as the ratio of the sum of within-cluster scatter and the value of between-cluster separation:

$$DB = \frac{1}{K}\sum_{k=1}^{K}\max_{m\neq k}\left\{\frac{\Delta(C_k)+\Delta(C_m)}{d(\nu_k,\nu_m)}\right\},\ \ \Delta(C_k) = \frac{\sum_{x_i\in C_k} d(x_i,\nu_k)}{|C_k|} \tag{6}$$

where $\Delta(C_k)$ is the average distance between all patterns in C_k and their cluster center $\nu_k = \frac{\sum_{x_i\in C_k} x_i}{|C_k|}$. Small values of DB correspond to clusters that are compact, and whose centers are far away from each other. The data partition that minimizes DB is the optimal one.

3.2 Validity Measures on the Clustering Ensemble Space

These validity indices rely on the agreement between the consensus partition, P^*, and the partitions in the clustering ensemble $\mathcal{P} = \{P_1, \cdots, P_N\}$.

Let $H(P) = -\sum_{k=1}^{K} p(k)\log p(k)$ be the entropy of data partition P, with $p(k) = \frac{n_k}{n}$, and n_k the number of patterns in the k^{th} cluster of P. The mutual information between two data partitions, P^* and P^l, is defined as:

$$MI(P^*, P^l) = \sum_{i}^{K^*}\sum_{j}^{K^l}\frac{p(i,j)}{p(i)p(j)}, \tag{7}$$

with $p(i,j) = \frac{1}{n}|C_i^* \cap C_j^l|$, the fraction of shared samples in clusters C_i^* and C_j^l. Strehl and Ghosh [2] define the Average Normalized Mutual Information as:

$$ANMI(P^*, \mathcal{P}) = \frac{1}{N}\sum_{l=1}^{N}\frac{MI(P^*, P^l)}{\sqrt{H(P^*)H(P^l))}}. \tag{8}$$

Higher values of $ANMI(P^*, \mathcal{P})$ suggest better quality consensus partitions.

The Average Cluster Consistency [4] (ACC) is another validity measure based on the similarity between the partitions of the clustering ensemble and the consensus partition. The main idea consists of measuring how well the clusters C_m^l of the clustering ensemble fit in a cluster C_k^* of the consensus partition. If all patterns $x_i \in C_m^l$ belong to the same cluster C_k^*, for all clusters of the clustering ensemble, then the average cluster consistency between the consensus partition and the clustering ensemble is perfect. The ACC measures the similarity between two partitions, P^* and P^l, based on a weighting of shared samples in matching clusters:

$$\text{sim}(P^*, P^l) = \frac{1}{n}\sum_{m=1}^{K^l}\max_{1\leq k\leq K^*}|C_k^* \cap C_m^l|\left(1 - \frac{|C_k^*|}{n}\right), \tag{9}$$

where $K^l \geq K^*$. Note that cluster intersection, $|C_k^* \cap C_m^l|$, is weighted by $(1 - \frac{|C_k^*|}{n})$ in order to prevent high similarity values in situations where P^* has a few clusters with almost all the data patterns. The drawback is that consensus partitions with balanced

cluster cardinality are preferred. The ACC is defined as the average similarity between each data partition in the clustering ensemble ($P^l \in \mathcal{P}$) and the consensus partition P^*:

$$ACC(P^*, \mathcal{P}) = \frac{1}{N} \sum_{i=1}^{N} \text{sim}(P^i, P^*). \tag{10}$$

From a set of possible choices, the *best* consensus partition is the one that achieves the highest $ACC(P^*, \mathcal{P})$ value.

3.3 Validity Measures on a Similarity Space

In the following, modifications of the validity indices presented in subsection 3.1 are proposed, aiming to accommodate the same principles to a pairwise similarity representation. Consider a pairwise similarity measure $s(x_i, x_j)$ between pairs of patterns (x_i, x_j). In this paper, we will define $s(x_i, x_j) = \mathbf{C}_{ij}$, the pairwise similarity induced from the clustering ensemble [1], summarized in matrix $\mathbf{C}$ (see section 2).

In order to compute a Silhouette-like validity index in a similarity space, we propose to measure the within-cluster compactness and the inter-cluster separability adapting the formulas defined in equation 2 as below:

$$a_{s_i} = \frac{1}{|C_l| - 1} \sum_{\substack{x_j \in C_l \\ j \neq i}} s(x_i, x_j), \quad b_{s_i} = \max_{k \neq l} \frac{1}{|C_k|} \sum_{x_j \in C_k} s(x_i, x_j). \tag{11}$$

While in equation 2 low values for a_i and high values for b_i corresponded to high cluster compactness and separation, in equation 11 it is the opposite since we are using similarities. In this case, high values for a_{s_i} and low values for b_{s_i} imply good data partitions. For this reason, the numerator of equation 3 (left) is changed for the computation of the silhouette width, being defined as:

$$s_{s_i} = \frac{a_{s_i} - b_{s_i}}{\max\{a_{s_i}, b_{s_i}\}}. \tag{12}$$

The average silhouette width using similarities is then computed as $S_s = \frac{1}{n} \sum_{i=1}^{n} s_{s_i}$.

For Dunn's index, the similarity between the q^{th} and the r^{th}, and the diameter of C_p were redefined:

$$\text{sim}(C_q, C_r) = \max_{x_i \in C_q, x_j \in C_r} s(x_i, x_j), \quad \text{diam}_{\text{s}}(C_p) = \min_{x_i, x_j \in C_p} s(x_i, x_j). \tag{13}$$

By the fact that we are using similarities instead of distances, we take the inverse of equation 4 to define a Dunn-like validation index:

$$D_s = \frac{\min\limits_{1 \leq p \leq K} \text{diam}_{\text{s}}(C_p)}{\max\limits_{1 \leq q \leq K} \max\limits_{1 \leq r \leq K, r \neq q} \text{sim}(C_q, C_r) + 1}. \tag{14}$$

Since the information regarding the cluster centers $\{\nu_1, \cdots, \nu_K\}$ is not available in a similarity-based data representation, in our adaptation of the Davies and Bouldin's

validity index, it was necessary to introduce a new concept of center of a cluster. In order to incorporate pairwise similarities instead of the original vectorial data representation, we estimate the central pattern ν_k of cluster C_k as the element with maximum mean similarity within each cluster (innermost pattern), as defined below.

$$\nu_k = \arg\max_{x_i \in C_k} \sum_{\substack{x_j \\ j \neq i}} s(x_i, x_j), \tag{15}$$

Davies and Bouldin's validity index is redefined as

$$DB_s = \frac{1}{K} \sum_{k=1}^{K} \max_{m \neq k} \left\{ \frac{s(\nu_k, \nu_m)}{\Delta_s(C_k) + \Delta_s(C_m)} \right\}, \tag{16}$$

where $\Delta_s(C_k)$ is the average similarity between all patterns in C_k.

4 Statistical Validity Index based on Pairwise Similarity

We now propose a new validity index to assess the quality of P^* based on the likelihood of the data constrained to the data partition, $L(\mathcal{X}|\mathrm{P}^*)$, assessed from pairwise similarities, as per in the co-association matrix, $\mathbf{C}$, defined in section 2.

Our work is inspired in the Parzen-window density estimation technique [8] with variable size window, also known as K-nearest neighbor density estimation. This technique estimates the probability density of pattern x, $p(x)$, within a region R with volume V_R. The volume R is defined as a function of the K_N nearest neighbors of x, i.e., V_R is the volume enclosed by the region that contains all the K_N nearest neighbors of x. The probability density $p(x)$ is estimated as $\hat{p}(x) = \frac{K_N}{nV_R}$.

The new validity measure based on the likelihood of the data $\mathcal{X}$ (assuming $x \in \mathcal{X}$ to be independent and identically-distributed random variables) given a partition P, is defined as:

$$L(\mathcal{X}|P) = \prod_{i=1}^{N} \Pr(x_i|P), \quad \Pr(x_i|P) = \sum_{k=1}^{K} \Pr(x_i|C_k \in P). \tag{17}$$

Following the idea behind the Parzen-window density estimation method, we define the probability density of x_i given cluster C_k as:

$$\Pr(x_i|C_k) = \frac{|C_k \bigcap KNN(x_i)|}{|C_k| \cdot V(x_i)} \tag{18}$$

where $KNN(x_i)$ is the set of the K_N most similar data patterns to x_i based on the pairwise similarity measure defined by the co-association matrix $\mathbf{C}$, and $V(x_i)$ represents the volume of a sufficiently small region that contains all the patterns of the neighborhood $KNN(x_i) \bigcup \{x_i\}$. Since we rely only on pairwise similarities, as induced from the clustering ensemble, we aproximate the intrinsic volume $V(x_i)$ by a quantity proportional to it, defined by:

$$V(x_i) \triangleq \alpha \, \mathrm{diam}(x_i), \qquad \mathrm{diam}(x_i) = 2 \left(1 - \min_{x_j \in KNN(x_i)} \mathbf{C}_{ij} \right) \tag{19}$$

where $\alpha > 0$ is a scalar chosen such that $\sum_{i=1}^{n} p(x_i) = 1$, and $\mathrm{diam}(x_i)$ represents the "diameter" of the region centered at x_i that contains the neighborhood of x_i. Since the similarity matrix, $\mathbf{C}$, takes values in the interval $[0; 1]$, the above transformation $1 - \mathbf{C}_{ij}$ leads to a dissimilarity measure; the diameter thus corresponds to twice the dissimilarity of the $K_N - th$ nearest neighbor of x_i.

Using equations 17-18, the likelihood of the data set $\mathcal{X}$ given a data partition P is defined as:

$$L(\mathcal{X}|P) = \prod_{i=1}^{N} \sum_{k=1}^{K} \frac{|C_k \bigcap KNN(x_i)|}{|C_k| \cdot V(x_i)}. \tag{20}$$

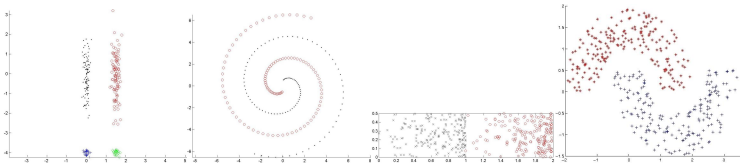
The underlying reasoning for using L as a validity index is the following.

Given a clustering ensemble, the co-association matrix, $\mathbf{C}$, corresponds to the maximum likelihood estimate of the probability of pairwise co-occurrence of patterns in a cluster. Taking this co-occurrence probability as the pattern pairwise similarity induced by the CE, the likelihood of the data set $\mathcal{X}$ given a combination partition P^* is estimated by $L(\mathcal{X}|P^*)$. The statistical validity index based on the pairwise similarity, L, thus corresponds to a goodness of fit of the combined partition, P^*, with the clustering ensemble and the pairwise information extracted from it. Best combination strategies should therefore lead to highest likelihood values, L, of the data.

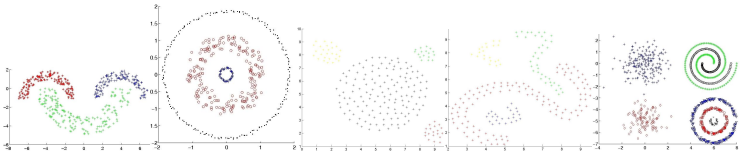
In a similar way, we can compute the likelihood of the data given the combination partition using the original data representation space. In this case, the likelihood L corresponds to a goodness of fit of the combined partition, P^*, with the statistical properties of the data on the original representation. In the following we denote by L_O the likelihood computed from the original data representation, and by L_S the likelihood computed from the co-association matrix (induced similarity).

5 Experimental Results

Five real (available at the UCI repository http://archive.ics.uci.edu/ml) and nine synthetic data sets were used to assess the performance of the validity measures on a wide variety of situations, including data sets with arbitrary cluster shapes, different cardinality and dimensionality, well-separated and touching clusters, and distinct cluster densities. The Iris data set consists of 50 patterns from each of three species of iris flowers, characterized by four features. The Std Yeast is composed of 384 patterns (normalized to have 0 mean 0 and unit variance) characterized by 17 features, split into 5 clusters concerning 5 phases of the cell cycle. The Optdigits is a subset of Handwritten Digits data set containing only the first 100 patterns of each digit, from a total of 3823 data samples characterized by 64 attributes. The House Votes data set is composed of two clusters of votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the Congressional Quarterly Almanac. From a total of 435 (267 democrats and 168 republicans) only the patterns without missing values were considered, resulting in 232 patterns (125 democrats and 107 republicans). The Wine data set consists of the results of a chemical analysis of wines grown in the same region in Italy divided into three clusters with 59, 71 and 48 patterns described by 13 features. Both House Votes and Wine data sets were normalized to have unit variance. The synthetic data sets are shown in figure 1.



(a) Cigar (b) Spiral (c) Bars (d) 2 Half Rings



(e) 3 Half Rings (f) Concentric (g) D1 (h) D2 (i) Complex

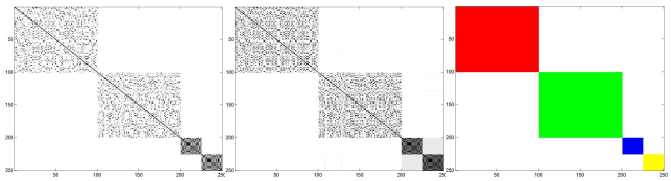
Fig. 1: Synthetic data sets.

Table 1: $NMI(P^*, P^0)$ for the consensus partitions selected by each validity measure.

	Clustering Ensemble Construction Method A											Clustering Ensemble Construction Method B									
Data Set	L_o	S_o	D_o	DB_o	L_s	S_s	D_s	DB_s	ANMI	ACC	Best	L_o	S_o	D_o	DB_o	L_s	S_s	D_s	DB_s	ANMI	ACC	Best
Iris	**0.81**	**0.81**	0.71	0.71	**0.81**	**0.81**	0.71	**0.81**	**0.81**	**0.81**	0.81	**0.81**	**0.81**	0.72	0.72	**0.81**	**0.81**	0.72	0.72	**0.81**	**0.81**	0.81
Std Yeast	0.49	0.49	0.08	**0.53**	0.49	**0.53**	0.24	0.08	0.49	0.49	0.53	0.48	0.48	0.37	0.32	0.48	**0.53**	0.23	0.48	0.48	0.48	0.53
Optdigits	**0.81**	**0.81**	0.71	0.63	**0.81**	**0.81**	0.63	**0.81**	**0.81**	**0.81**	0.81	0.81	**0.83**	**0.83**	0.72	0.81	0.81	0.72	**0.83**	0.81	**0.83**	0.83
House Votes	**0.50**	**0.50**	0.03	0.03	**0.50**	0.14	0.14	0.14	**0.50**	**0.50**	0.50	**0.49**	**0.49**	0.02	**0.49**	**0.49**	**0.49**	0.14	0.14	**0.49**	**0.49**	0.49
Wine	**0.77**	**0.77**	0.66	0.08	**0.77**	0.66	0.08	0.66	**0.77**	**0.77**	0.77	0.77	**0.80**	0.06	0.17	**0.80**	0.77	0.17	0.06	0.77	0.77	0.80
Cigar	**1.00**	**1.00**	**1.00**	0.23	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00	0.84	**1.00**	**1.00**	**1.00**	0.84	**1.00**	0.38	**1.00**	0.84	0.84	1.00
Spiral	**1.00**	0.00	0.05	0.05	**1.00**	0.00	**1.00**	**1.00**	0.00	0.00	1.00	0.01	0.01	0.08	0.08	0.01	0.01	**1.00**	**1.00**	0.01	0.01	1.00
Bars	**0.94**	**0.94**	0.06	0.06	**0.94**	**0.94**	0.06	**0.94**	**0.94**	**0.94**	0.94	**0.94**	**0.94**	0.21	0.21	**0.94**	**0.94**	0.21	0.21	**0.94**	**0.94**	0.94
2 Half Rings	**0.99**	**0.99**	0.17	0.17	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	0.99	0.87	**0.99**	0.21	0.21	0.87	0.87	**0.99**	**0.99**	0.87	0.87	0.99
3 Half Rings	**1.00**	**1.00**	0.08	0.08	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00
Concentric	**1.00**	**1.00**	0.09	0.09	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00	0.70	**1.00**	0.14	0.14	0.70	0.70	**1.00**	**1.00**	0.70	0.70	1.00
D1	**1.00**	**1.00**	**1.00**	0.05	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00	0.40	**1.00**	**1.00**	**1.00**	0.40	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00
D2	**1.00**	**1.00**	0.14	0.14	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00	0.57	0.71	0.34	0.34	0.57	0.57	**1.00**	**1.00**	0.71	0.71	1.00
Complex	**0.83**	0.44	**0.83**	0.44	**0.83**	**0.83**	0.82	0.82	**0.83**	**0.83**	0.83	**0.70**	**0.70**	**0.70**	0.63	**0.70**	0.63	0.56	**0.70**	**0.70**	**0.70**	0.87
#Best criterion	13	11	3	1	13	11	7	10	12	12		5	11	5	4	6	7	6	9	6	7	

For each data set, two different methods were used to build the clustering ensembles. In the first method (A), the K-means algorithm was used to produce $N = 150$ data partitions, each one with exactly $K = 20$ clusters for the Iris data set, $K = 50$ for the Concentric data sets, $K = 120$ for the Complex data set, and $K = 30$ for all the other data sets. In the second method (B), the K-means algorithm was also used to build clustering ensembles with the same size, but the number of clusters for each data partition was randomly chosen to be an integer in the interval $[10; 30]$. The clustering ensemble construction method A (leading to $\mathcal{P}^A$) is expected to be a "good" clustering ensemble, in the sense that its clusters have less probability of mixing patterns from different "natural" clusters than the clustering ensemble construction method B ($\mathcal{P}^B$), since K^l, $\forall C_l \in \mathcal{P}^A$ is always higher than $\min_{C^l \in \mathcal{P}^B} K^l$. The consensus partitions were obtained applying the EAC method using the Single-Link, Average-Link, Complete-Link, Centroid-Link and Ward-Link hierarchical clustering algorithms at the final step.

Table 1 shows the $NMI(P^*, P^0)$ values between the best data partition P^*, according to each validity measure, and the "real" (ground-truth) data partition P^0. The subscripts $_o$ and $_s$ point out that the validity measure was evaluated on the original



(a) (b) (c)

Fig. 2: Co-association matrices for (a) CE construction method A, (b) CE construction method B and (c) "natural" partition of data, for Cigar data set.

space or the similarity space, respectively. The columns designated by "Best" indicate the value of NMI for the best obtained consensus partition. In order to use the criterion based on the likelihood estimates (L) on the original space, the diameter of a region was computed as $\mathrm{diam}(x_i) = 2\max_{x_j \in KNN(x_i)} d(x_i, x_j)$, using the Euclidean distance to measure dissimilarity, and $KNN(x_i)$ corresponds to the set of the K_N closest patterns to x_i. The results for the clustering ensemble construction method A show that the L validity measure had the best performance, both on the original and similarity spaces, selecting the best consensus partition in 13 out of 14 data sets, followed by $ANMI$ and ACC criteria with 12, and S_O and S_S with 11. While L_o and L_s selected the same partitions, S_o and S_s had different choices on several data sets. The performances of D and DB were better on the pairwise similarity space than on the original space, suggesting that the first should be preferred. For the clustering ensemble construction method B, S on the original space was the best validity measure, being the best criterion in 11 data sets. DB was the best on the similarity space by selecting in 9 data sets equal or better partitions than the other indices. L was the best criterion only 5 times on the original space and 6 on the similarity space. The poor performance of L is due to its sensibility to "bad" clustering ensembles. Figure 2 shows the co-association matrices for construction methods A and B and the "natural" partition for the Cigar data set. While in figure 2 (a) there are no co-associations between patterns belonging to different "natural" clusters, in figure 2 (b) it can be seen (especially on the lower right corner) that some patterns from distinct "natural" clusters have co-association different from 0. This explains why the L_s performed correctly on the clustering ensemble construction method A and not on B.

From the comparison involving the criteria on the original and similarity spaces, we conclude that L (on both spaces) is the best choice if the clustering ensemble is "good", S is robust on the original space, and D was the worst criterion (despite that the similarity space version presents better results than the original space version). We also conclude that the consensus partition evaluation may also be restricted to the co-association matrix. This has the advantages of exploring sparse similarities representations (particularly when using L_s) and complying with data privacy. Evaluating consensus partitions on the original space has also another disadvantage: *how to validate a consensus partition if the partitions belonging to the clustering ensemble were produced using different representations (e.g. distinct subset of feature, random projections, etc)?*

By comparing the criteria on the similarity spaces with the criteria based on the consistency between the clustering ensemble partitions and the consensus partition, L_s was better than ANMI and ACC in construction method A, and DB_s was better in construction method B; so we can discard both ANMI and ACC, and rely instead on the similarity-based criteria in order to assess the consensus partitions.

6 Conclusions

The validation of clustering solutions were proposed at three distinct levels: original data representation, learned pairwise similarity, and consistency with the clustering ensemble partitions. A new validity measure based on the likelihood estimation of pattern pairwise co-occurrence probabilities was introduced. Experimental results seem to indicate that: the new validity measure is a good choice for performing consensus clustering validation when the clusters belonging to the clustering ensemble are not likely to contain patterns of different "natural" clusters; the learned similarity-based criteria can be used, instead of the traditional clustering ensemble measure; and the similarity-based criteria are a good option when the original data representation is not available. More extensive evaluation of the validity indices is being conducted over a larger number of data sets and on the comparison of consensus results produced by different combination strategies.

7 Acknowledgments

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). This work was partially supported by the Portuguese Foundation for Science and Technology (FCT), Portuguese Ministry of Science and Technology, under grant PTDC/EIACCO/103230/2008.

References

1. Fred, A., Jain, A.: Combining multiple clustering using evidence accumulation. IEEE Trans Pattern Analysis and Machine Intelligence **27**(6) (June 2005) 835–850
2. Strehl, A., Ghosh, J.: Cluster ensembles — a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3** (2003) 583–617
3. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. Intelligent Information Systems Journal **17**(2-3) (2001) 107–145
4. Duarte, F.J., Duarte, J.M.M., Rodrigues, M.F.C., Fred, A.L.N.: Cluster ensemble selection using average cluster consistency. In: KDIR '09: Proc. of Int. Conf. on Knowledge Discovery and Information Retrieval. (October 2009)
5. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics **20** (1987) 53–65
6. Dunn, J.C.: A fuzzy relative of the isodata process and its use in detecting compact, well separated clusters. Cybernetics and Systems **3**(3) (1974) 32–57
7. Davies, D., Bouldin, D.: A cluster separation measure. IEEE Transaction on Pattern Analysis and Machine Intelligence **1**(2) (1979)
8. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification (2nd Edition). 2 edn. Wiley-Interscience (November 2000)

[8] - (SIMBAD Technical Report n. 2010_28)

Lourenço, A., Fred, A.L., Jain, A.K.: On the scalability of evidence accumulation clustering. In: 20th International Conference on Pattern Recognition (ICPR 2010), Istanbul Turkey, IEEE Computer Society (August 23-26 2010)

On the Scalability of Evidence Accumulation Clustering

André Lourenço*†‡, Ana L. N. Fred†‡ and Anil K. Jain§
*Instituto Superior de Engenharia de Lisboa
†Instituto Superior Técnico
‡Instituto de Telecomunicações, Lisboa, PORTUGAL
§Michigan State University, USA
alourenco@deetc.isel.ipl.pt afred@lx.it.pt jain@msu.edu

***Abstract*—This work focuses on the scalability of the Evidence Accumulation Clustering (EAC) method. We first address the space complexity of the co-association matrix. The sparseness of the matrix is related to the construction of the clustering ensemble. Using a split and merge strategy combined with a sparse matrix representation, we empirically show that a linear space complexity is achievable in this framework, leading to the scalability of EAC method to clustering large data-sets.**

***Keywords*-Cluster analysis; combining clustering partitions; cluster fusion, evidence accumulation; large data-sets.**

I. Introduction

Clustering combination techniques are a recent and promising trend in clustering [1], [2], [3], [4], [5], [6]. Combining the information provided by a set of N different partitions (the *clustering ensemble* (CE) - $\mathbb{P}$) of a given data set, clustering combination results typically outperform the result of a single clustering algorithm, achieving better and more robust partitioning of the data.

The Evidence Accumulation Clustering (EAC) method, proposed by Fred and Jain [1], [2], seeks to find consistent data partitions by considering pair-wise relationships. The method can be decomposed into three major steps: (a) construction of the clustering ensemble, $\mathbb{P}$; (b) ensemble combination, through evidence accumulation; and (c) extraction of the final partition.

In the combination step (b), the clustering ensemble, $\mathbb{P}$, is transformed into a learned pair-wise similarity, summarized in a $n_s \times n_s$ co-association matrix, $\mathcal{C}$

$$\mathcal{C}(i,j) = \frac{n_{ij}}{N}, i,j \in 1,\ldots,N, \qquad (1)$$

where n_s is the number of objects to be clustered, and n_{ij} represents the number of times a given object pair (i,j) is placed in the same cluster over the N partitions of the ensemble.

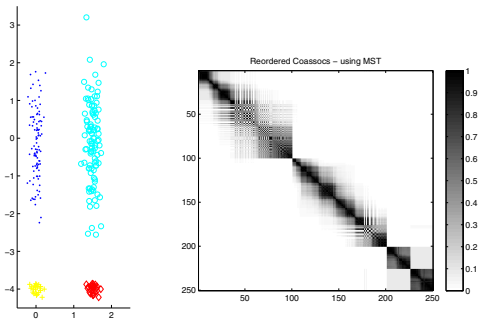
In order to recover the "natural"clusters, a clustering algorithm is applied to the learned similarity matrix, $\mathcal{C}$, yielding the combined data partition, P^*. Although it is mostly the hierarchical agglomerative methods that have been applied in step (c) [2], any clustering algorithm can be used, either taking $\mathcal{C}$ as a pair-wise similarity matrix, or deriving a feature space from it using multi-dimensional scaling (MDS).

EAC is a powerful and robust method, but direct or naive implementation of its basic steps can, however, limits the scalability of the EAC method, namely due to the $O(n_s^2)$ space complexity required to store the co-association matrix [6], [5].

In this paper we address the scalability of EAC, theoretically analyzing the method from a space complexity perspective. We propose: (1) a compact representation of the co-association matrix, $\mathcal{C}$, exploring its intrinsic sparseness; and (2) guidelines for the construction of the clustering ensemble $\mathbb{P}$, that further increases the sparseness of $\mathcal{C}$, leading to an overall split and merge strategy for the EAC. Experimental results, on several benchmark data-sets, confirm that this strategy leads to a linear space complexity. We show that this significant space complexity improvement does not compromise, and may even lead to increased performance of clustering combination results.

II. Co-association Matrix Representation

Typically, the co-association matrix, $\mathcal{C}$, generated by the EAC method, is very sparse. This is illustrated with the synthetic 2D cigar data set (figure 1(a)), and the corresponding $\mathcal{C}$ matrix (figure 1(b)). The color scheme ranges from white ($\mathcal{C}(i,j) = 0$) to black ($\mathcal{C}(i,j) = 1$), corresponding to the magnitude of similarity.



(a) Cigar data. (b) Co-association Matrix.

Figure 1. Synthetic 2-D data set (a) with $n_s = 250$ objects, and the corresponding co-association matrix (b).

Under the *working hypothesis* of well separated and

balanced clusters, the structure of the co-association matrix resembles a perfect block diagonal matrix, where each block corresponds to a cluster, and the number of co-associations (non-zero elements of the matrix $\mathcal{C}$), is given by:

$$N_{assocs} = \sum_{k=1}^{K} (n_{sk})^2, \tag{2}$$

where K is the number of "natural"clusters in the data-set, and n_{sk} is the number of samples in cluster k.

Taking into account the symmetric nature of this matrix, and the fact that the principal diagonal elements have value 1, the required elements that need to be retained consist only of the upper (or lower) triangular matrix, with a total number of co-associations given by:

$$N_{assocs\triangle} = \sum_{k=1}^{K} n_{sk} \times (n_{sk} - 1)/2. \tag{3}$$

We propose to use a **sparse representation** of the co-association matrix, $\mathcal{C}$, storing only the upper triangular non-zero elements of this matrix, substantially reducing the space complexity of the method and making it more attractive for large data sets.

III. Building CEs: Split and Merge Strategy

The sparseness of a matrix can be quantified by its *density*, or normalized ℓ^0 norm, defined by $\| \mathcal{C} \|_0 = nnz/n_s^2$, where nnz is the number of non-zero elements. The density of a perfect K block diagonal matrix $\mathcal{C}$ is given by

$$\| \mathcal{C} \|_0 = \frac{\sum_{k=1}^{K} (n_{sk})^2}{(n_s)^2} \tag{4}$$

Considering balanced clusters, each cluster has $\frac{n_s}{K}$ elements, and the density becomes $\| \mathcal{C} \|_0 = \frac{1}{K}$. Empirically, the value of $\| \mathcal{C} \|_0$ becomes smaller than $1/K$, as the number of co-associations becomes less than N_{assocs}. This number depends on the strategy used for generating the clustering ensemble.

The splitting of "natural"clusters into smaller clusters induces micro-blocks (smaller than the perfect block diagonal structures) in the $\mathcal{C}$ matrix, resulting in an increased sparseness (lower density). In order to achieve this, we propose the following strategy for building clustering ensembles:

CE construction rule: Apply several clustering algorithm(s) with many different values of K, the number of clusters in each partition of the clustering ensemble; K is randomly chosen in an interval $[K_{min}, K_{max}]$.

A large value of K_{min}, in addition to inducing high granularity partitioning and consequently reduced space complexity, is important in order to prevent the existence of clusters in the CE with samples from different "natural"clusters. Overall, this follows a split & merge strategy [2], with the split "natural"clusters being combined in $\mathcal{C}$ during the combination step (b) of the EAC method; they are eventually recovered during the merging step (c) produced by clustering the matrix $\mathcal{C}$. One possible choice for K_{min} is to base it on the minimum number of gaussians in a gaussian mixture decomposition of the data [7].

We propose and analyze two alternative criteria for determining $\{K_{min}, K_{max}\}$, as a function of n_s, the number of samples in the data set:

(A) Sqrt: $\{K_{min}, K_{max}\} = \{\lceil \sqrt{n_s}/2 \rceil, \lceil \sqrt{n_s} \rceil\}$;

(B) Linear: $\{K_{min}, K_{max}\} = \{\lceil n_s/A \rceil, \lceil n_s/B \rceil\}$, with $A > B$

The number of non-zero elements in $\mathcal{C}$ is related to the number of associations within each "natural"cluster over each partition of the ensemble, i.e., the partitioning granularity. According to the working hypothesis of well separated and balanced clusters, each cluster C_m, from a data partition with K clusters, should have $n_{sm} = n_s/K$ objects, contributing to $(n_{sm})^2$ entries in $\mathcal{C}$; overall, a single partition produces $K(n_{sm})^2 = (n_s)^2/K$ non-zero values in $\mathcal{C}$. Over the N partitions of the clustering ensemble, a random partitioning of the "natural"clusters leads to the construction of partially overlapping clusters. Notice that shared elements of two overlapping clusters produce exactly the same co-associations in the matrix $\mathcal{C}$; new entries in the co-association matrix are the result of the non-overlapping elements. The density of $\mathcal{C}$ is thus larger for smaller values of K (with K_{min} giving the minimum value) and lower cluster overlap. On average, we consider that the overall contribution of the clustering ensemble (including unbalanced clusters) duplicates the co-associations produced in a single balanced clustering with K_{min} clusters, leading to the following estimate of the number of associations using the proposed clustering ensemble construction rule:

$$N_{assocs\_S\&M} = \frac{2(n_s)^2}{K_{min}} \overset{\text{(SqrtT)}}{=} 4n_s\sqrt{n_s} \tag{5}$$

$$\overset{\text{(LinearT)}}{=} 2A \cdot n_s \tag{6}$$

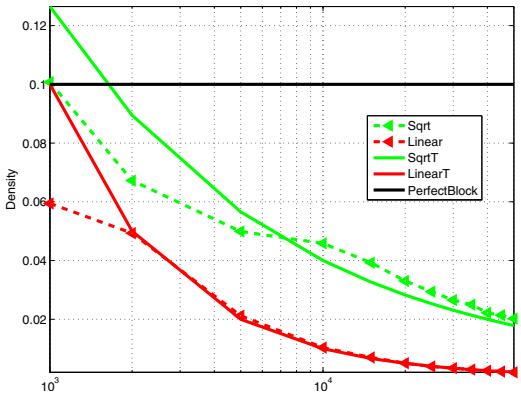
where (5) and (6) represent, respectively, the estimates for the criteria (A) and (B), the latter corresponding to linear space complexity. The corresponding estimated densities are $\| \mathcal{C} \|_0 = \frac{4}{\sqrt{n_s}}$ and $\| \mathcal{C} \|_0 = \frac{2A}{n_s}$.

For the sake of simplicity, in the next section we illustrate and evaluate this strategy using K-means clusterings for constructing the ensemble.

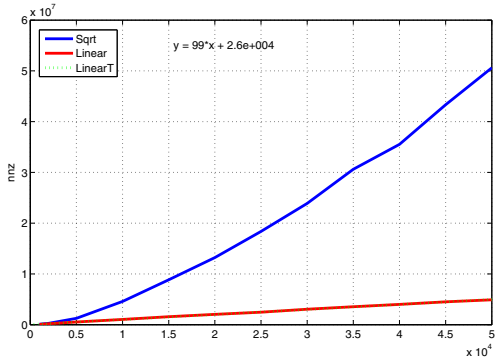
IV. Experimental Evaluation

To illustrate the performance of the proposed sparse representation, consider a mixture of Gaussians composed

of 10 2D-gaussians, with equal number of samples ($n_s/10$), means $\mu_i = [0, 12i]$ and covariances $\Sigma_i = [1, 0; 0, 1]$, with n_s in the interval $[10^3; 5 \times 10^4]$. For each n_s, we create a clustering ensemble with N=150 partitions, produced using the K-means algorithm with random number of clusters, following the criteria proposed in section III (with A=50, and B=20). Figure 2 plots the evolution of the density of $\mathcal{C}$ and the number of non-zero elements in $\mathcal{C}$, as a function of the number of samples (n_s), for criteria (A) Sqrt and (B) Linear.



(a) Density of co-association matrix (n_s is in logarithmic scale).



(b) Number of Non-Zero Elements.

Figure 2. Density (a) and number of non-zero elements (b) of co-association matrices as a function of n_s for a mixture of gaussians.

As shown in figure 2(a), both criteria lead to a decrease in density as n_s increases. Theoretical estimates (curves SqrtT and LinearT) seem reasonable for large n_s values, providing a good match with the corresponding experimental results (curves Sqrt and Linear), in particular for the Linear criterion (B). Both criteria lead to experimental density values far below the curve PerfectBlock, corresponding to the density of a perfect block diagonal matrix, as per equation (4).

Figure 2(b) plots the number of non-zero elements of $\mathcal{C}$ as a function of n_s. The line LinearT represents a linear regression over the empirical results using criteria (B). The plot is consistent with the proposed theoretical estimate, reinforcing the observation that the use of criteria (B) enables linear space complexity.

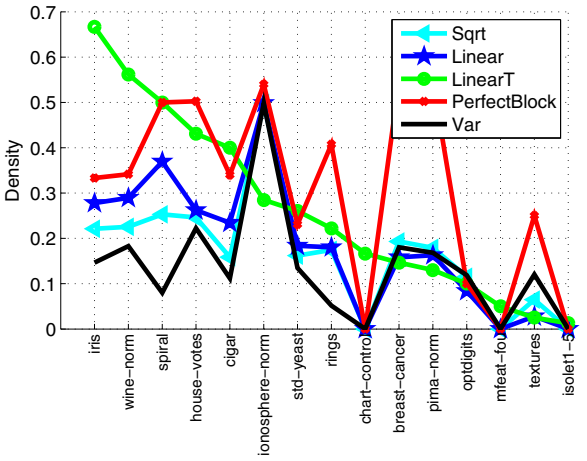


Figure 3. Density of co-association matrices for benchmark data-sets.

Table I characterizes several benchmark data-sets (synthetic and other from the UCI repository [8]), values of k_{min} and k_{max} used for building clustering ensembles, and accuracy of corresponding combined partitions using the EAC method (columns CI – represent the Consistency Index [1], obtained by matching the clusters in the combined partition with the ground truth labels, corresponding to the percentage of correct labeling). In addition to criteria (A) and (B), the columns (Var) represent K intervals with $k_{min} \geq 10$, ensuring that k_{min} is always larger than the minimum number of components in a gaussian mixture decomposition [7].

Figure 3 presents the densities of the co-association matrices obtained for these data-sets for several clustering ensemble building criteria. By cross analysis of table I and figure 3, we can observe that higher k_{min} values lead to lower density. As a result, the criterion Var achieves lower densities for most data sets, corresponding to situations of higher k_{min} values than those produced by the other criteria. It is also evident that the proposed criteria induce lower densities, when compared to the PerfectBlock curve. Experimental results with criterion (B) in these data sets show typically lower densities than the theoretical estimate, LinearT.

In table I, for each data set we have marked maximal k_{min} and CI values. Analysis of these results show that the granularity of the clustering ensemble, dictated by the value of k_{min}, positively influences the quality of the clustering results. In general, higher k_{min} values do not compromise and may even lead to higher CI values.

Data-Sets	K	n_s	Var			(A)			(B)		
			k_{min}	k_{max}	CI	k_{min}	k_{max}	CI	k_{min}	k_{max}	CI
iris	3	150	10	20	0.91	6	12	0.84	3	8	0.84
wine-norm	3	178	10	30	0.96	7	13	0.96	4	9	0.97
spiral	2	200	20	30	0.71	7	14	0.57	4	10	0.54
house.votes	2	232	10	30	0.93	8	15	0.88	5	12	0.88
cigar	4	250	10	30	0.71	8	16	0.71	5	13	0.71
ionosphere.norm	2	351	10	30	0.64	9	19	0.64	7	18	0.64
std-yeast	5	384	10	30	0.69	10	20	0.66	8	19	0.68
rings	3	450	20	50	1.00	11	21	0.60	9	23	0.72
chart-synthetic-control	10	600	13	33	0.57	13	21	0.57	12	30	0.54
breast-cancer	2	683	10	30	0.97	13	26	0.97	14	34	0.97
pima-norm	2	768	10	30	0.65	14	28	0.65	15	38	0.65
optdigits-1000	10	1000	10	30	0.79	16	32	0.80	20	50	0.84
mfeat-fou	4	2000	40	60	0.39	23	45	0.39	40	100	0.39
textures	4	4000	10	30	0.90	32	63	0.97	80	200	0.91
isolet1-5	26	7797	156	176	0.60	44	88	0.61	156	390	0.60

Table I
BENCHMARK DATA-SETS AND CLUSTERING RESULTS, IN TERMS OF CONSISTENCY INDEX, CI, USING THE AVERAGE LINK HIERARCHICAL CLUSTERING TO OBTAIN THE FINAL PARTITION. VAR: MIXTURE OF GAUSSIANS; A: SQRT CRITERION; B: LINEAR CRITERION.

V. CONCLUSIONS

We have addressed the scalability problem of the evidence accumulation clustering method, intrinsically related to the storage of the co-association matrix. Taking advantage of the sparseness of this matrix, we adopted a sparse matrix representation, reducing the space complexity of the method.

In order to further reduce the space complexity, we have proposed a clustering ensemble construction rule, following a split and merge strategy, according to which the clustering algorithms are applied with K, the number of clusters, randomly chosen in the interval $[K_{min}, K_{max}]$. Criteria for the choice of these extreme values were also proposed and analyzed, showing that both space complexity and quality of combination results dependent on the partitioning granularity, dictated by the value of K_{min}.

Experimental results confirm that this strategy leads to linear space complexity of evidence accumulation clustering on several benchmark data, enabling the scalability of this framework to large data-sets. We have shown that this significant space complexity improvements do not compromise, and may even lead to increased performance of clustering combination. The experiments also confirmed linear time complexity. Additional experiments on larger data sets are underway.

ACKNOWLEDGMENT

We acknowledge the financial support from the FET programme, within the EU FP7, under the SIMBAD project (contract no.213250).

REFERENCES

[1] A. Fred, "Finding consistent clusters in data partitions," in *Multiple Classifier Systems*, J. Kittler and F. Roli, Eds., vol. 2096. Springer, 2001, pp. 309–318.

[2] A. Fred and A. Jain, "Combining multiple clustering using evidence accumulation," *IEEE Trans Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835–850, June 2005.

[3] A. Strehl and J. Ghosh, "Cluster ensembles - a knowledge reuse framework for combining multiple partitions," *J. of Machine Learning Research 3*, 2002.

[4] A. Topchy, A. Jain, and W. Punch, "A mixture model of clustering ensembles," in *Proc. of the SIAM Conf. on Data Mining*, April 2004.

[5] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *Proc ICML '04*, 2004.

[6] H. G. Ayad and M. S. Kamel, "Cumulative voting consensus method for partitions with variable number of clusters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 1, pp. 160–173, 2008.

[7] M. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 381–396, 2002.

[8] A. Asuncion and D. Newman, "UCI ML repository," 2007. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html

[11] - (no technical report)

Medina, L.A., Fred, A.L.N.: Clustering data with temporal evolution: Application to electrophysiological signals. In Filipe, J., Fred, A., Sharp, B., eds.: Agents and Artificial Intelligence. Volume 129 of Communications in Computer and Information Science. Springer Berlin Heidelberg (2011) 101–115 10.1007/978-3-642-19890-8_8.

Clustering Data with Temporal Evolution: Application to Electrophysiological Signals

Liliana A.S. Medina and Ana L.N. Fred

Instituto de Telecomunicações, Instituto Superior Técnico
Lisbon, Portugal
{lmedina,afred}@lx.it.pt
http://www.it.pt

Abstract. Electrocardiography signals (ECGs) are typically analyzed for medical diagnosis of pathologies and are relatively unexplored as physiological behavioral manifestations. In this work we analyze these signals by employing unsupervised learning methods with the intent of assessing the existence of significant changes of their features related to stress occurring in the performance of a computer-based cognitive task. In the clustering context, this continuous change of the signal means that it is difficult to assign signal samples to clusters such that each cluster corresponds to a differentiated signal state.

We propose a methodology based on clustering algorithms, clustering ensemble methods and evolutionary computation for detection of patterns in data with continuous temporal evolution. The obtained results show the existence of differentiated states in the data sets that represent the ECG signals, thus confirming the adequacy and validity of the proposed methodology in the context of the exploration of these electrophysiological signals for emotional states detection.

Keywords: Unsupervised learning, Temporal data, Genetic algorithm, Electrocardiogram, ECG, Stress detection.

1 Introduction

Of the existing classification methods, unsupervised learning is especially appealing to organize data which has little or no labeling information associated to it [3]. A clustering algorithm organizes the patterns into k groups or clusters, based on the similarity or dissimilarity values between pairs of objects, such that objects in the same cluster are more similar than objects of different clusters [5][3]. The adopted similarity might be statistical or geometrical, such as a proximity measure based on a distance metric in the d-dimensional representation space of the d features that characterize the data [5]. The result will be a partition of the analyzed data set.

The work presented here is centered on the analysis of time series of electrophysiological signals, from an unsupervised learning perspective, to assess in particular the existence of differentiated emotional states. Given that typically the signal is characterized by a continuous temporal evolution, this means that the values of the features that represent it will also change gradually with time and that will possibly reflect transient emotional states present in the structure of the signal. In the clustering context, the fact

J. Filipe, A. Fred, and B. Sharp (Eds.): ICAART 2010, CCIS 129, pp. 101–115, 2011.

that such transient states occur, means that it is difficult to assign signal samples to clusters such that each cluster corresponds to a differentiated state.

This represents a challenge in the implementation of clustering methods to analyze time series, like the aforementioned electrophysiological signals, because the clusters are not well separated, which in turn introduces ambiguities in the observation of differentiated emotional states. In order to assess and evaluate the existence of these emotional states, we propose an analysis methodology based on a genetic algorithm combined with clustering techniques. The goal of this work methodology is to eliminate transient states in order to clarify the existence of well separated clusters, each corresponding to differentiated states present in the data time series. This methodology can be applied to electrophysiological signals acquired during the performance of cognitive tasks, such as electrocardiography signals (ECG) or electroencephalography signals (EEG). In this paper we specifically address the identification of stress from ECG signals.

2 Proposed Methodology

We propose a methodology for analysis of temporal data series, represented in Fig. 1. It is based on unsupervised learning techniques in order to unveil similarity relations between the temporal patterns that represent the data, and also to detect differentiated states in the temporal sequences that represent the data, by applying a genetic algorithm specifically conceived for this purpose.

After the acquisition and preprocessing of electrophysiological signals, these are represented by a set of j samples. Each sample corresponds to a given segment of the signal, therefore being associated to a time stamp, and is characterized by a d-dimensional feature vector, $f = [f_1 ... f_d]$.

The proposed methodology encompasses steps of learning similarities between temporal patterns, and the detection of states from these. These two main steps are described in the following subsections. The overall process consists of refining the state detection by means of a genetic algorithm that uses the output of these clustering and state detection procedures.

2.1 Learning Similarities with Evidence Accumulation

Different clustering algorithms lead in general to different clustering results. A recent approach in unsupervised learning consists of producing more robust clustering results by combining the results of different clusterings. Groups of partitions of a data set are called clustering ensembles and can be generated by choice of clustering algorithms or algorithmic parameters, as described in [2]. Evidence Accumulation (EAC) is a clustering ensemble method that deals with partitions with different number of clusters by employing a voting mechanism to combine the clustering results, leading to a new measure of similarity between patterns represented by a co-association matrix. The underlying assumption is that patterns belonging to a natural cluster are very likely to be assigned in the same cluster in different partitions. Taking the co-occurrences of pairs

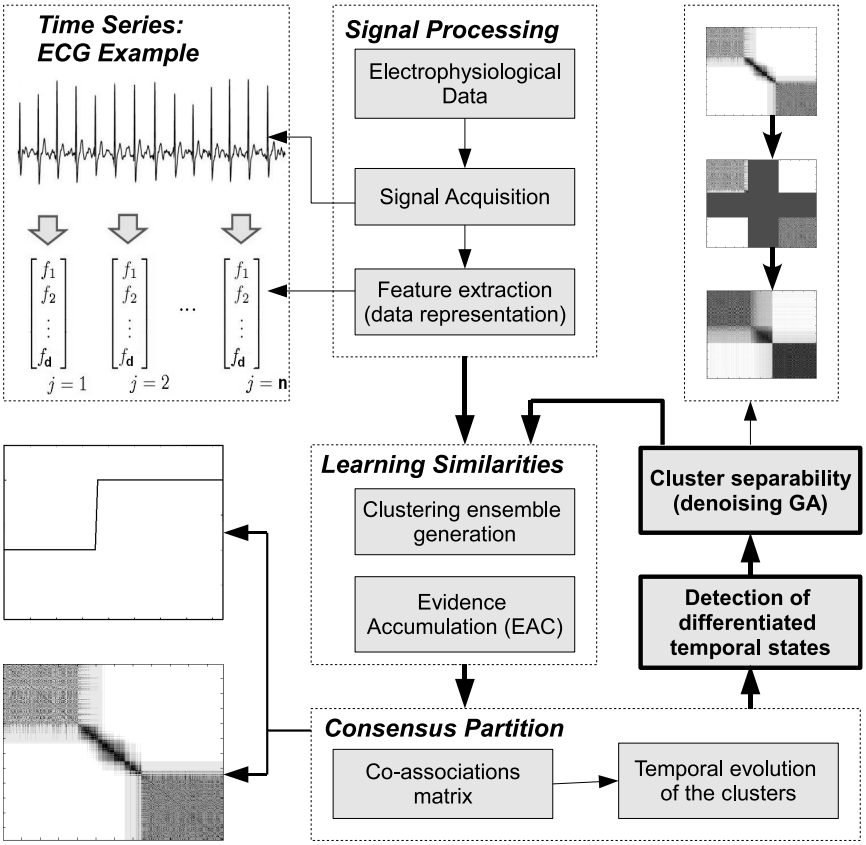


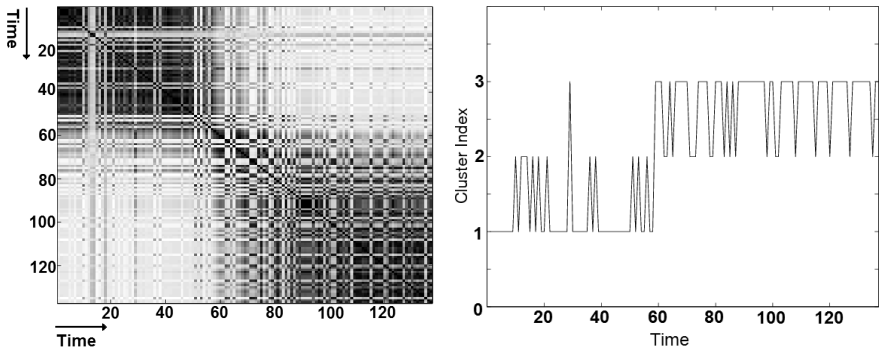
Fig. 1. Proposed work methodology for analysis of temporal series. The time series illustrated corresponds to ECG signals.

of patterns in the same cluster as votes for their association, the N data partitions of n patterns are mapped into a $n \times n$ co-association matrix:

$$C(i,j) = \frac{n_{ij}}{N} \tag{1}$$

where n_{ij} is the number of times the pattern pair (i,j) is assigned to the same cluster among the N partitions.

Graphically, the clusters can be visualized in the representation of the co-association matrix: if contiguous patterns belong to the same cluster, then quadrangular shapes will be present in this representation [8]. A co-association matrix is illustrated in Fig. 2(a). The chosen color scheme ranges from white to black (grayscale), corresponding to the gradient of similarity. Pure black corresponds to the highest similarity. Given that our major goal is to test that the temporal evolution of emotional states corresponds to a temporal evolution of the analyzed signal, the graphical representation of the co-association matrix is a powerful tool to assess the relationships of signal samples ordered by instant of occurrence.



(a) Co-association matrix (b) Cluster evolution over time

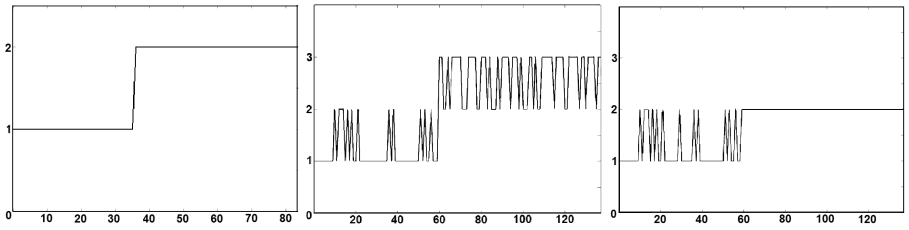
Fig. 2. Graphical representation examples: co-association matrix and temporal evolution of the clusters extracted from it

A consensus partition can be extracted from the co-association matrix by applying an hierarchical clustering method [2][8]. Hierarchical algorithms are either divisive or agglomerative based on whether the partitioning process is top-down or bottom-up [4]: agglomerative methods initially treat each pattern as a single cluster and will agglomerate these clusters based on proximity values, represented by a proximity matrix, while divisive methods assume initially that the entire data set is a single cluster [4]. These processes of agglomerating or dividing clusters are represented graphically by a dendrogram. A partition with k clusters is obtained by cutting the dendrogram at the k-th level. On the other hand, the decision on the number of clusters might be based on specific criteria, such as the cluster lifetime criterion: the k cluster lifetime is defined as the range of threshold values on the dendrogram that lead to the identification of k clusters [2].

An example of an extracted partition is depicted in Fig. 2(b), where the relationship between the temporally sequenced samples (x-axis) and the cluster to which they are assigned (y-axis), is plotted. It is possible to observe that cluster transitions generally occur between adjacent clusters: cluster 1 evolves to cluster 2, cluster 2 evolves between clusters 2 and 3, etc [8]. This is a meaningful result for the testing of the hypothesis of temporal evolution of emotional states.

2.2 Detection of Temporal States and Cluster Separability

The detection of temporal states is performed by comparing and examining the temporal evolution of clusters of one or more partitions produced from the learned similarity matrix. The goal of this analysis is the assessment of underlying structures that might correspond to the temporal evolution of differentiated states. The proposed criteria for this assessment are illustrated in Fig. 3. Each criterion considers sample segments of the temporal evolution of the clusters. Differentiated states are detected if: **(1)** there are segments such that all the samples of each segment belong to a single cluster (Fig. 3(a)); **(2)** each segment is comprised of samples belonging to different clusters, such that each



(a) Criterion 1 (b) Criterion 2 (c) Criterion 3

Fig. 3. Visualization of the proposed state detection criteria

Algorithm 1. State detection method

input : Cluster labels.
output: Consensus partition.

1 Split the n acquired samples into w windows $(L_1, L_2, ..., L_w)$ each with n samples.
2 Create the cluster indicator matrix M_b with w rows and i columns. $M_b(s, i) = 1$ if cluster label i is present in the L_s window. Repeat this procedure to all windows.
3 Identify meta-clusters by clustering over M_b with EAC, by examining the graphical representation of the state evolution of the consensus partition.

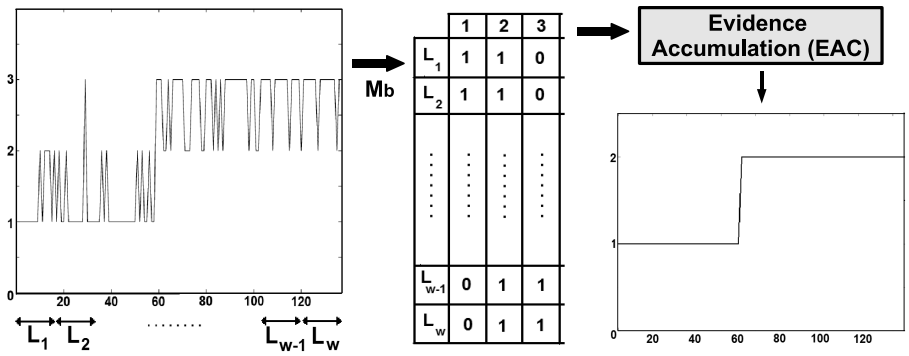
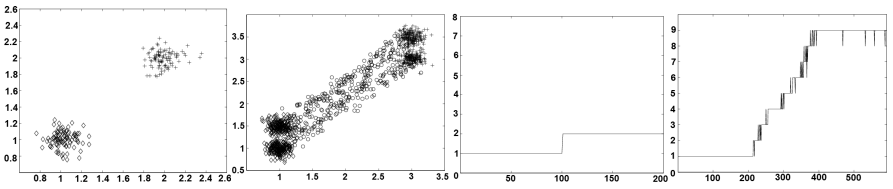


Fig. 4. State detection procedure

of those segments correspond to a unique combination of clusters (Fig. 3(b)) and **(3)** there are segments that correspond either to a single cluster or to a unique combination of clusters (Fig. 3(c)).

In order to identify states corresponding to distinct combinations of clusters, as per criteria 2 and 3, a meta-clustering procedure is used. Splitting the observation period (total duration of the acquired data) into adjacent windows (L_1, L_2, ..., L_w) each containing n samples, we define a cluster indicator $w \times k$ matrix, M_b. Each row of the matrix corresponds to a window, and each column to a cluster label, where the cluster combination for each window is expressed. Specifically, in each row, columns with the value 1 indicate the presence of the associated cluster in that window and so forth. This procedure is described in Algorithm 1, as well as in Fig.4.



(a) Data Set 1 (b) Data Set 2 (c) DS1: clustering (d) DS2: clustering

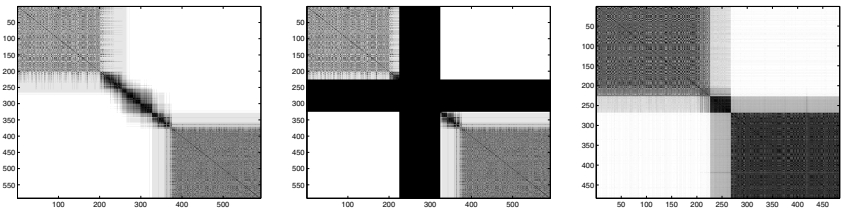
Fig. 5. Original synthetic data sets: feature values change abruptly between the clusters of DS1 (Fig. 5(a)) and feature values change smoothly over time for the samples of DS2 (Fig. 5(b)). Temporal evolution of the clusters obtained after applying EAC over K-Means partitions: the samples of DS1 are correctly partitioned but samples of DS2 corresponding to the transition between the two main clusters are wrongly assigned.

3 Genetic Algorithm for Denoising of Data with Temporal Evolution

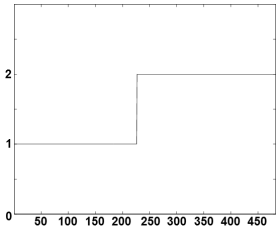
In this section we propose a genetic algorithm (GA) specifically designed to overcome the ambiguities induced by transient states present in signals characterized by temporal evolution. This GA is based on the assumption that, after removal from the original data set of the subset of samples that corresponds to transient states and performing EAC on the clustering ensemble based on this new reduced data set, a structure of separate states might emerge from the respective co-association matrix.

Fig. 5 illustrates the difficulty of clustering temporal data with smooth transitions. In these examples, data is represented by 2-D feature vectors. In data set 1 (DS1, see Fig. 5(a)), each state is modeled by a gaussian distribution, the transition between the two states (well separated mean values) occurring abruptly in time. Data set 2 (DS2, see Fig. 5(b)) illustrates a smooth evolution between states, each being composed by a mixture of two gaussians, where the mixtures of gaussians are quite distinct at the initial (left) and final (right) time periods; however the transitions between these occur smoothly. The clustering results obtained after applying Evidence Accumulation over K-means [3] produced partitions from both synthetic data sets are also depicted in Fig. 5. The partition found for the first data set represents truthfully the two states, (Fig. 5(c)); however the clustering method fails to label correctly the samples of the second data set due to smooth transitions between its groups of samples (Fig. 5(d)). It is not possible to assert, with confidence, where the first state ends and where the second state begins.

Fig. 6 illustrates the testing of the existence of well separated states, each corresponding to a single cluster, by denoising with a task specific genetic algorithm. Fig. 6(a) represents the co-associations matrix that corresponds to the original data set DS2; Fig. 6(b) illustrates the group of samples (marked pure black) to be removed in order to obtain separated quadrangular blocks, with no transient structures between them; finally Fig. 6(c) illustrates the co-association matrix obtained from the new data set. The temporal evolution of the clusters obtained for the original and denoised data sets are shown in Fig. 5(d) and Fig. 6(d), respectively. Two well separated states may be observed in the temporal evolution of the clusters obtained from the denoised data set



(a) Original matrix (b) Samples to be removed (pure black) (c) Denoised matrix



(d) Cluster evolution after denoising the data

Fig. 6. Example of the application of denoising

(see Fig. 6(d)). The intermediate groups of samples to be removed are determined by applying a task-specific genetic algorithm (GA). Several operators and procedures must be declared in order to define a particular GA [9]. These operators were defined for the denoising GA as follows. The algorithm itself is summarized by Algorithm 2.

Representation. Each individual is a set of samples, obtained after removal of one or more subsets of intermediate samples from the original data set. The first pattern of each removed subset is called minimum limit, (l_{min}), and the last pattern is referred to as maximum limit, (l_{max}).

Fitness Function. The evaluation of the fitness value of each individual is comprised of two stages, each concerning a partial fitness value function.

1. Determine if two or more of the H partitions that are associated to the individual m are equal: if partitions P_1 and P_2 are equal, then $I(P_1, P_2) = 1$, else $I(P_1, P_2) = 0$. The fitness value associated to consensus partitions equality is given by F_1 such that

$$F_1 = \frac{1}{\binom{H}{2}} \sum_{i=1}^{H-1} \sum_{j=i+1}^{H} I(P_i, P_j) \tag{2}$$

2. Determine, for each of the H partitions associated to the individual, the degree of cluster separability between temporal segments. We define the following two segments: segment (A) comprised of all the samples that occur before l_{min}, and

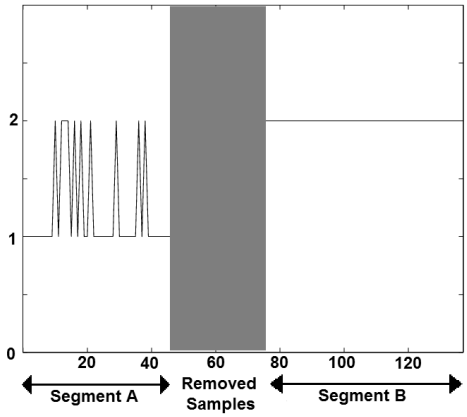


Fig. 7. Evaluation of clusters temporal separability. The subset of removed samples is marked gray.

segment (B) comprised of all the samples that occur after l_{max} (see Fig. 7). The subset of samples between l_{min} and l_{max} is the subset of removed samples from the original data set. For each segment, we determine the dominant cluster label. Samples with a different cluster label are considered outliers.

The fitness value associated to temporal separability is given by F_2:

$$F_2 = \frac{1}{H} \sum_{H} \frac{n_{samples} - n_{outliers}}{n_{samples}} \tag{3}$$

where $n_{samples}$ is the total number of samples associated to the evaluated individual. For the example shown in Fig. 7, the dominant cluster for segment A is cluster 1, with 9 outliers (that belong to cluster 2). The dominant cluster for segment B is cluster 2, with 0 outliers.

The final fitness value, F_{total}, for each individual is then

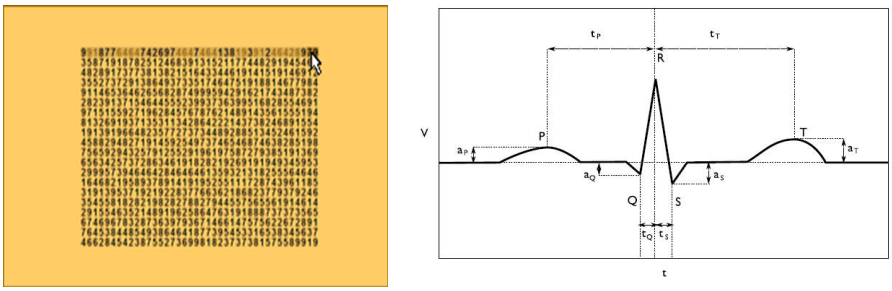
$$F_{total} = \alpha F_1 + (1 - \alpha) F_2 \tag{4}$$

where α is a weighting coefficient such that $\alpha \in [0, 1]$.

Selection. The selection of individuals for recombination is based on their fitness values by employing deterministic tournament selection (see [9]). Two individuals are selected for each selection step.

Recombination. Recombination of l_{min} and l_{max} of the selected individuals for generation of two new individuals with probability of occurrence p_r: the new l_{min} and l_{max} are chosen randomly as intermediate values of the selected individuals l_{min} and l_{max}, respectively.

Mutation. Modification of l_{min} and l_{max} of the new individuals by adding or subtracting to them a randomly chosen number, n_{mut}, with probability of occurrence p_m. Addition and subtraction have the same probability of occurrence.



(a) Concentration task (b) PQRST wave

Fig. 8. Stress inducing task (examination and annotation of a matrix with 800 numbers) and typical morphology of an ECG wave

4 Application Domain: Detection of Stress from ECG Signals

The temporal series analyzed correspond to electrocardiography, or ECG, signals. These signals are part of a more vast experience of multi-modal acquisition of physiological signals - the HiMotion project [1]. The ECG signals were acquired from a group of 24 subjects performing a stress inducing cognitive task, illustrated in Fig. 8(a). This task is a concentration test that consists of the identification and annotation of pairs of numbers that add to 10, by examination of the lines of a matrix of 20 lines per 40 columns of numbers [1]. The population of subjects is comprised of 18 males and 9 females, being their mean ages 23.4 years. For each one of these subjects, a montage with two electrodes called V_2 bipolar single lead electrocardiogram was used to collect signals from the heart [1]. Given that this concentration task is stress inducing, the presented methodology is applied to the ECG signals in order to assess the existence of stress states.

4.1 ECG Processing and Feature Extraction

An ECG signal is a recording of the electrical activity of the heart that consists of sequences of heart beats. Each heart beat has a typical morphology which consists of five waves (P, Q, R, S, T), schematically represented in Fig. 8(b).

From the acquired time series corresponding to the ECG signal, signal processing techniques were applied for signal segmentation [1] and a mean wave form was calculated based on 10 consecutive heart beats, to remove some spurious noise. All the waves were aligned with respect to the R wave. The recorded signal for each subject is then summarized in a temporal sequence of 137 mean waves, each wave represented by a feature vector [1].

The representation of the ECG signals is based on the P, Q, S, T waves. The R wave is used for time alignment, setting the initial instant of the beat ($t_R = 0$). The following rules are used to locate the position of each of the P, Q, S, T waves and to extract the eight main features of each mean wave [1], as depicted in Fig. 8(b):

Algorithm 2. Genetic algorithm for denoising of data characterized by temporal evolution of its features

input : Data to be clustered.
Clustering algorithm: *cAlgo*;
Number of partitions of each clustering ensemble: *nParts*;
H distinct hierarchical extraction algorithms;
Cluster extraction criterion;
Fitness threshold: *th*;
Number of generations: *G*.
Number of individuals of each population: *M*.
output: Denoised representation of the data, solution.

obtain an initial population from the original data set, pop$(g = 1)$*, of M individuals, randomly*;
while $g <= G$ **do**
  $currentPop = pop(g)$;
  foreach $m \in currentPop$ **do**
    cE $\leftarrow$ `clusteringEnsemble(`*m,cAlgo,nParts*`);`
    coAssocs $\leftarrow$ `EAC(`cE`);`
    for $h \leftarrow 1$ **to** H **do**
      $m(h).partition \leftarrow$ `extract(`coAssocs,h`);`
    end
    fitValue $\leftarrow$ `fitness(`m`);`
    if $fit >= th$ **then**
      solution $\leftarrow m$;
      `break`;
    end
    solution $\leftarrow m$: fitValue $=$ `maxFitnessValue(`*currentPop*`);`
  end
  $g \leftarrow g + 1$;
  $m \leftarrow 0$;
  while `sizePop(`$pop(g+1)$`)` $< M$ **do**
    parents $\leftarrow$ `select(`$currentPop$`);`
    (recIndOne, recIndTwo) $\leftarrow$ `recombine(`parents`);`
    (mutIndOne, mutIndTwo) $\leftarrow$ `mutate(`recIndOne,recIndTwo`);`
    `insert(`*pop*$(g+1)$,mutIndOne,mutIndTwo`);`
    `sizePop(`$pop(g+1)$`)` $\leftarrow$ `sizePop(`$pop(g+1)$`)`$+2$;
  end
end

1. t_P - the first maximum before the R wave;
2. a_P - the amplitude of the P wave;
3. t_Q - the first minimum before the R wave;
4. a_Q - the amplitude of the Q wave;
5. t_S - the first minimum after the R wave;
6. a_S - the amplitude of the S wave;
7. t_T - the first maximum after the R wave;
8. a_T - the amplitude of the T wave.

Table 1. Algorithmic parameters

Description	**Notation**	**Parameter Values**
Clustering Algorithms	$cAlgo1$ and $cAlgo2$	$k \in [2, 6]$ $\sigma \in [0.3, 0.4, ..., 2.9, 3.0]$ $nParts = 140$ (for each $cAlgo$)
Number of individuals	M	20
Number of partitions associated to each individual	H	5
Number of generations	G	20
Minimum threshold of fitness value	th	0.95
Fitness coefficient	α	0.1
Probability of recombination	p_r	0.9
Probability of mutation	p_m	0.1
Range of mutation values	n_{mut}	$n_{mut} \in [0, 5]$
Window size	n	10 samples

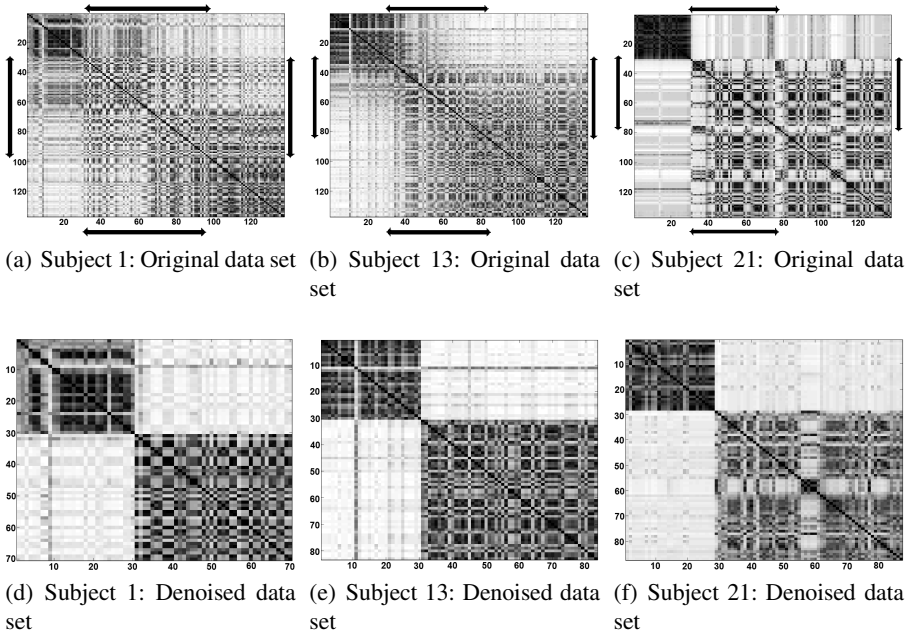
Each mean wave is represented by a 53-dimensional feature vector: the aforementioned 8 features, plus the amplitudes of the signal at 45 points of the signal obtained by re-sampling of the mean wave [1] . Thus, for each of the 24 subjects, there is a group of 137 temporally sequenced samples or patterns, corresponding each sample to a vector of 53 features.

4.2 Algorithmic Parameters and Experiments

Table 1 synthesizes the algorithms and algorithmic parameters employed. Two spectral clustering algorithms were used to produce clustering ensembles for each of the 24 data sets. These algorithms were originally proposed by Ng et al, [6], and Shi et al [7], and referred to in Table 1 by $cAlgo1$ and $cAlgo2$, respectively. Each partition of the clustering ensemble is generated such that it corresponds to a combination of possible values of the spectral algorithms parameters (which consist on the number of clusters, k and a scaling parameter σ).

Five agglomerative hierarchical methods were used for the consensus partition extraction from the co-associations matrix thus generated, using the cluster lifetime criterion: Single Link (SL), Complete Link (CL), Average Link (AL), Ward's Link (WL) and Centroid's Link (CenL). Detailed descriptions and studies of these algorithms may be found for example in [3] or [10].

For the final step of the detection method described in Section 2.2, five different partitions are extracted with the cluster lifetime criterion (one for each of the hierarchical methods already mentioned), as well. The final state structure is chosen to be the one that the majority of the hierarchical methods finds.



(a) Subject 1: Original data set (b) Subject 13: Original data set (c) Subject 21: Original data set

(d) Subject 1: Denoised data set (e) Subject 13: Denoised data set (f) Subject 21: Denoised data set

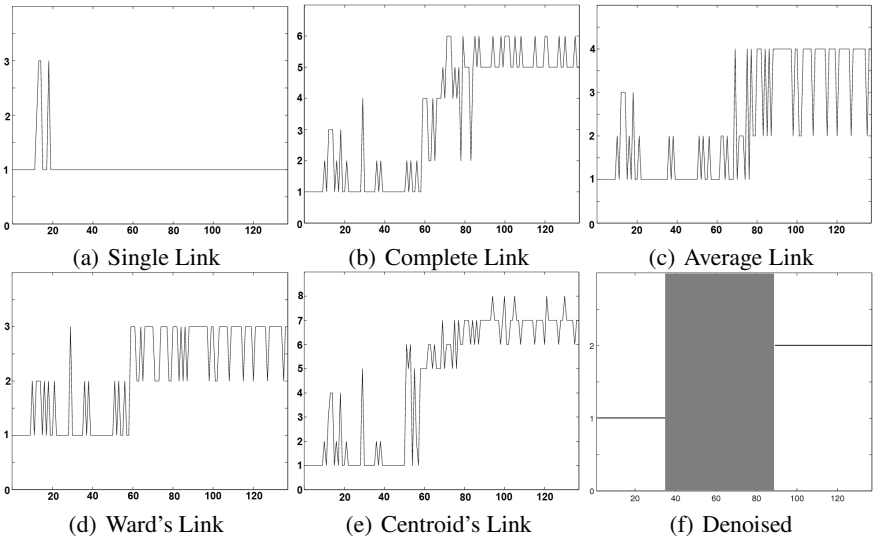
Fig. 9. Subjects 1, 13 and 21. The original data sets co-association matrices are depicted by Figs. 9(a),9(b) and 9(c), respectively. The denoised data sets co-association matrices are depicted by Figs. 9(d), 9(e) and 9(f), respectively. The samples removed from the original co-association matrices are within the area delimited by the arrows alongside the axes of Figs. 9(a) to 9(c).

4.3 Results and Discussion

Fig. 9 represents co-association matrices obtained for the original data set and for the denoised data set of subjects 1, 13 and 21, which show different levels of separability of the evolution into stress states.

By comparing the representations of both co-association matrices for the same subject, it is possible to observe that denoising of the original data sets will lead to the revelation of the structure of states in the ECG time series. Similarity relationships between contiguous samples are thus emphasized, which means that the clusters are separated such that a structure of differentiated states emerges, with no ambiguities in the observation and detection of these states.

This better separability of emotional states by the proposed GA-based method is further evaluated by observing the temporal evolution of clustering results produced from the learned similarities. Figs. 10(a) to 10(e) illustrates the temporal evolution of clusters obtained for the original data set of subject 6 (each partition corresponds to one of the five hierarchical methods used for combined partition extraction). By inspection of these 5 representations, we observe that different methods extract different partitions, in terms of number of clusters and samples assigned to each cluster. Though a structure of the data appears to be present, the transitions between clusters induce ambiguities in the observation of differentiated states.



(a) Single Link (b) Complete Link (c) Average Link

(d) Ward's Link (e) Centroid's Link (f) Denoised

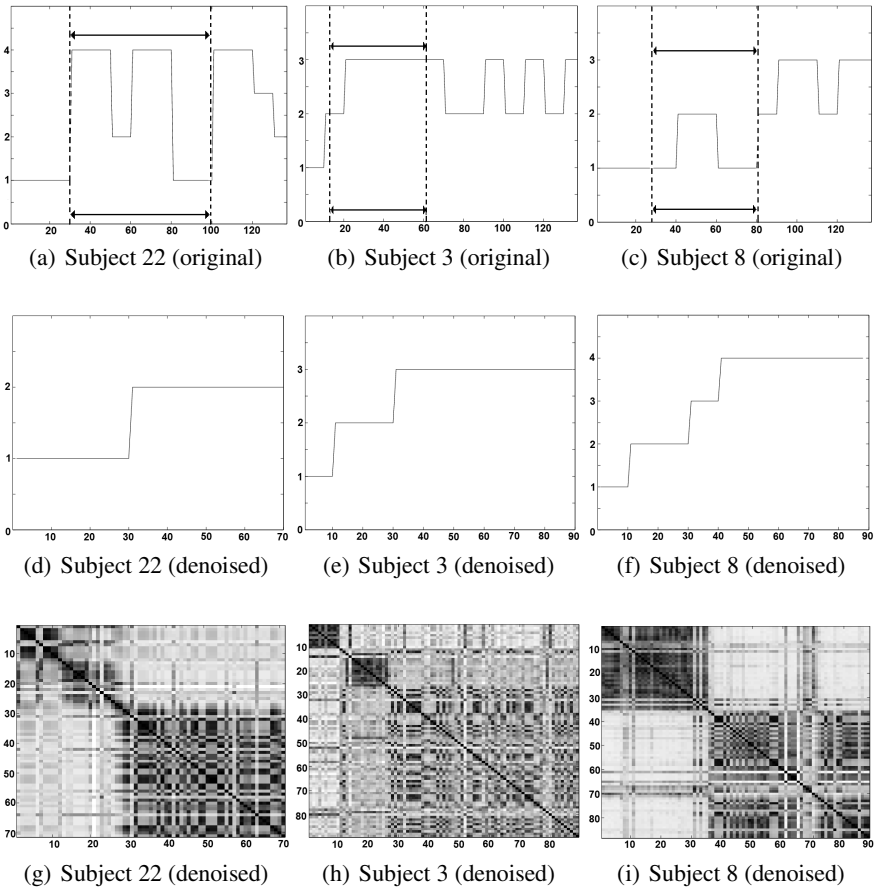
Fig. 10. Subject 6: original data set partition (Figs. 10(a) to 10(e)) and denoised data set partition (Fig. 10(f)), where the removed samples are marked gray. The x-axis represents the temporally ordered ECG samples and the y-axis the clusters to which they are assigned.

After applying the denoising GA the five methods extract the same partition of the data (depicted in Fig. 10(f)). This partition reveals two completely separated clusters each corresponding to a different emotional state. Thus, these results validate the observations of the original data set and it is possible to conclude that emotional states are observable in the ECG temporal series of subject 6.

Before denoising the data sets, the state detection method described (see Section 2.2) produces two or three states for 10 of the 24 subjects. The structures obtained have many transitions between clusters, which induces uncertainty in the observations of these results. Only in 3 of these 10 structures can we observe distinct state structures with no ambiguity.

For the remaining subjects, it is not clear how many states exist, nor which samples belong to each states (i.e., where does one state ends and the next begins). Figs. 11(a) to 11(c) depict structures found for 3 of such subjects: Fig. 11(a) refers to subject 22, Fig. 11(b) refers to subject 3, and Fig. 11(c) refers to subject 8. In all of these structures we observe several transitions between clusters.

After applying the denoising GA the results obtained by the state detection method reveal, with no ambiguities, the existence of distinct states for 18 subjects. We observe three typical structure types, depicted in Figs. 11(d) to 11(f): two distinct states, found for 5 subjects (of which subject 22 is an example, as shown in Fig. 11(d)), three distinct states, found for 11 subjects (of which subject 3 is an example, as shown in Fig. 11(e)), and four distinct states found for two subjects (of which subject 8 is an example, as shown in Fig. 11(f)). The duration of each state differs between subjects. The amount of samples removed by the denoising GA is also different for each individual, ranging from 31% to 59% of the total samples.



(a) Subject 22 (original) (b) Subject 3 (original) (c) Subject 8 (original)

(d) Subject 22 (denoised) (e) Subject 3 (denoised) (f) Subject 8 (denoised)

(g) Subject 22 (denoised) (h) Subject 3 (denoised) (i) Subject 8 (denoised)

Fig. 11. State detection: three types of structures are observable in the denoised data sets (Figs. 11(d) to 11(f)). These structures were detected in partitions extracted from the denoised co-association matrices represented in Figs. 11(g) to 11(i). The structures obtained with the detection method for the original data sets have more transitions between clusters, thus inducing uncertainty in the observation of differentiated states. The removed subsets of samples are delimited by dashed lines (Figs.11(a) to 11(c)).

5 Conclusions

In this work we proposed a methodology for the analysis of data characterized by temporal evolution, such as electrophysiological signals. This methodology is based on a clustering ensemble method, and on a genetic algorithm for assessment of the existence of differentiated states in time series. The presented results pertain to the application of the proposed techniques on ECG temporal series acquired during the performance of a cognitive task. These results validate our assumption that it is possible to infer the existence of differentiated emotional states in these signals by using the proposed methodology.

Ongoing work consists on a further extensive validation of this methodology in the herein presented application domain, as well as extrapolation to the automatic analysis of other time series, such as EEG data.

Acknowledgements. We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). This work was partially supported by the Portuguese Foundation for Science and Technology (FCT), Portuguese Ministry of Science and Technology, under grant PTDC/EIA CCO/1032230/ 2008.

References

1. Gamboa, H.: Multi-Modal Behavioral Biometrics Based on HCI and Electrophysiology. PhD Thesis, Instituto Superior Técnico (2008), http://www.lx.it.pt/~afred/pub/thesisHugoGamboa.pdf
2. Fred, A.L.N., Jain, A.K.: Combining multiple clusterings using evidence accumulation. IEEE Trans. Pattern Anal. Mach. Intell. 27, 835–850 (2005)
3. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, Inc., Englewood Cliffs (1988)
4. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Computing Surveys 31(3), 264–323 (1999)
5. Xu, R., Wunsch, D.: Survey of Clustering Algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)
6. Ng, A.Y., Jordan, M.I., Weiss, Y.: Clustering: Analysis and an Algorithm. In: Advances in Neural Information Processing Systems, vol. 14. MIT Press, Cambridge (2002)
7. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 22(8), 888–905 (2000)
8. Lourenço, A., Fred, A.L.N.: Unveiling Intrinsic Similarity - Application to Temporal Analysis of ECG. Biosignals (2), 104–109 (2008); INSTICC - Institute for Systems and Technologies of Information, Control and Communication
9. Sumathi, S., Hamsapriya, T., Surekha, P.: Evolutionary Intelligence. Springer, Heidelberg (2008)
10. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008), http://nlp.stanford.edu/IR-book/html/htmledition/irbook.html

[15] - (SIMBAD Technical Report n. 2011_08)

Aidos, H., Fred, A.L.N.: On the distribution of dissimilarity increments. In Vitrià, J., Sanches, J.M., Hernández, M., eds.: Pattern Recognition and Image Analysis. Volume 6669 of Lecture Notes in Computer Science. Springer (2011) 192–199 Iberian Conference on Pattern Recognition and Image Analysis - IbPRIA 2011, Las Palmas de Gran Canaria, Spain.

On the Distribution of Dissimilarity Increments

Helena Aidos and Ana Fred

Instituto de Telecomunicações, Instituto Superior Técnico, Lisbon, Portugal
{haidos, afred}@lx.it.pt

Abstract. This paper proposes a statistical model for the dissimilarity changes (increments) between neighboring patterns which follow a 2-dimensional Gaussian distribution. We propose a novel clustering algorithm, using that statistical model, which automatically determines the appropriate number of clusters. We apply the algorithm to both synthetic and real data sets and compare it to a Gaussian mixture and to a previous algorithm which also used dissimilarity increments. Experimental results show that this new approach yields better results than the other two algorithms in most datasets.

Keywords: clustering, dissimilarity increments, likelihood ratio test, Gaussian mixture

1 Introduction

Clustering techniques are used in various application areas, namely in exploratory data analysis and data mining [4]. Also known as unsupervised classification of patterns into groups (clusters), the aim is to find a data partition such that patterns belonging to the same cluster are somehow "more similar" than patterns belonging to distinct clusters. Clustering algorithms can be partitional or hierarchical, and can use a multitude of ways to measure the (dis)similarity of patterns [4, 7].

Partitional methods assign each data pattern to exactly one cluster; the number of clusters, K, is usually small, and often set *a priori* by the user, as a design parameter. Otherwise, the choice of K may be addressed as a model selection problem. The most iconic partitional algorithm is also the most simple: K-means, using the centroid as cluster representative, attempts to minimize a mean-square error criterion based on the Euclidean distance as measure of pairwise dissimilarity [7]. Also, common methods to estimate probability density functions from data, such as Gaussian mixture decomposition algorithms [1], can also be used as clustering techniques.

Hierarchical methods, on the other hand, yield a set of nested partitions which is graphically represented by a dendrogram. A data partition is obtained by cutting the dendrogram at a certain level. Linkage algorithms, such as the single-link and the complete-link [4], are the most commonly used.

Fred and Leitão [3] have proposed a hierarchical clustering algorithm using the concept of *dissimilarity increments*. These increments, which are formally defined in Section 2, use three data patterns at a time, and therefore yield information that goes beyond pairwise dissimilarities. Fred and Leitão showed empirical evidence suggesting

that dissimilarity increments vary smoothly within a cluster, and proposed an exponential distribution as statistical model governing the dissimilarity increments in each cluster [3]. They also noted that abrupt changes in the increments values means that the merging of two well separated clusters should not occur.

In this paper we propose a novel dissimilarity increments distribution (DID), supported on a theoretical-based analytical derivation for Gaussian data in $\mathbb{R}^2$. We use this distribution to construct a partitional clustering algorithm that uses a split&merge strategy, which iteratively accepts or rejects the merging of two clusters based on the distribution of their dissimilarity increments. We apply this algorithm to 6 synthetic data sets and 5 real data sets using as starting condition the clusters yielded by a Gaussian mixture algorithm proposed by Figueiredo and Jain [1], although any Gaussian mixture algorithm could be used instead.

This paper is structured as follows: Section 2 presents the derivation of the dissimilarity increments distribution (DID), and in Section 3 we propose a rewriting of the latter that depends on a single parameter: the expected value of increments. In Section 4, we show how to use this DID in a clustering algorithm. We present, in Section 5, the results of the proposed algorithm for 6 synthetic data sets with different characteristics (gaussian clusters, non-gaussian clusters, arbitrary shape clusters and densities) and 5 real data sets from the UCI Machine Learning Repository. These results are compared with the initial Gaussian mixture decomposition and with the hierarchical clustering algorithm proposed by Fred and Leitão in [3]. Conclusions are drawn in Section 6.

2 Dissimilarity Increments Distribution for 2D Gaussian Data

Consider a set of patterns, X. Given $\mathbf{x}_i$, an arbitrary element of X, and some dissimilarity measure between patterns, $d(\cdot,\cdot)$, let $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$ be the triplet of nearest neighbors, where $\mathbf{x}_j$ is the nearest neighbor of $\mathbf{x}_i$ and $\mathbf{x}_k$ is the nearest neighbor of $\mathbf{x}_j$ different from $\mathbf{x}_i$. The dissimilarity increment [3] between the neighboring patterns is defined as

$$d_{inc}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) = |d(\mathbf{x}_i, \mathbf{x}_j) - d(\mathbf{x}_j, \mathbf{x}_k)| \,. \tag{1}$$

Assume that $X \in \mathbb{R}^2$, and that elements of X are independent and identically distributed, drawn from a normal distribution, $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ. In this paper, the Euclidean distance is used as the dissimilarity measure, and our goal is to find the probability distribution of d_{inc}.

Let X^* be the result of an affine transformation of the patterns in X such that the transformed data has zero mean and the covariance matrix is a multiple of the identity matrix. Define $\mathbf{D}^* \equiv \mathbf{x}^* - \mathbf{y}^*$ as the dissimilarity in this space. Then the distribution of dissimilarities in this space becomes

$$(D^*)^2 \equiv \|\mathbf{x}^* - \mathbf{y}^*\|^2 = \sum_{i=1}^{2} \frac{(x_i^* - y_i^*)^2}{\Sigma_{ii}^*} \sim \chi^2(2),$$

where $\chi^2(2)$ is the chi-square distribution with 2 degrees of freedom, which is equivalent to an exponential distribution with parameter 1/2 [5].

Since the transformed data has circular symmetry, we have $\mathbf{D}^* = D^* \cos\alpha\, \mathbf{e}_1 + D^* \sin\alpha\, \mathbf{e}_2$, with $\alpha = \text{angle}(\mathbf{D}^*) \sim Unif([0, 2\pi[)$. Furthermore, $\mathbf{D} \equiv \mathbf{x} - \mathbf{y} = \sqrt{\Sigma^*_{11}}\, D^* \cos\alpha\, \mathbf{e}_1 + \sqrt{\Sigma^*_{22}}\, D^* \sin\alpha\, \mathbf{e}_2$, and

$$D^2 \equiv \|\mathbf{D}\|^2 = \underbrace{(\Sigma^*_{11} \cos^2\alpha + \Sigma^*_{22} \sin^2\alpha)}_{A(\alpha)^2} \underbrace{(D^*)^2}_{\|\mathbf{D}^*\|^2}, \tag{2}$$

where $A(\alpha)^2$ is called the expansion factor. Naturally this expansion factor will depend on the angle α. In practice it is hard to properly deal with this dependence. Therefore we will use the approximation that the expansion factor is constant and equal to the average value of the true expansion factor. We must find $\mathbb{E}[A(\alpha)^2]$, where $\alpha \sim Unif([0, 2\pi[)$ and $p_\alpha(\alpha) = \frac{1}{2\pi}$. After some computations, the expected value is given by

$$\mathbb{E}[A(\alpha)^2] = \int_0^{2\pi} p_\alpha(\alpha) A(\alpha)^2 \, \mathrm{d}\alpha = \frac{1}{2} \operatorname{tr}(\Sigma^*).$$

Under this approximation, the transformation equation (2) from the normalized space to the original space is $D^2 = \frac{1}{2} \operatorname{tr}(\Sigma^*)(D^*)^2$ and the probability density function of $D = d(\mathbf{x}, \mathbf{y})$ is (recall that $(D^*)^2 \sim \text{Exp}(1/2)$)

$$p_{_D}(z) = \frac{2z}{\operatorname{tr}(\Sigma^*)} \exp\left(-\frac{z^2}{\operatorname{tr}(\Sigma^*)}\right), \; z \in [0, \infty). \tag{3}$$

We can conclude that $D_1 = d(\mathbf{x}, \mathbf{y})$ and $D_2 = d(\mathbf{y}, \mathbf{z})$ follow the distribution in equation (3). The probability density function for $W = D_1 - D_2$ is given by the convolution

$$p_{_W}(w) = \int_{-\infty}^{\infty} \frac{4t(t+w)}{\operatorname{tr}(\Sigma^*)^2} \exp\left(-\frac{t^2 + (t+w)^2}{\operatorname{tr}(\Sigma^*)}\right) \mathbf{1}_{\{t \geq 0\}} \, \mathbf{1}_{\{t+w \geq 0\}} \, \mathrm{d}t. \tag{4}$$

Since we want to find the probability density function for the dissimilarity increments, we need to consider the probability density function of $|W| = d_{inc}$. Therefore, the probability density function for the dissimilarity increments is given by (derivations were omitted due to limited space)

$$\begin{aligned} p_{_{d_{inc}}}(w; \Sigma^*) = \frac{w}{\operatorname{tr}(\Sigma^*)} \exp\left(-\frac{w^2}{\operatorname{tr}(\Sigma^*)}\right) + \frac{\sqrt{\pi}}{\sqrt{2}\left(\operatorname{tr}(\Sigma^*)\right)^{3/2}} (\operatorname{tr}(\Sigma^*) - w^2) \times \\ \times \exp\left(-\frac{w^2}{2\operatorname{tr}(\Sigma^*)}\right) \operatorname{erfc}\left(\frac{w}{\sqrt{2\operatorname{tr}(\Sigma^*)}}\right), \end{aligned} \tag{5}$$

where $\operatorname{erfc}(\cdot)$ is the complementary error function.

3 Empirical Estimation of DID

The DID, as per equation 5, requires explicit calculation of the covariance matrix, Σ^*, in the transformed normalized space. In the sequel we propose data model fitting by

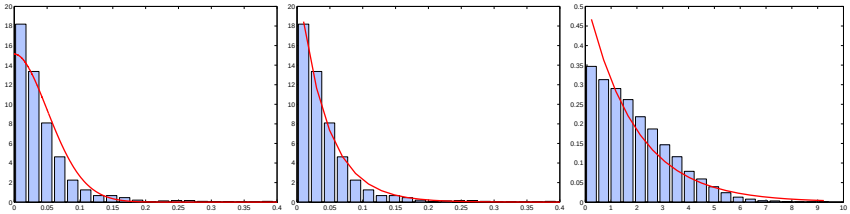
rewritting the distribution as a function of the mean value of the dissimilarity increments, $\lambda = \mathbb{E}[w]$. This is given by (after some calculation)

$$\lambda = \mathbb{E}[w] = \int_0^\infty w p_w(w)\, \mathrm{d}w = \frac{\sqrt{\pi}}{2} \left(\mathrm{tr}(\Sigma^*)\right)^{1/2} \left(2 - \sqrt{2}\right).$$

Hence, $\left(\mathrm{tr}(\Sigma^*)\right)^{1/2} = \frac{2\mathbb{E}(w)}{\sqrt{\pi}\left(2-\sqrt{2}\right)}$. Replacing in (5) we obtain an approximation for the dissimilarity increments distribution of a cluster that only depends of the mean of all the increments in that cluster:

$$p_{d_{inc}}(w;\lambda) = \frac{\pi\left(2-\sqrt{2}\right)^2}{4\lambda^2}\, w\, \exp\left(-\frac{\pi\left(2-\sqrt{2}\right)^2}{4\lambda^2}\, w^2\right) + \frac{\pi^2\left(2-\sqrt{2}\right)^3}{8\sqrt{2}\lambda^3} \times$$
$$\times \exp\left(-\frac{\pi\left(2-\sqrt{2}\right)^2}{8\lambda^2}\, w^2\right) \left(\frac{4\lambda^2}{\pi\left(2-\sqrt{2}\right)^2} - w^2\right) \mathrm{erfc}\left(\frac{\sqrt{\pi}\left(2-\sqrt{2}\right)}{2\sqrt{2}\lambda}\, w\right). \tag{6}$$

Figure 1 provides histograms for two data sets consisting of 1000 samples drawn from a Gaussian distribution in dimensions $M = 2$ and $M = 100$. As shown in figures 1(a) and 1(b), for $M = 2$ both the derived probability density function (6) and the exponential distribution suggested in [3] lead to a good fit to the histogram of the dissimilarity increments. However, as figure 1(c) shows, the latter provides a poor fit for high dimensions, such as $M = 100$, while the proposed distribution, even though derived for the 2D case, is much more adequate.



(a) M = 2, DID (b) M = 2, Exponential (c) M = 100, Exponential

Fig. 1. Histograms of the dissimilarity increments computed over M-dimensional Gaussian data and fitted distribution: (a) DID; (b) and (c) Exponential distribution.

4 Clustering using the Dissimilarity Increments Distribution

Let P^{init} be a partitioning of the data produced by a gaussian mixture decomposition. Then, each cluster in P^{init} follows a gaussian model; if $\mathbf{x}_i \in \mathbb{R}^2$, we are under the underlying hypothesis of the model presented in Section 2. One of the main difficulties of

the gaussian mixture decomposition is the inability to identify arbitrarily shaped clusters. We propose to overcome this intrinsic difficulty by a process of merging clusters using the previously derived model for dissimilarity increments. The decision to merge two clusters will depend on the dissimilarity increments distribution of each of the two clusters separately and of the two clusters combined.

We use the Mahalanobis distance [7] (which computes the distance between two gaussian distributions) to decide which clusters to test, by testing first clusters that are closer. The test we perform is a likelihood ratio test [6] consisting on the logarithm of the ratio between the likelihood of two separate clusters (two DID models) and the likelihood of the clusters together (single DID model). This likelihood ratio test is approximated by a chi-square distribution with one degree of freedom[1]. Therefore,

$$-2\log\left(\frac{p_{d_{inc}}(w;\lambda_1,\lambda_2)}{p_{d_{inc}}(w;\lambda_{12})}\right) \sim \chi^2(1). \tag{7}$$

Two clusters are merged if the p-value from the $\chi^2(1)$ distribution is less than a significance level α. This test is performed for all pairs of clusters until all the clusters that remain, when tested, are determined not to be merged. The overall procedure of this algorithm is summarized in algorithm 1.

Algorithm 1 GMDID

Input: 2-dimensional data, α
$P^{init} = \{C_1, \ldots, C_N\} \leftarrow$ data partition produced by a gaussian mixture decomposition
$D_{ij} \leftarrow$ Mahalanobis distance between clusters i and j
for all pairs (i, j) in ascending order of D_{ij} **do**
  $p_i \leftarrow$ DID for cluster i, $p_j \leftarrow$ DID for cluster j (eq. 6)
  $p_{ij} \leftarrow$ DID for cluster produced merging clusters i and j (eq. 6)
  p-value $\leftarrow$ Likelihood ratio test between $p_i p_j$ and p_{ij} (eq. 7)
  if p-value $< \alpha$ **then**
    merge clusters i and j
  else
    do not merge clusters i and j
  end if
end for
Return: $P = \{C_1, \ldots, C_K\} \leftarrow$ final data partition $K \leq N$

4.1 Graph-based Dissimilarity Increments Distribution

In order to choose the parameter α, we propose to use the Graph-based Dissimilarity Increments Distribution index, hereafter designated by G-DID, which is a cluster validity

[1] We have two parameters in the numerator – the expected value of the increments for each of the two clusters separately – and one parameter in the denominator – the expected value of the increments for the two clusters combined.

index proposed by Fred and Jain [2] based on the minimum description length (MDL) of the graph-based representation of partition P. The selection among N partitions, produced by different values of α, using the G-DID index, is as follows

$$\text{Choose } P^i : i = \underset{j}{\operatorname{argmin}}\{\text{G-DID}(P^j)\}, \tag{8}$$

where G-DID$(P) = -\log \hat{f}(P) + \frac{k_P}{2}\log(n)$ is the graph description length, and $\hat{f}(P)$ is the probability of partition P, with k_P clusters, according to a Probabilistic Attributed Graph model taking into account the dissimilarity increments distribution (see [2] for details). In [2], graph edge probability was estimated from an exponential model associated with each cluster, and the G-DID of the corresponding partition was used to select a design parameter of the hierarchical algorithm in [3]. For the selection of α for the GMDID algorithm describd above, we will use instead the DID model according to formula 6.

5 Experimental Results and Discussion

We can use any Gaussian mixture algorithm to obtain the conditions of the proposed clustering method; we chose the algorithm proposed by Figueiredo and Jain [1]. This method optimizes an expectation-maximization (EM) algorithm and selects the number of components in an unsupervised way, using the MDL criterion. With this Gaussian mixture algorithm we get a partition for the data set with as many clusters as gaussians.

To test the performance of the proposed method, we used 11 data sets: 6 synthetic data sets, and 5 real data sets from the UCI Machine Learning Repository[2]. The synthetic data sets were chosen to take into account a wide variety of situations: well-separated and touching clusters; gaussian and non-gaussian clusters; arbitrary shapes; and diverse cluster densities. These synthetic data sets are shown in figure 2. The *Wisconsin Breast-Cancer* data set consists of 683 patterns represented by nine features and has two clusters. The *House Votes* data set consists of votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the Congressional Quarterly Almanac. It is composed by two clusters and only the patterns without missing values were considered, for a total of 232 samples (125 democrats and 107 republicans). The *Iris* data set consists of three species of Iris plants (Setosa, Versicolor and Virginica). This data set is characterized by four features and 50 samples in each cluster. The *Log Yeast* and *Std Yeast* is composed of 384 samples (genes) over two cell cycles of yeast cell data. Both data sets are characterized by 17 features and consisting of five clusters corresponding to the five phases of the cell cycle.

We compared the proposed method (GMDID) to the Gaussian mixture (GM) algorithm [1] used to initialize the dissimilarity increments distribution, and to the method proposed by Fred and Leitão [3] based on Dissimilarity Increments (SL-AGLO). All these methods find the number of clusters automatically. GMDID and SL-AGLO have an additional parameter, so we compute partitions for several values of parameters. GMDID has a significance level α and we used 1%, 5%, 10% and 15% to decide whether

[2] http://archive.ics.uci.edu/ml

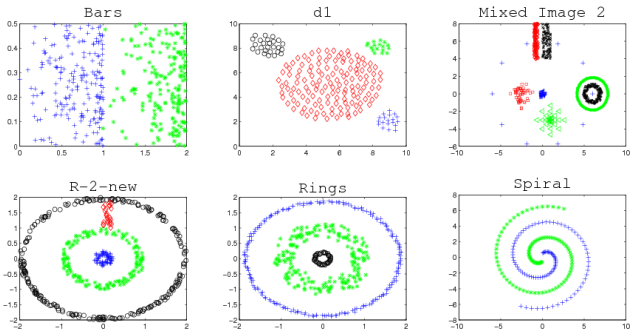


Fig. 2. Synthetic data sets

two clusters should be merged or not. The SL-AGLO algorithm has an isolation parameter which is a threshold set in the tail of the exponential distribution of the dissimilarity increments of a cluster (with parameter the inverse of the mean of the increments of that cluster). We used values ranging from the mean of the exponential distribution to 10 times this mean for this threshold, and the choice of the best value was also done using G-DID.

We assess the quality of each resulting partition P using the consistency index (CI), which is the percentage of correctly clustered patterns. Table 1 summarizes the results. The Gaussian mixture algorithm has problems finding the correct number of clusters, if the data sets are non-gaussians or can not be approximated by a single Gaussian. However, GMDID improved the results given by the Gaussian mixture, because it depends of the dissimilarity changes between neighboring patterns. If a cluster does not have a Gaussian behavior and the Gaussian mixture produces at least two gaussian components for that cluster, it may be possible that GMDID can find the cluster, for a certain statistical significance level, by merging those components together.

Despite the fact that the dissimilarity increments distribution proposed here is for 2-dimensional Gaussian distributions, we noticed that when the algorithm is applied to real data sets (which have dimensions higher than two) the results are still slightly better when compared to the Gaussian mixture and to SL-AGLO. In the future we will develop the dissimilarity increments distribution to a general M-dimensional data set, which should further improve these results.

6 Conclusions

We derived the probability density function for the dissimilarity increments under the assumption that the clusters follow a 2-dimensional Gaussian distribution. We applied this result by proposing a novel clustering approach based on that distribution. The application example presented here used the Gaussian mixture algorithm from Figueiredo and Jain [1], however any Gaussian mixture decomposition could be used. We showed that the proposed method is better or equally good when compared to the Gaussian mixture or to another method based also on dissimilarity increments.

Table 1. Consistency values of the partitions found by the three algorithms. The values in parenthesis correspond to the number of clusters found by each algorithm. The first two columns correspond to the number of patterns (N) and the true number of clusters (Nc) of each data set.

	N	Nc	GM	GMDID	SL-AGLO
Bars	400	2	0.4375 (12)	**0.9525** (2)	0.9050 (4)
d1	200	4	**1.0000** (4)	**1.0000** (4)	**1.0000** (4)
Mixed Image 2	739	8	0.4709 (20)	**1.0000** (8)	0.9743 (10)
R-2-new	500	4	0.2880 (29)	**0.7360** (8)	0.6160 (3)
Rings	450	3	0.2444 (27)	**1.0000** (3)	**1.0000** (3)
Spiral	200	2	0.1400 (27)	**1.0000** (2)	**1.0000** (2)
Breast Cancer	683	2	0.5593 (5)	**0.7467** (3)	0.5783 (24)
House Votes	232	2	**0.8103** (2)	**0.8103** (2)	0.6810 (4)
Iris	150	3	**0.8000** (4)	0.6667 (2)	0.4800 (6)
Log yeast	384	5	0.3281 (10)	**0.3594** (4)	0.3229 (4)
Std yeast	384	5	0.4349 (2)	0.4349 (2)	**0.5391** (7)

The proposed method has a condition that the data should consist of 2-dimensional Gaussian clusters. However, we presented results for real data sets with dimension higher than two and the algorithm performs reasonably well compared to others. In future work we will extend the increment distribution to a generic dimension M.

7 Acknowledgments

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). This work was partially supported by the Portuguese Foundation for Science and Technology (FCT), scholarship number SFRH/BD/39642/2007, and grant PTDC/EIA-CCO/103230/2008.

References

1. Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of finite mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(3), 381–396 (2002)
2. Fred, A., Jain, A.: Cluster validation using a probabilistic attributed graph. In: Proceedings of the 19th International Conference on Pattern Recognition (ICPR 2008) (2008)
3. Fred, A., Leitão, J.: A new cluster isolation criterion based on dissimilarity increments. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(8), 944–958 (2003)
4. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3), 264–323 (1999)
5. Johnson, N.L., Kotz, S., Balakrishnan, N.: Continuous Univariate Distributions, Applied Probability and Statistics, vol. 1. John Wiley & Sons Ltd., 2 edn. (1994)
6. Lehmann, E.L., Romano, J.P.: Testing Statistical Hypotheses. Springer Texts in Statistics, Springer, 3 edn. (2005)
7. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Elsevier Academic Press, 2 edn. (2003)

[14] - (no technical report)

Aidos, H., Fred, A.: Statistical modeling of dissimilarity increments for d-dimensional data: Application in partitional clustering. Submitted to Journal Pattern Recognition (2011)

Statistical Modeling of Dissimilarity Increments for d-dimensional Data: Application in Partitional Clustering

Helena Aidos, Ana Fred

Instituto de Telecomunicações, Instituto Superior Técnico, Lisbon, Portugal

Abstract

This paper addresses the use of high order dissimilarity models in data mining problems. We explore dissimilarities between triplets of nearest neighbors, called *dissimilarity increments* (DIs). We derive a statistical model of DIs for d-dimensional data (d-DID) assuming that the objects follow a multivariate Gaussian distribution. Empirical evidence shows that the d-DID is well approximated by the particular case $d = 2$. We propose the application of this model in clustering, with a partitional algorithm that uses a merge strategy on Gaussian components. Experimental results, in synthetic and real datasets, show that clustering algorithms using DID usually outperform well known clustering algorithms.

Keywords: dissimilarity increments, partitional clustering, likelihood ratio test, minimum description length, Gaussian mixture decomposition

Email addresses: haidos@lx.it.pt (Helena Aidos), afred@lx.it.pt (Ana Fred)

1. Introduction

Clustering has been applied in several areas like machine learning, pattern recognition, web mining, image segmentation, genetics, and biology [1, 2, 3]. The main goal of clustering is to arrange data objects in groups (clusters), such that objects belonging to the same cluster are similar. It is a form of unsupervised learning, since no information about the groups to which the objects belong is known *a priori*. Two major clustering strategies have been adopted in published methods: partitional and hierarchical [4, 1, 5]. Hierarchical clustering techniques group objects with a sequence of nested partitions, either from singleton clusters to a cluster including all data (agglomerative strategy) or in the opposite way (divisive strategy), while partitional clustering techniques divide the data into clusters without the hierarchical structure. For an overview of clustering techniques see [4, 1, 5, 6].

Partitional methods put each data point into exactly one cluster. Often, the user must set the number of clusters, k, beforehand, and k is usually small. On the other hand, the choice of k can be considered itself a model selection problem [7] which is often non-trivial, especially for real-world datasets. One important class of partitional methods is the one of prototype-based methods, such as k-means [6] (with an associated minimum squared error criterion; it is the simplest and most widespread clustering algorithm), iterative self-organizing data analysis technique (ISODATA) [8], k-medoids [3] and squared-error clustering [9], which can work very well for compact and hyperspherical clusters. Another class of partitional methods is the one of parametric density approaches, including methods that estimate probability density functions from data, such as Gaussian mixture decompo-

sition algorithms [10, 11, 12].

Hierarchical methods produce a set of nested partitions in a hierarchical structure according to the proximity matrix; this structure is graphically represented by a dendrogram. Agglomerative methods start by considering each data point as one cluster, and each partition is obtained from the previous one by merging two clusters into a single cluster. Methods in this class include single-link, complete-link, average-link, median-link, centroid-link, weighted-link, Ward link [4], and more recent hierarchical algorithms for handling large-scale datasets such as CURE [13], ROCK [14], Chameleon [15] and BIRCH [16]. Divisive methods work in the opposite way: one starts with a single cluster with all the objects and a divisive procedure is applied repeatedly until all clusters are singletons. This class of methods is not very used in practice due to its computational cost: for a cluster with N objects, there are $2^{N-1} - 1$ possible divisions [1]. A drawback of most classical hierarchical techniques is the failure to identify clusters with arbitrary shapes and sizes, and the tendency to form spherical structures in the data. Most of the hierarchical methods are inspired in graph theory, such as single-link and complete-link. However, graph theory can also be used in a different kind of clustering algorithms: the clusters can be described in terms of weighted graphs. CLICK [17] is an example of this kind of methods.

Most of the clustering techniques require, implicitly or explicitly, a similarity measure between patterns, the choice of which is difficult to make if one has no prior knowledge about cluster shapes or struture. Most clustering algorithms use pairwise distances between patterns, the most typical one being the Euclidean distance. However, many other measures can be used,

such as the Mahalanobis distance [1, 5]. More recently, a new third order dissimilarity measure has been proposed [18], the *dissimilarity increments* (DIs), which are computed over triplets of nearest neighbor patterns. The fact that this measure uses three data points at a time gives more information about the patterns lying in the same cluster, since a smooth evolution of the DIs should occur if the patterns are in the same cluster, and high values should occur for patterns lying in different clusters [18].

Based on this new dissimilarity measure, a hierarchical clustering algorithm has also been proposed in [18]. The statistical model proposed for the DIs in each cluster, based on visual inspection, was the exponential distribution, with parameter equal to the inverse of the mean of the increments. In this paper we theoretically derive the dissimilarity increments distribution (DID) under some approximations, and empirically show that this new distribution is a better approximation to the empirical distribution of the DIs than the exponential one.

The novel DID is derived under the hypothesis of local Gaussian generative models for the data in $\mathbb{R}^d$, and is called d-DID. We particularize the model for $d = 2$, hereafter referred as 2-DID; using two statistical distance measures, we empirically show that 2-DID is a good approximation to d-DID, and that both are better approximations of the true DID than the exponential distribution. Using the 2-DID, we then construct a partitional clustering algorithm consisting of a merge strategy, which iteratively accepts or rejects the merging of two clusters based on this new distribution. In [19] we proposed a likelihood ratio test as the merge criterion, which merges pairs of clusters with a p-value less than a given significance level α. In this pa-

per we propose a new parameter-free merge criterion based on the Minimum Description Length principle.

This paper is structured as follows: Section 2 presents the derivation of the DID for d-dimensional data (d-DID), and we write this distribution as a function of a single parameter: the expected value of increments. In Section 3 we present the particular case of $d = 2$, and in Section 4 we show empirical evidence that the new distribution derived here is a better approximation to the empirical distribution than the one proposed in [18]. In Section 5 we show how to use this DID in a clustering algorithm, proposing two merge criteria: likelihood ratio test (LRT), presented in [19], and minimum description length (MDL). We present, in Section 6, the performance of the proposed algorithm on six synthetic datasets with different characteristics (Gaussian clusters, non-Gaussian clusters, arbitrary shape clusters and densities) and on eight real-world datasets from the UCI Machine Learning Repository and 20-Newsgroups. These results are compared with the initial Gaussian mixture decomposition (GMD), the hierarchical clustering algorithm proposed in [18] and with some traditional clustering algorithms (single-link, average-link, complete-link, Ward-link and k-means), when the true number of clusters is known. We also present a study of the proposed method when the number of clusters is not known *a priori*. Discussion and conclusions are in Sections 7 and 8, respectively. In the Appendix, we present the derivation details for the DID.

2. Dissimilarity Increments Distribution for *d*-dimensional Data (*d*-DID)

Consider a set of patterns X. Given $\mathbf{x}_i \in X$ and some dissimilarity measure between patterns, $d(\cdot, \cdot)$, let $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$ be a triplet of nearest neighbors, obtained as follows: $\mathbf{x}_j$ is the nearest neighbor of $\mathbf{x}_i$, and $\mathbf{x}_k$ is the nearest neighbor of $\mathbf{x}_j$ different from $\mathbf{x}_i$. The *dissimilarity increment* (DI) [18] between these patterns is defined as

$$d_{inc}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) = |d(\mathbf{x}_i, \mathbf{x}_j) - d(\mathbf{x}_j, \mathbf{x}_k)| \,. \tag{1}$$

In the following subsections we will derive the probability density function (PDF) for the DIs, using the Euclidean distance as the dissimilarity measure.

2.1. Derivation of the DID Model

Assume that X is a d-dimensional set of patterns (henceforth called a *cluster*), and that its elements are independent and identically distributed according to a multivariate Gaussian distribution, $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$. With no loss of generality, we assume that the mean of the distribution of the elements of X is $\boldsymbol{\mu} = 0$ and that its covariance matrix Σ is diagonal (this only involves translation and rotation of the data set, which does not affect the Euclidean distances). If $\mathbf{x}$ denotes a sample from this Gaussian, we define the sphered data $\mathbf{x}^*$ as having its i-th coordinate given by $x_i^* \equiv x_i/\Sigma_{ii}$ (this transformation is commonly called "whitening" or "sphering"); x_i^* thus follows the standard normal distribution, $\mathcal{N}(0, 1)$. It is clear that the difference between samples from two univariate standard normal distributions follows a normal distribution with covariance 2. It can be shown that the squared

Euclidean distance, $(D^*)^2 = \sum_{i=1}^{d} (z_i^*)^2$, where $z_i^* \equiv \frac{x_i^* - y_i^*}{\sqrt{2}} \sim \mathcal{N}(0,1)$, follows a chi-square distribution with d degrees of freedom [20]. Therefore, the PDF for $(D^*)^2$ is given by:

$$p_{(D^*)^2}(x) = \frac{2^{-d/2}}{\Gamma(d/2)}\, x^{d/2-1} \exp\left(-\frac{x}{2}\right),\ x \in [0, +\infty[. \tag{2}$$

Furthermore, after the sphering, the transformed data has circular symmetry in $\mathbb{R}^d$. We define angular coordinates in a $(d-1)$-sphere, with $\theta_i \in [0, \pi[, i = 1, \ldots, d-2$ and $\theta_{d-1} \in [0, 2\pi[$. Define $\mathbf{D} \equiv \mathbf{x} - \mathbf{y} \equiv (b_1, b_2, \ldots, b_d)$, where b_i can be expressed in terms of polar coordinates as

$$\begin{aligned}
b_1 &= \sqrt{2\Sigma_{11}}\, D^* \cos\theta_1 \\
b_i &= \sqrt{2\Sigma_{ii}}\, D^* \left[\prod_{k=1}^{i-1} \sin\theta_k\right] \cos\theta_i, i = 2, \ldots, d-1 \\
b_d &= \sqrt{2\Sigma_{dd}}\, D^* \left[\prod_{k=1}^{d-1} \sin\theta_k\right].
\end{aligned}$$

The squared Euclidean distance in the original space is

$$\begin{aligned}
D^2 &= 2\left[\Sigma_{11}\cos^2\theta_1 + \sum_{i=2}^{d-1} \Sigma_{ii} \left(\prod_{k=1}^{i-1} \sin^2\theta_k\right) \cos^2\theta_i + \Sigma_{dd}\left(\prod_{k=1}^{d-1}\sin^2\theta_k\right)\right](D^*)^2 \\
&\equiv 2A(\Theta)(D^*)^2,
\end{aligned} \tag{3}$$

where $A(\Theta)$, with $\Theta = (\theta_1, \theta_2, \ldots, \theta_{d-1})$, is called the *expansion factor*. Naturally, this expansion factor will depend on the angle vector Θ. In practice, it is hard to properly deal with this dependence; therefore, we will use the approximation that the expansion factor is constant and equal to the expected value of the true expansion factor over all angles Θ. This expected value, which is given by

$$\mathbb{E}[A(\Theta)] = \frac{\pi^{-d/2+1}}{2\Gamma(1+\frac{d}{2})}\, \eta, \tag{4}$$

where $\eta \equiv \mathrm{tr}(\Sigma)$ (see Appendix A for the derivation).

Under this approximation, the transformation equation (3) from the normalized space to the original space is given by

$$D^2 = \frac{\pi^{-d/2+1}}{\Gamma(1+d/2)}\,\eta\,(D^*)^2. \tag{5}$$

From (2) and (5) one can obtain the PDF of D^2, and from there one can obtain the PDF of $D = d(\mathbf{x}, \mathbf{y})$ as

$$p_{_D}(w) = 2G_d(\eta)w^{d-1}\exp\left(-C_d(\eta)w^2\right),\ w \in [0, +\infty[, \tag{6}$$

where we define $G_d(\eta) \equiv d^{d/2}\Gamma(d/2)^{d/2-1}2^{-d}\eta^{-d/2}\pi^{d/2(d/2-1)}$ and $C_d(\eta) \equiv d\Gamma(d/2)(4\eta)^{-1}\pi^{d/2-1}$.

The dissimilarity increment is defined as the absolute value of the difference of two Euclidean distances. We have just derived the PDF of the Euclidean distance between two patterns; therefore, the PDF of the DIs can be shown to be

$$p_{_{d_{inc}}}(w;\eta) = \frac{G_d(\eta)^2}{2^{d-5/2}C_d(\eta)^{d-1/2}}\exp\left(-\frac{C_d(\eta)}{2}\,w^2\right)\left[\sum_{k=0}^{d-1}\sum_{i=0}^{2d-2-k}(-1)^i\binom{d-1}{k}\right.$$
$$\left.\binom{2d-2-k}{i}w^{k+i}2^{k/2-i/2}C_d(\eta)^{k/2+i/2}\Gamma\left(\frac{2d-1-k-i}{2}, \frac{C_d(\eta)}{2}\,w^2\right)\right], \tag{7}$$

where $\Gamma(a, x)$ is the incomplete gamma function [21] (see Appendix B for the derivation).

2.2. Empirical estimation

The DID in equation (7) requires explicit knowledge of the diagonal covariance matrix, Σ. In the following, we propose fitting the data model

by rewriting the distribution as a function of the mean value of the DIs, $\lambda = \mathbb{E}[w]$. In other words, we will write η as a function of λ. This is given by

$$\eta = \lambda^2 Q_d^2 \left[\sum_{k=0}^{d-1} \sum_{i=0}^{2d-2-k} (-1)^i \binom{d-1}{k} \binom{2d-2-k}{i} 2^k \right. \\ \left. B_{\frac{1}{2}} \left(\frac{k+i}{2} + 1, d - \frac{k+i+1}{2} \right) \right]^{-2}, \tag{8}$$

where $Q_d \equiv 2^{d-7/2} d^{1/2} \pi^{d/4-1/2} \Gamma(d/2)^{5/2} \Gamma(d+1/2)^{-1}$ and $B_x(a,b)$ is the incomplete beta function [21] (see Appendix C for details). Plugging (8) into (7) we obtain an approximation, for the DID of a cluster, $p_{d_{inc}}(w;\lambda)$, that only depends on the mean of all increments in that cluster.

3. Dissimilarity Increments Distribution for 2-dimensional Data (2-DID)

We now present the particular case $d = 2$. Consider a 2-dimensional set of patterns in the same conditions as in Section 2. If we replace $d = 2$ in equation (7), we get

$$p_{d_{inc}}(w;\eta) = \frac{w}{2\eta} \exp\left(-\frac{w^2}{2\eta}\right) + \frac{\sqrt{\pi}}{2\eta^{3/2}} \left(\eta - \frac{w^2}{2}\right) \exp\left(-\frac{w^2}{4\eta}\right) \operatorname{erfc}\left(\frac{w}{2\sqrt{\eta}}\right), \tag{9}$$

where $\operatorname{erfc}(\cdot)$ is the complementary error function. The empirical estimation based on the expected value of the increments is obtained by replacing d with 2 in equation (8), which gives

$$\eta = \frac{4\lambda^2}{2\pi\beta^2},$$

with $\beta \equiv \left(2 - \sqrt{2}\right)$.

Replacing in (9) we obtain an approximation for the DID of a cluster that only depends of the mean of all the increments in that cluster:

$$p_{d_{inc}}(w;\lambda) = \frac{\pi\beta^2}{4\lambda^2} w \exp\left(-\frac{\pi\beta^2}{4\lambda^2}w^2\right) + \frac{\pi^2\beta^3}{8\sqrt{2}\lambda^3} \exp\left(-\frac{\pi\beta^2}{8\lambda^2}w^2\right) \times \left(\frac{4\lambda^2}{\pi\beta^2} - w^2\right) \operatorname{erfc}\left(\frac{\sqrt{\pi}\beta}{2\sqrt{2}\lambda}w\right). \quad (10)$$

4. DID Models and Data Fitting

In this section, our goal is to show that one can approximate the DID for any dimension using the particular case $d = 2$. This is an important result, because using 2-DID instead of d-DID saves considerable computation time for large d values.

We will show this empirically, using simulated data. We generate datasets with 2000 patterns in d dimensions, with d from 2 to 25. Without loss of generality, the Gaussians are all centered in the origin and the covariance matrices are diagonal, with elements randomly chosen between 0 and 1.

4.1. Cramér-von-Mises Criterion

The Cramér-von-Mises criterion [22] is a criterion used to determine how well a cumulative distribution function F fits a given empirical cumulative distribution function F_n. It is defined as

$$\omega^2 \equiv \int_{-\infty}^{\infty} \left[F_n(x) - F(x)\right]^2 \, \mathrm{d}F(x). \quad (11)$$

Let $x_1, x_2, \ldots, x_n$ be the empirically observed values, in nondecreasing order; in [22] it is shown that

$$T \equiv n\omega^2 = \frac{1}{12n} + \sum_{i=1}^{n} \left[\frac{2i-1}{2n} - F(x_i)\right]^2. \quad (12)$$

For common distributions, one should use tables of values of T calculated for the distribution F. We do not know the tables for the d-DID or 2-DID distributions, so we will simply compare the values of T directly: smaller values mean that the considered distribution is closer to the empirical distribution.

4.2. Jensen-Shannon Divergence

The Jensen-Shannon divergence [23] is a measure of the similarity between two probability distributions P and Q. It is similar to the Kullback-Leibler divergence but it is always finite, and is symmetric. It is defined as

$$JSD(P,Q) \equiv \frac{1}{2}D_{KL}(P,M) + \frac{1}{2}D_{KL}(Q,M), \tag{13}$$

where $M = \frac{1}{2}(P+Q)$ and $D_{KL}(P,M) = \int P(x) \log \frac{P(x)}{M(x)} \mathrm{d}x$ is the Kullback-Leibler divergence.

Two probability distributions are similar if the Jensen-Shannon divergence has a small value. The higher the value, the more dissimilar the distributions are.

4.3. Best approximation to d-DID

The d-DID expression (7) is not only computationally heavy, but also involves the computation of exponentials of very large numbers (of the order of 1000 for $d = 50$), which are troublesome to handle numerically. In this part of the paper we empirically show that 2-DID is a good approximation to d-DID, and also that both distributions are a much better approximation to the real distribution than the exponential distribution considered in [18].

Analysis of Fig. 1 (*left*) shows that, according to the Cramér-von-Mises criterion, the DID distributions presented in this paper are a better fit to

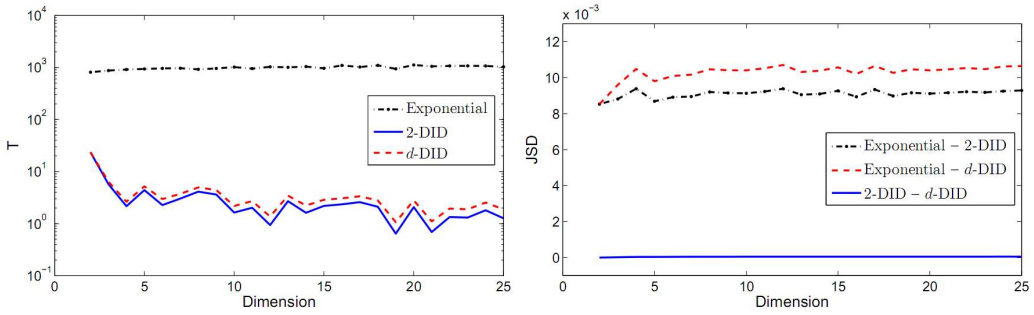


Figure 1: Empirical distribution comparison for Gaussian datasets with 2000 patterns in d dimensions, with d ranging from 2 to 25. *Left:* Cramér-von-Mises test (T) between each theoretical distribution (Exponential, 2-DID and d-DID) and the empirical distribution. *Right:* Jensen-Shannon divergence (JSD) between pairs of theoretical distributions (Exponential, 2-DID and d-DID).

the histogram of the real distribution than the exponential distribution – T is better by approximately $10^{2.5}$. Also, the 2-DID seems slightly better than the d-DID, although we suspect, from our experiments, that this is due to approximation errors that occur during the computation of the d-DID.

Figure 1 (*right*) compares pairs of distributions through the Jensen-Shannon divergence. We can conclude that 2-DID and d-DID are very similar to each other. This empirically shows that 2-DID is a good approximation to d-DID. From now on we will always use 2-DID, and will only refer to it as DID, because it is a very good approximation to d-DID, and is computationally much more manageable.

4.4. Fitting DID to Non-Gaussian Data

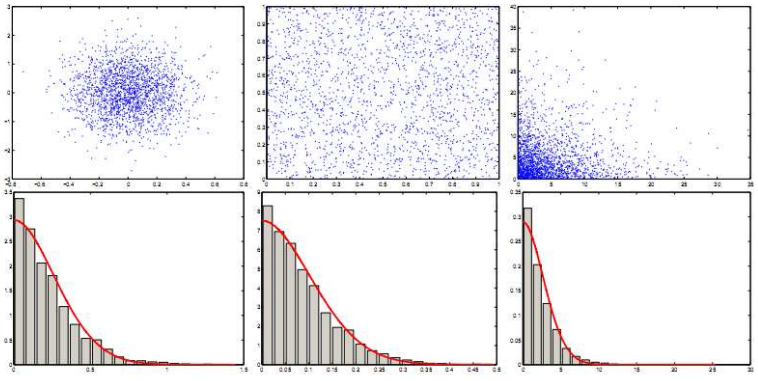
Although the underlying hypothesis of the DID that we derived is that the data comes from a cluster with a Gaussian distribution, the approximate distribution in equation (10) only depends of the mean of the increments

in that cluster. This allows us to use this distribution with non-Gaussian clusters. Although this could result in poor fits, we now empirically show that this does not happen for some common data distributions. We study datasets generated from several continuous and discrete distributions, namely Gaussian, Uniform, Exponential, Poisson, and Geometric distributions. For all the datasets we generate 2000 points in 20 dimensions. In Fig. 2 we see that the DID fits the histogram well for all the distributions that were tested. In fact, the values of T are of the same order of magnitude for all these distributions.

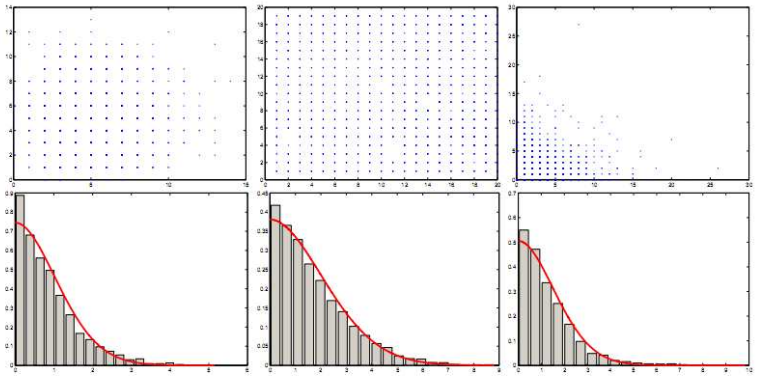
5. Clustering using the Dissimilarity Increments Distribution

In this section we propose a clustering algorithm based on the DID derived above. The DID will be used to determine whether two clusters should be merged into one cluster or not. To obey the underlying hypothesis of the DID model, we will start from a partition of the data produced by a Gaussian mixture decomposition (GMD), which we denote by P^{init}. A major problem in Gaussian mixture decompositions is the inability to detect clusters of arbitrary shapes. We aim to surpass this problem by merging together clusters using the DID model derived above. The decision to merge two clusters will depend on the DID of each separate cluster and the DID of the two clusters combined.

To decide in which order to compare pairs of clusters we sort the pairs using the Mahalanobis distance [1] and test first clusters that are closer to one another. This test is performed for all pairs of clusters, until all the clusters that remain, when tested, are found not to be merged. The overall



(a) $T = 2.4753$ (b) $T = 0.9812$ (c) $T = 3.8872$



(d) $T = 2.9028$ (e) $T = 0.9153$ (f) $T = 2.1267$

Figure 2: Histograms (bar plots) and fitted dissimilarity increments distribution (solid line curves) computed on several datasets of dimension 20. *First row:* (a) scatterplot of the first two dimensions of a Gaussian distribution; (b) scatterplot of the first two dimensions of a Uniform distribution; (c) scatterplot of the first two dimensions of an Exponential distribution. *Second row:* corresponding histograms of dissimilarity increments and fit of the DID. *Third row:* (d) scatterplot of the first two dimensions of a Poisson distribution; (e) scatterplot of the first two dimensions of a Uniform distribution; (f) scatterplot of the first two dimensions of a Geometric distribution. *Fourth row:* corresponding histograms of dissimilarity increments and fit of the DID. Below each pair of figures, we also present the corresponding value of the Cramér-von-Mises test (T).

procedure of this algorithm is summarized in Table 1.

Table 1: Schematic description of the clustering algorithm proposed (GMDID).

Algorithm: GMDID
Input: data with N samples
Output: data partition $P = \{C_1, \ldots, C_K\}$, with $K \leq N$
Initialization: $P^{init} = \{C_1, \ldots, C_N\}$, data partition produced by a GMD
$D_{ij} \leftarrow$ Mahalanobis distance between clusters i and j
for all pairs (i, j) in ascending order of D_{ij} **do**
$p_i \leftarrow$ DID for cluster i, $\quad p_j \leftarrow$ DID for cluster j (eq. 10)
$p_{ij} \leftarrow$ DID for cluster produced merging clusters i and j (eq. 10)
if merge criterion is true **then** merge clusters i and j
end for

5.1. Merge Criterion

We consider two merge criteria: likelihood-ratio test (LRT) and minimum description length (MDL). The first has a parameter consisting of a significance level α; the second one does not have any parameters.

5.1.1. Likelihood-ratio test

We perform a likelihood ratio test (LRT) [24] consisting of the logarithm of the ratio between the joint likelihood of two separate clusters (two DID models) and the likelihood of the merged clusters (single DID model). This LRT is approximated by a chi-square distribution with one degree of free-

dom[1]. Therefore,

$$-2\log\left(\frac{\mathcal{L}(W|\lambda_1,\lambda_2)}{\mathcal{L}(W|\lambda_{12})}\right)\sim\chi^2(1). \tag{14}$$

Two clusters are merged if the p-value from the $\chi^2(1)$ distribution is less than a given significance level α. This significance level is a parameter to be chosen according to some criterion; in this paper we will use the Graph-based Dissimilarity Increments Distribution (G-DID) index, proposed by Fred and Jain [25], to make the choice.

5.1.2. Minimum Description Length

This test consists of computing the minimum description length (MDL) [26] of the separate clusters and the merged clusters. The MDL of a cluster is defined as the sum of the length of the model description, according to the DID hypothesis, and the cost of encoding our estimation of the DIDs. Consider models M_2 and M_1 corresponding to two separate DID models (two separate clusters) and one single DID model (merged clusters), respectively. For $k\in\{1,2\}$, the description length of each model M_k is given by

$$DL^k_{DID}=-\left(\sum_{j=1}^{k}\sum_{i=1}^{|M_k|}\log p_{d_{inc}}(w_i;\lambda_j)\right)+\frac{k}{2}\log|M_k|, \tag{15}$$

where $|M_k|$ is the total number of increments of model k. After this calculation, the MDL test is

$$\text{Choose } M_k : k=\underset{i}{\operatorname{argmin}}\{DL^i_{DID}\}. \tag{16}$$

Two clusters are merged if L^1_{DID} is less than L^2_{DID}.

[1]We have two parameters in the numerator – the expected value of the increments for each of the two clusters separately – and one parameter in the denominator – the expected value of the increments for the two clusters combined.

6. Experimental Results

To obtain the initial data partition, according to the underlying hypothesis of the DID proposed in this paper, one can use any Gaussian mixture decomposition (GMD) algorithm. We used the algorithm proposed by Figueiredo and Jain [10], which is an expectation-maximization (EM) algorithm that finds the number of components using the MDL criterion. With this GMD algorithm we get a partition of the data set with as many clusters as Gaussian components.

6.1. Data

To test the performance of the proposed method, we used 14 datasets: 6 synthetic datasets, 5 real-world datasets from the UCI Machine Learning Repository[2] and 3 datasets containing Usenet articles from different discussion groups, which were obtained from 20-Newsgroups[3]. The synthetic datasets were chosen to take into account a wide variety of situations: well-separated and touching clusters, Gaussian and non-Gaussian clusters, arbitrary shapes and diverse cluster densities. These synthetic datasets are shown in figure 3, and a summary of the real-world datasets is given in table 2. Crabs, House-votes and Wine were normalized to have unit variance.

6.2. Known number of clusters

We start by comparing the performances of GMDID for different initializations and merge criteria. We then choose the best initialization and

[2]http://archive.ics.uci.edu/ml

[3]http://www.ai.mit.edu/people/jrennie/20Newsgroups/

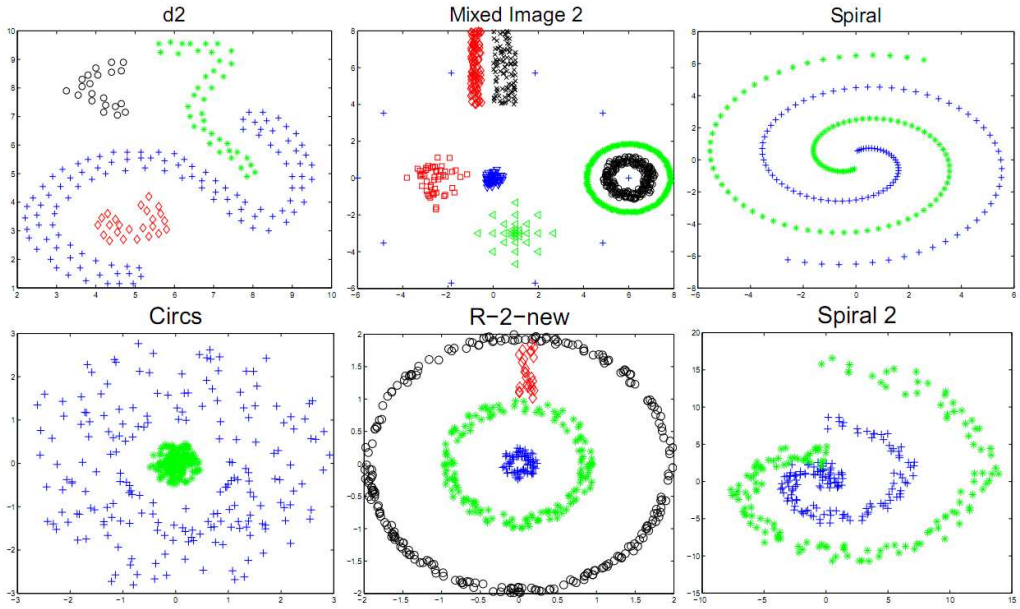


Figure 3: Synthetic datasets used in this paper.

merge criteria, and compare that version of GMDID (which we will denote with GMDID*) with k-means, typical hierarchical methods, and also with the method SLAGLO [18] based on DIs.

One possible initialization for GMD is formed by M random points from the dataset (we denote this as GMD^{rand}). Another interesting possibility is using the final partition given by k-means as part of the initialization of the GMD (denoted by GMD^{init}). We use the k centroids of k-means as initial centroids for the GMD, and add additional randomly chosen centroids, from the dataset, in a total of 50 initial centroids. However, the k-means algorithm has a lot of variability due to random initialization: different initializations yield different final partitions. To initialize k-means itself, we use the Variance Partitioning method proposed by Su and Dy [27], since, with

Table 2: Real-world datasets with the corresponding number of patterns (Ns), number of features (Nf) and number of clusters (Nc).

Data	Ns	Nf	Nc	Data	Ns	Nf	Nc
Breast-cancer	683	9	2	Wine	178	13	3
Crabs	200	5	2	Diff-300	300	10	3
House-votes	232	16	2	Same-300	297	10	3
Iris	150	4	3	Sim-300	291	10	3

it, k-means results were comparable to the best run out of 20 with random initializations. In table 3, we present the results produced by these initializations. Note that although the true number of clusters is known, GMD and GMDID algorithms can use that information only as a lower bound to the number of clusters. Therefore, in Table 3 these algorithms sometimes overestimate the number of clusters, but never underestimate it.

We assess the quality of each resulting partition P using the consistency index (CI) [28], which is the percentage of agreement between P and the ground truth information (also known as accuracy). Tables 3 and 4 summarize the results for the case in which the true number of clusters is known.

From table 3, we see that the proposed method, using the MDL criterion (GMDID_M), is the best, in synthetic datasets, when we use as initial partition the GMD^{rand}. The GMD^{init} as initial partition affects the results given by the proposed method due to the fact that k-means is not well suitable to situations in which the clusters have arbitrary shapes and densities, and therefore the centroids are not correctly initialized. However, in the real-world datasets GMD^{init} has good results in half of the datasets.

If we use the intrinsic criterion of the GMD [10] (which is MDL-based)

Table 3: Consistency index (%) for several variants of GMD [10] and GMDID when the true number of clusters is known. The values in parentheses correspond to the number of clusters found by each algorithm. GMD^{rand} is the Gaussian Mixture Decomposition with random initialization and GMD^{init} uses k-means output as initialization; GMDID is the proposed algorithm ($(\cdot)_M$ means using MDL criterion and $(\cdot)_L$ means using LRT criterion). The best results for each dataset are shown in bold.

	Nc	GMD^{rand}	GMDID_M^{rand}	GMDID_L^{rand}	GMD^{init}	GMDID_M^{init}	GMDID_L^{init}
d2	4	34.00 (13)	**100** (4)	90.50 (5)	34.50 (13)	**100** (4)	90.50 (5)
Mixed Image 2	8	34.10 (38)	**100** (8)	99.19 (8)	41.00 (29)	99.86 (8)	99.05 (8)
Spiral	2	7.00 (45)	**100** (2)	**100** (2)	11.00 (36)	61.50 (5)	58.50 (4)
Circs	2	35.75 (14)	**99.00** (2)	79.00 (3)	49.25 (12)	98.25 (3)	98.25 (2)
R-2-new	4	18.00 (41)	**75.00** (6)	62.60 (8)	21.40 (34)	**75.00** (5)	61.00 (8)
Spiral 2	2	21.33 (26)	71.33 (5)	24.33 (12)	13.67 (25)	**77.00** (5)	23.67 (14)
Breast-cancer	2	45.10 (5)	75.11 (3)	75.11 (3)	68.23 (5)	**79.65** (3)	74.96 (4)
Crabs	2	68.50 (3)	**74.50** (2)	**74.50** (2)	58.50 (2)	58.50 (2)	58.50 (2)
House-votes	2	87.07 (2)	87.07 (2)	87.07 (2)	**89.22** (2)	**89.22** (2)	**89.22** (2)
Iris	3	71.33 (5)	**98.00** (3)	**98.00** (3)	76.00 (5)	96.00 (3)	96.00 (3)
Wine	3	39.89 (1)	39.89 (1)	39.89 (1)	**96.63** (3)	**96.63** (3)	**96.63** (3)
Diff-300	3	**90.67** (3)	**90.67** (3)	**90.67** (3)	73.67 (3)	73.67 (3)	73.67 (3)
Same-300	3	47.47 (2)	47.47 (2)	47.47 (2)	**53.87** (3)	**53.87** (3)	**53.87** (3)
Sim-300	3	37.80 (2)	37.80 (2)	37.80 (2)	**49.14** (3)	**49.14** (3)	**49.14** (3)
Average CI		45.57	78.27	71.87	52.58	**79.16**	73.07

to choose between these initial partitions, only in the Wine, Same-300 and Sim-300 datasets, GMD^{init} is chosen; in all the remaining datasets GMD^{rand} is chosen. If we take a look at the results of GMDID corresponding to these selections, we see that it has the best results, except for the Breast-cancer and House-votes datasets. Thus, MDL could be a good criterion to choose between fully random initialization and some other kind of initialization. In table 4 we denote by GMDID^* the GMDID method using the MDL criterion to merge the components and the initial partition chosen using the intrinsic

MDL criterion of the GMD [10]. This is almost always the best choice.

Table 4: Consistency index (%) for clustering algorithms when the true number of clusters is known. The values in parentheses correspond to the number of clusters found by each algorithm. GMDID* is GMDID_M with initial partition chosen by intrinsic MDL of GMD [10]. Hierarchical clustering algorithms: SLAGLO as proposed in [18], single-link (SL), average-link (AL), complete-link (CL), Ward-link (Ward). The best results for each dataset are shown in bold.

	Nc	k-means	GMDID*	SLAGLO	SL	AL	CL	Ward
d2	4	60.50 (4)	**100** (4)	**100** (4)	**100**	61.50	61.50	51.00
Mixed Image 2	8	41.95 (8)	**100** (8)	34.51 (32)	47.50	51.56	51.01	53.99
Spiral	2	55.00 (2)	**100** (2)	**100** (2)	**100**	52.00	52.00	52.00
Circs	2	72.75 (2)	99.00 (2)	**100** (2)	**100**	61.50	71.00	69.75
R-2-new	4	46.60 (4)	**75.00** (6)	65.00 (2)	58.80	34.00	36.80	43.40
Spiral 2	2	55.67 (2)	71.33 (5)	**77.67** (2)	51.67	53.33	52.33	56.33
Breast-cancer	2	96.05 (2)	75.11 (3)	57.83 (24)	65.15	94.29	85.21	**96.63**
Crabs	2	54.00 (2)	**74.50** (2)	50.00 (1)	50.50	51.50	52.00	55.50
House-votes	2	**89.22** (2)	87.07 (2)	53.45 (5)	53.02	88.36	81.03	85.78
Iris	3	89.33 (3)	**98.00** (3)	66.67 (2)	68.00	90.67	84.00	89.33
Wine	3	**96.63** (3)	**96.63** (3)	56.18 (5)	37.64	38.76	83.71	92.70
Diff-300	3	58.67 (3)	**90.67** (3)	40.33 (3)	35.67	36.67	36.67	66.00
Same-300	3	**54.21** (3)	53.87 (3)	35.35 (2)	35.02	33.67	36.70	38.05
Sim-300	3	42.96 (3)	**49.14** (3)	38.49 (7)	34.36	34.36	34.36	35.05
Average CI		62.25	**83.59**	62.53	59.81	55.87	58.45	63.25

The hierarchical methods and k-means require the user to input the number of clusters. SLAGLO finds the number of clusters automatically. However, it has a cluster isolation parameter[4]. We use the Graph-based Dissimilarity Increments Distribution (G-DID) index [25] to choose the isolation parameter, with values ranging from the mean of the exponential distribu-

[4]Which is a threshold set on the tail of the exponential distribution of the DIs of a cluster (with parameter the inverse of the mean of the increments of that cluster) [18].

tion to 10 times this mean.

In table 4, we see that SLAGLO outperforms the remaining hierarchical clustering algorithms in most synthetic datasets. However, GMDID* yields better results except in the Circs and Spiral 2 datasets. For real-world datasets, the Ward link could be a good choice within the hierarchical clustering algorithms, but if we compare to the partitional algorithms (k-means and GMDID*), we see that the Ward link is the best method only for the Breast-cancer dataset. For the remaining real-world datasets, k-means or GMDID* are a better choice. k-means yields poor results in the synthetic datasets. In some real-world datasets, the clusters are compact and hyperspherical (like Breast-cancer), and k-means works very well. In other datasets, the clusters are Gaussian (as in Iris) and GMDID* works quite well.

6.3. Unknown number of clusters

In this section we assume that no *a priori* information about the number of clusters is available. Therefore, we run GMDID using as initial partition the GMD with fully random initialization. Also, we run k-means initialized with Variance Partitioning [27], but with several values of k. The choice of the best partition was made using G-DID [25] and that partition was used as initial partition for the proposed method; this approach is called KMDID.

Again, we assess the quality of each resulting partition using the consistency index. The results are shown in table 5.

Again, GMDID$_M$ is the best method for the synthetic datasets when the initial partition is given by GMD. KMDID has poor results in synthetic datasets and in most of the real-world datasets.

If we compare the results for GMDID when the true number of clusters

Table 5: Consistency index (%) for partitional clustering algorithms when the true number of clusters is unknown. The values in parentheses correspond to the number of clusters found by each algorithm. KM is k-means run for several values of k and choosen according to the G-DID [25]; GMD is Gaussian Mixture Decomposition [10] with random initialization; KMDID and GMDID are the proposed algorithm ($(\cdot)_M$ means using MDL criterion and $(\cdot)_L$ means using LRT criterion). The best results for each dataset are shown in bold.

	Nc	KM	KMDID$_M$	KMDID$_L$	GMD	GMDID$_M$	GMDID$_L$
d2	4	44.50 (2)	58.00 (1)	44.50 (2)	34.00 (13)	**100** (4)	90.50 (5)
Mixed Image 2	8	46.41 (7)	45.87 (6)	45.87 (6)	34.10 (38)	**100** (8)	99.19 (8)
Spiral	2	55.00 (2)	55.00 (2)	55.00 (2)	7.00 (45)	**100** (2)	**100** (2)
Circs	2	57.25 (11)	**100** (2)	**100** (2)	35.75 (14)	99.00 (2)	79.00 (3)
R-2-new	4	34.40 (12)	**75.20** (4)	**75.20** (4)	18.00 (41)	75.00 (6)	62.60 (8)
Spiral 2	2	55.67 (2)	50.00 (1)	50.00 (1)	21.33 (26)	**71.33** (5)	24.33 (12)
Breast-cancer	2	**96.05** (2)	65.01 (1)	65.01 (1)	45.10 (5)	75.11 (3)	75.11 (3)
Crabs	2	54.00 (2)	54.00 (2)	54.00 (2)	**68.50** (3)	50.00 (1)	50.00 (1)
House-votes	2	75.43 (3)	75.43 (3)	75.43 (3)	**87.07** (2)	**87.07** (2)	**87.07** (2)
Iris	3	89.33 (3)	66.67 (2)	66.67 (2)	**71.33** (5)	66.67 (2)	66.67 (2)
Wine	3	**60.11** (2)	39.89 (1)	**60.11** (2)	39.89 (1)	39.89 (1)	39.89 (1)
Diff-300	3	58.33 (2)	33.33 (1)	33.33 (1)	**90.67** (3)	62.33 (2)	62.33 (2)
Same-300	3	35.35 (2)	35.35 (2)	35.35 (2)	**47.47** (2)	**47.47** (2)	**47.47** (2)
Sim-300	3	**47.42** (9)	37.80 (3)	37.80 (3)	37.80 (2)	37.80 (2)	37.80 (2)
Average CI		57.80	56.54	57.02	45.57	**72.26**	65.85

is available (GMDID* from table 4) and when it is not available (GMDID$_M$ from table 5), we see that the results are similar except for the Crabs, Iris and Diff-300 datasets. This suggests that those datasets may have the same DID for different clusters and, consequently, the method incorrectly merges them. On the other hand, this is a strong indication that GMDID can deal well with situations where the true number of clusters is not known.

7. Discussion

Partitional clustering algorithms, like k-means and GMD, have a major problem in identifying clusters with arbitrary shapes; moreover, k-means also has problems in detecting non-compact clusters. The method proposed here is able to surpass some of these shortcomings. However, it depends on an initial partition, which we have generated by GMD or by k-means.

From our experiments we noticed that the initialization of GMD affects the final results (see Table 3). We initialize the GMD in two ways: (i) randomly choose the initial centroids from the dataset, and (ii) use k-means to initialize the centroids. In some datasets, the results of the proposed method (especially, GMDID$_M$) are independent of the initialization, as in d2 and in R-2-new. But most of the synthetic datasets have better results when GMD is initialized randomly, and most of the real-world datasets have better results when the k-means initialization is given. If no information about the number of clusters is available, we use two different initial partitions. Again, the results are better, in general, if we use GMD instead of KM. However, KMDID$_M$ and KMDID$_L$ present good results in some real-world datasets.

Another problem caused by the random initial partition occurs when the initial number of components is already less than the true number of clusters; for example, in the Wine dataset, the GMD^{rand} only found a single component (see table 3). In these cases, the proposed method is not applicable, since there is nothing to merge. Also, if the proposed method finds less clusters than the true number of clusters (for example, Crabs, Iris and Diff-300 in the last two columns of Table 5), it might mean that the clusters have the same DID, which results in the merging of those clusters.

One way to surpass the difficulties caused by the initialization of the GMD and by the lack of structure in the increments, is to find a good initialization of the GMD and/or to introduce a split strategy. We are still working on these possibilities. GMD^{init} is a reasonable initialization method, but the results show that it still needs to be improved.

8. Conclusions

We have derived a statistical model of dissimilarity increments for high-dimensional data (d-DID) and have particularized the model for $d = 2$ (2-DID). We empirically compared these two distributions with a previous model considered in [18] (exponential distribution) using two statistical distance measures: Cramér-von-Mises criterion and Jensen-Shannon divergence. Empirical evidence showed that d-DID and 2-DID are a better approximation to the empirical distribution than the exponential distribution and that 2-DID is a good approximation to d-DID, while being simpler to compute.

In [19] we proposed the use of this distribution in a novel clustering algorithm: the starting point is a partition given by a Gaussian mixture decomposition and the decision of merging components is based on a likelihood ratio test between the statistical model for the combined components and the statistical model for the separate components. In this paper we propose another merge criterion based on the minimum description length.

Experimental results show that the proposed algorithm depends on the initial partition: if the algorithm used to build the initial partition produces single clusters which in reality should be two separate clusters, our algorithm cannot undo this. However, the proposed clustering algorithm improves the

results and in most of the datasets is better than other clustering algorithms, when the true number of clusters is known. If no information about the true number of clusters is available, the proposed method can detect, in most cases, the true number.

Our results also show that one can choose among multiple initializations easily and reliably using a simple MDL criterion, and that the proposed method, using this choice, yields excellent results on both synthetic and real-world datasets.

9. Acknowledgments

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). This work was partially supported by the Portuguese Foundation for Science and Technology (FCT), scholarship number SFRH/BD/39642/2007, and grant PTDC/EIA-CCO/103230/2008.

Appendix A. Expected value of the expansion factor

In this appendix, we will derive, under some approximations, equation (4). We want to compute

$$\mathbb{E}[A(\Theta)] = \int_{S^{d-1}} \left[\prod_{i=1}^{d-2} p_{\theta_i}(\theta_i) \right] p_{\theta_{d-1}}(\theta_{d-1}) A(\Theta) \, \mathrm{d}_{S^{d-1}} V,$$

where Θ is the vector of angular coordinates of a point in the $(d-1)$-sphere and $A(\Theta)$ is defined in equation (3). Therefore, the volume element in this sphere is $\mathrm{d}_{S^{d-1}} V = \left(\prod_{i=1}^{d-2} \sin^{d-(i+1)} \theta_i \right) \mathrm{d}\theta_1 \ldots \mathrm{d}\theta_{d-1}$. Since we sphered the data, we can assume for simplicity that $\theta_i \sim Unif([0, \pi[)$ for $i = 1, \ldots, d -$

2 and that $\theta_{d-1} \sim Unif([0, 2\pi[)$; then $p_{\theta_i}(\theta_i) = \frac{1}{\pi}$ and $p_{\theta_{d-1}}(\theta_{d-1}) = \frac{1}{2\pi}$. Therefore,

$$\mathbb{E}[A(\Theta)] = \frac{1}{2\pi^{d-1}} \int_0^{2\pi} \int_0^{\pi} \cdots \int_0^{\pi} \left[\Sigma_{11} \cos^2\theta_1 + \sum_{i=2}^{d-1} \Sigma_{i,i} \left(\prod_{k=1}^{i-1} \sin^2\theta_k \right) \cos^2\theta_i + \Sigma_{dd} \left(\prod_{k=1}^{d-1} \sin^2\theta_k \right) \right] \left(\prod_{i=1}^{d-2} \sin^{d-(i+1)}\theta_i \right) \mathrm{d}\theta_1 \, \mathrm{d}\theta_2 \, \ldots \mathrm{d}\theta_{d-2} \, \mathrm{d}\theta_{d-1}$$

We shall decompose the expression within the square brackets into its d additive terms, and solve the multiple integral, separately, for each term. We will use the following results, for $k \in \mathbb{N}$,

$$\int_0^{\pi} \cos^2(x) \sin^k(x) \, \mathrm{d}x = \frac{\sqrt{\pi}\, \Gamma(\frac{1+k}{2})}{2\, \Gamma(2 + \frac{k}{2})} \quad \text{and} \quad \int_0^{\pi} \sin^k(x) \, \mathrm{d}x = \frac{\sqrt{\pi}\, \Gamma(\frac{1+k}{2})}{\Gamma(1 + \frac{k}{2})}, \tag{A.1}$$

and

$$\prod_{k=1}^{M} \frac{\sqrt{\pi}\, \Gamma(\frac{1+k}{2})}{\Gamma(1 + \frac{k}{2})} = \frac{\pi^{M/2}}{\Gamma(1 + \frac{M}{2})} \quad \text{and} \quad \prod_{k=M}^{d} \frac{\sqrt{\pi}\, \Gamma(\frac{1+k}{2})}{\Gamma(1 + \frac{k}{2})} = \frac{\pi^{\frac{d-M+1}{2}}\, \Gamma(\frac{M+1}{2})}{\Gamma(1 + \frac{d}{2})}. \tag{A.2}$$

We start with the first term of the sum (corresponding to Σ_{11}), then we will proceed with Σ_{jj} with $j = 2, \ldots, d-1$, finalizing with the last term of the sum, corresponding to Σ_{dd}.

From (A.1) and (A.2),

$$\int_0^{2\pi} \mathrm{d}\theta_{d-1} \int_0^{\pi} \sin\theta_{d-2} \mathrm{d}\theta_{d-2} \cdots \int_0^{\pi} \sin^{d-j-1}\theta_j \mathrm{d}\theta_j \cdots \int_0^{\pi} \Sigma_{11} \cos^2\theta_1 \sin^{d-2}\theta_1 \mathrm{d}\theta_1$$
$$= \Sigma_{11}\, 2\pi \left(\prod_{k=1}^{d-3} \frac{\sqrt{\pi}\, \Gamma(\frac{1+k}{2})}{\Gamma(1 + \frac{k}{2})} \right) \frac{\sqrt{\pi}\, \Gamma(\frac{d-1}{2})}{2\, \Gamma(2 + \frac{d-2}{2})} = \Sigma_{11}\, \frac{\pi^{d/2}}{\Gamma(1 + \frac{d}{2})};$$

$$\int_0^{2\pi} \mathrm{d}\theta_{d-1} \int_0^{\pi} \sin\theta_{d-2} \mathrm{d}\theta_{d-2} \cdots \int_0^{\pi} \sin^{d-j-2}\theta_{j+1} \mathrm{d}\theta_{j+1}$$
$$\int_0^{\pi} \Sigma_{jj} \cos^2\theta_j \sin^{d-j-1}\theta_j \mathrm{d}\theta_j \int_0^{\pi} \sin^{d-j+2}\theta_{j-1} \mathrm{d}\theta_{j-1} \cdots \int_0^{\pi} \sin^{d}\theta_1 \mathrm{d}\theta_1$$
$$= \Sigma_{jj}\, 2\pi \left(\prod_{k=1}^{d-j-2} \frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{\Gamma(1+\frac{k}{2})} \right) \frac{\sqrt{\pi}\,\Gamma(\frac{d-2}{2})}{2\,\Gamma(2+\frac{d-3}{2})} \left(\prod_{k=d-j+2}^{d} \frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{\Gamma(1+\frac{k}{2})} \right)$$
$$= \Sigma_{jj}\, \frac{\pi^{d/2}}{\Gamma(1+\frac{d}{2})},$$

with $j = 2, \ldots, d-1$.

$$\Sigma_{dd} \int_0^{2\pi} \sin^2\theta_{d-1} \mathrm{d}\theta_{d-1} \int_0^{\pi} \sin^3\theta_{d-2} \mathrm{d}\theta_{d-2} \cdots \int_0^{\pi} \sin^{d-j+1}\theta_j \mathrm{d}\theta_j \cdots \int_0^{\pi} \sin^{d}\theta_1 \mathrm{d}\theta_1$$
$$= \Sigma_{dd}\, \pi \left(\prod_{k=3}^{d} \frac{\sqrt{\pi}\,\Gamma(\frac{1+k}{2})}{\Gamma(1+\frac{k}{2})} \right) = \Sigma_{dd}\, \frac{\pi^{d/2}}{\Gamma(1+\frac{d}{2})}$$

If we add all the previous results, we get

$$\mathbb{E}[A(\Theta)] = \frac{1}{2\pi^{d-1}}\, \frac{\pi^{d/2}}{\Gamma(1+\frac{d}{2})}\, \mathrm{tr}(\Sigma) = \frac{\pi^{-d/2+1}}{2\Gamma(1+\frac{d}{2})}\, \mathrm{tr}(\Sigma)$$

Appendix B. Probability Density Function of DIs

Define $D_1 = d(\mathbf{x}, \mathbf{y})$ and $D_2 = d(\mathbf{y}, \mathbf{z})$, which follow the distribution in equation (6). The PDF of $W = D_1 - D_2$ is given by the convolution

$$p_{_W}(w) = \int_{-\infty}^{\infty} (2G_d(\eta))^2 \left(t(t+w)\right)^{d-1} \exp\left(-C_d(\eta)\left[t^2 + (t+w)^2\right]\right)$$
$$\mathbf{1}_{\{t\geq 0\}} \mathbf{1}_{\{t+w\geq 0\}}\, \mathrm{d}t. \tag{B.1}$$

with $\eta \equiv \mathrm{tr}(\Sigma)$, $G_d(\eta) \equiv d^{d/2}\Gamma(d/2)^{d/2-1} 2^{-d} \eta^{-d/2} \pi^{d/2(d/2-1)}$ and $C_d(\eta) \equiv d\Gamma(d/2)(4\eta)^{-1}\pi^{d/2-1}$. We need to consider two cases: $w \geq 0$ and $w < 0$.

Case 1 ($w \geq 0$)**:** In this case the integral we need to solve is

$$p_{_W}(w) = \int_0^\infty (2G_d(\eta))^2 \left(t(t+w)\right)^{d-1} \exp\left(-C_d(\eta)\left[t^2 + (t+w)^2\right]\right)\, \mathrm{d}t. \quad \text{(B.2)}$$

If we use the binomial formula to decompose $[t(t+w)]^{d-1}$ and replace in equation (B.2), we get

$$\begin{aligned} p_{_W}(w) &= 4G_d(\eta)^2 \int_0^\infty \sum_{k=0}^{d-1} \binom{d-1}{k} t^{2d-2-k} w^k \exp\left(-C_d(\eta)\left[t^2 + (t+w)^2\right]\right) \mathrm{d}t \\ &= (2G_d(\eta))^2 \left[\sum_{k=0}^{d-1} \binom{d-1}{k} w^k \int_0^\infty t^{2d-2-k} \exp\left(-C_d(\eta) w^2/2\right) \right. \\ &\qquad \left. \exp\left(-C_d(\eta)\left(\sqrt{2}t + w/\sqrt{2}\right)^2\right)\, \mathrm{d}t\right]. \end{aligned}$$

Define $u \equiv \sqrt{2}t + w/\sqrt{2}$. With the new variable, the integration interval is $\left[w/\sqrt{2}, +\infty\right[$, and $\frac{dt}{du} = 1/\sqrt{2}$. Now, the integral can be written as

$$\begin{aligned} p_{_W}(w) = (2G_d(\eta))^2 \left[\sum_{k=0}^{d-1} \binom{d-1}{k} \frac{w^k}{\sqrt{2}} \exp\left(-C_d(\eta) w^2/2\right) \right. \\ \left. \int_{w/\sqrt{2}}^\infty \left(\frac{u}{\sqrt{2}} - \frac{w}{2}\right)^{2d-2-k} \exp\left(-C_d(\eta) u^2\right)\, \mathrm{d}u\right]. \end{aligned}$$

Using, again, the binomial formula to decompose $\left(u/\sqrt{2} - w/2\right)^{2d-2-k}$ and substituting in the previous integral, we get

$$\begin{aligned} p_{_W}(w) = (2G_d(\eta))^2 \left[\sum_{k=0}^{d-1} \binom{d-1}{k} \frac{w^k}{\sqrt{2}} \exp\left(-C_d(\eta) w^2/2\right) \left[\sum_{i=0}^{2d-2-k} (-1)^i w^i \right.\right. \\ \left.\left. \binom{2d-2-k}{i} 2^{-d+k/2-i/2+1} \int_{w/\sqrt{2}}^\infty u^{2d-2-k-i} \exp\left(-C_d(\eta) u^2\right)\, \mathrm{d}u\right]\right]. \end{aligned}$$

If we perform a new change of variables, $x = u^2$, the integration interval with the new variable is $[w^2/2, \infty)$ and $\frac{du}{dx} = x^{-1/2}/2$. The integral can be

written as

$$p_{_W}(w) = (2G_d(\eta))^2 \left[\sum_{k=0}^{d-1} \binom{d-1}{k} \frac{w^k}{\sqrt{2}} \exp\left(-C_d(\eta)w^2/2\right) \left[\sum_{i=0}^{2d-2-k} (-1)^i w^i \binom{2d-2-k}{i} 2^{-d+k/2-i/2} \int_{w^2/2}^{\infty} x^{d-3/2-k/2-i/2} \exp\left(-C_d(\eta)x\right) \mathrm{d}x\right]\right].$$

Furthermore, the upper incomplete gamma function is $\Gamma(a,x) = \int_x^\infty t^{a-1}e^{-t}\mathrm{d}t$ and it is easy to prove that $\Gamma(a,bx)/b^a = \int_x^\infty t^{a-1}e^{-bt}\mathrm{d}t$. Thus,

$$p_{_W}(w) = (2G_d(\eta))^2 \left[\sum_{k=0}^{d-1} \binom{d-1}{k} \frac{w^k}{\sqrt{2}} \exp\left(-C_d(\eta)w^2/2\right) \left[\sum_{i=0}^{2d-2-k} (-1)^i w^i \binom{2d-2-k}{i} 2^{-d+k/2-i/2} \frac{\Gamma\left(d-1/2-k/2-i/2, C_d(\eta)w^2/2\right)}{C_d(\eta)^{d-1/2-k/2-i/2}}\right]\right].$$

Case 2 ($w < 0$): In this case, equation (B.1) is given by

$$p_{_W}(w) = \int_{-w}^{\infty} (2G_d(\eta))^2 \left(t(t+w)\right)^{d-1} \exp\left(-C_d(\eta)\left[t^2 + (t+w)^2\right]\right) \mathrm{d}t.$$

To solve the previous integral we use ideas analogous to the case $w \geq 0$. We get, for $w < 0$,

$$p_{_W}(w) = (2G_d(\eta))^2 \left[\sum_{k=0}^{d-1} \binom{d-1}{k} \frac{(-w)^k}{\sqrt{2}} \exp\left(-C_d(\eta)w^2/2\right) \left[\sum_{i=0}^{2d-2-k} (-1)^i (-w)^i \binom{2d-2-k}{i} 2^{-d+k/2-i/2} \frac{\Gamma\left(d-1/2-k/2-i/2, C_d(\eta)w^2/2\right)}{C_d(\eta)^{d-1/2-k/2-i/2}}\right]\right].$$

Both $W = w$ and $W = -w$ yield the same value for $|W|$, and therefore the PDF for $|W|$ obeys $p_{_{|W|}}(w) = p_{_W}(w) + p_{_W}(-w) = 2p_{_W}(w)$. This yields the PDF of the DIs is given by equation (7), after some simplifications.

Appendix C. Expected Value of the DIs

In this appendix we will derive the expected value of the DIs and obtain expression (8). The expected value is given by

$$\mathbb{E}[w] = \frac{G_d(\eta)^{1/d}}{2^{d-5/2}\Gamma(d/2)^{2-1/d}} \left[\sum_{k=0}^{d-1} \sum_{i=0}^{2d-2-k} (-1)^i \binom{d-1}{k} \binom{2d-2-k}{i} 2^{k/2-i/2} \right.$$
$$\left. C_d(\eta)^{k/2+i/2} \int_0^\infty w^{k+i+1} \exp\left(-\frac{C_d(\eta)}{2}\, w^2\right) \Gamma\left(\frac{2d-1-k-i}{2}, \frac{C_d(\eta)}{2}\, w^2\right) \right] \mathrm{d}w.$$

We will use the following result [21]

$$\int_0^\infty x^{a-1} e^{-sx}\, \Gamma(b,x)\, \mathrm{d}x = \frac{\Gamma(a+b)}{a(1+s)^{a+b}}\, F\left(1, a+b; 1+a; \frac{s}{1+s}\right),$$
$$\text{with } Re(s) > -1,\ Re(a+b) > 0,\ Re(a) > 0 \qquad \text{(C.1)}$$

where $F(a,b;c;z)$ is the hypergeometric function.

Making the change of variables $x = \frac{C_d(\eta)}{2} w^2$, i.e., $w = \sqrt{2} C_d(\eta)^{-1/2} x^{1/2}$, the new integration interval is $[0,\infty)$ and $\frac{dw}{dx} = (2C_d(\eta))^{-1/2} x^{-1/2}$. The integral can be written as

$$\int_0^\infty 2^{k/2+i/2} C_d(\eta)^{-k/2-i/2-1}\, x^{k/2+i/2}\, e^{-x}\, \Gamma\left(\frac{2d-k-i-1}{2}, x\right)\, \mathrm{d}x.$$

Since $0 \le k \le d-1$, $0 \le i \le 2d-k-2$ and $d \ge 2$, we have $s = 1 > -1$, $a = k/2 + i/2 + 1 > 0$ and $a + b = d + 1/2 > 0$, which means we are in the conditions of equation (C.1). So the integral is given by

$$\frac{2^{k/2+i/2}}{C_d(\eta)^{k/2+i/2+1}} \frac{\Gamma(d+1/2)}{(k/2+i/2+1)2^{d+1/2}} F\left(1, d+1/2; k/2+i/2+2; \frac{1}{2}\right).$$

We use Euler's transformation formula [21], $F(a,b;c;z) = (1-z)^{c-a-b}\, F(c-a, c-b; c; z)$. The previous equation becomes (after some simplifications)

$$\frac{1}{2C_d(\eta)^{k/2+i/2+1}} \frac{\Gamma(d+1/2)}{k/2+i/2+1} F\left(\frac{k+i}{2}+1, \frac{k+i+3}{2}-d; \frac{k+i}{2}+2; \frac{1}{2}\right).$$

Also, the incomplete beta function is related to the hypergeometric function by $B_z(p,q) = \frac{z^p}{p}\ F(p, 1-q; p+1; z)$, for $p > 0$, $q > 0$ and $0 \leq z \leq 1$ [21]. So, we can write

$$\frac{2^{k/2+i/2}}{C_d(\eta)^{k/2+i/2+1}}\Gamma(d+1/2)B_{\frac{1}{2}}\left(\frac{k+i}{2}+1, d-\frac{k+i+1}{2}\right).$$

Therefore,

$$\begin{aligned}\mathbb{E}[w] &= \frac{G_d(\eta)^2}{2^{d-5/2}C_d(\eta)^{d+1/2}}\left[\sum_{k=0}^{d-1}\sum_{i=0}^{2d-2-k}(-1)^i\binom{d-1}{k}\binom{2d-2-k}{i}2^k\right. \\ &\quad \left.\Gamma(d+1/2)B_{\frac{1}{2}}\left(\frac{k+i}{2}+1, d-\frac{k+i+1}{2}\right)\right] \\ &= \eta^{1/2}Q_d^{-1}\left[\sum_{k=0}^{d-1}\sum_{i=0}^{2d-2-k}(-1)^i\binom{d-1}{k}\binom{2d-2-k}{i}2^k\right. \\ &\quad \left.B_{\frac{1}{2}}\left(\frac{k+i}{2}+1, d-\frac{k+i+1}{2}\right)\right],\end{aligned}$$

with $Q_d \equiv 2^{d-7/2}d^{1/2}\pi^{d/4-1/2}\Gamma(d/2)^{5/2}\Gamma(d+1/2)^{-1}$.

References

[1] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Elsevier Academic Press, 2 edition, 2003.

[2] S. Draghici, Data Analysis Tools for DNA Microarrays, Chapman & Hall, 2 edition, 2003.

[3] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2nd edition, 2009.

[4] A. K. Jain, R. C. Dubes, Algorithms for Clustering Data, Prentice Hall, 1988.

[5] R. Xu, D. W. II, Survey of clustering algorithms, IEEE Trans. on Neural Networks 16 (2005) 645–678.

[6] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM Computing Surveys 31 (1999) 264–323.

[7] H. Tenmoto, M. Kudo, M. Shimbo, Mdl-based selection of the number of components in mixture models for pattern recognition, in: Joint IAPR Int. Workshops on Structural and Syntactic Pattern recognition and Statistical Techniques in Pattern Recognition (SSPR/SPR 1998).

[8] G. H. Ball, D. J. Hall, ISODATA, A Novel Method of Data Analysis and Pattern Classification, Technical Report, Stanford Research Institute, 1965.

[9] B. Mirkin, Concept learning and feature selection based on square-error clustering, Machine Learning 35 (1999) 25–39.

[10] M. A. T. Figueiredo, A. K. Jain, Unsupervised learning of finite mixture models, IEEE Trans. on Pattern Analysis and Machine Intelligence 24 (2002) 381–396.

[11] A. P. Benavent, F. E. Ruiz, J. M. S. Martínez, Ebem: An entropy-based em algorithm for gaussian mixture models, in: Proc. of the 18th Int. Conf. on Pattern Recognition (ICPR 2006), pp. 451–455.

[12] N. Ueda, R. Nakano, Z. Ghahramani, G. E. Hinton, Smem algorithm for mixture models, Neural Computation 12 (2000) 2109–2128.

[13] S. Guha, R. Rastogi, K. Shim, Cure: An efficient clustering algorithm for large datasets, in: Proc. of the ACM SIGMOD Int. Conf. of Management of Data (SIGMOD 1998), pp. 73–84.

[14] S. Guha, R. Rastogi, K. Shim, Rock: A robust clustering algorithm for categorical attributes, in: Proc. of the 15th Int. Conf. on Data Engineering (ICDE 2000), pp. 512–521.

[15] G. Karypis, E.-H. Han, V. Kumar, Chameleon: Hierarchical clustering using dynamic modeling, Computer 32 (1999) 68–75.

[16] T. Zhang, R. Ramakrishnan, M. Livny, Birch: An efficient data clustering method for very large databases, in: Proc. of the ACM SIGMOD Int. Conf. on Management of Data (SIGMOD 1996), pp. 103–114.

[17] R. Sharan, R. Shamir, Click: A clustering algorithm with applications to gene expression analysis, in: Proc. of the 8th Int. Conf. on Intelligent Systems for Molecular Biology (ISMB 2000), pp. 307–316.

[18] A. Fred, J. Leitão, A new cluster isolation criterion based on dissimilarity increments, IEEE Trans. on Pattern Analysis and Machine Intelligence 25 (2003) 944–958.

[19] H. Aidos, A. Fred, On the distribution of dissimilarity increments, in: Proc. of the 5th Iberian Conf. on Pattern Recognition and Image Analysis (IbPRIA 2011), pp. 192–199.

[20] N. L. Johnson, S. Kotz, N. Balakrishnan, Continuous Univariate Distributions, volume 1, John Wiley & Sons Ltd., 2 edition, 1994.

[21] F. W. J. Olver, D. W. Lozier, R. F. Boisvert, C. W. Clark (Eds.), NIST Handbook of Mathematical Functions, Cambridge University Press, 2010.

[22] T. W. Anderson, On the distribution of the two-sample cramér-von-mises criterion, Annals of Mathematical Statistics 33 (1962) 1148–1159.

[23] J. Lin, Divergence measures based on the shannon entropy, IEEE Trans. on Information Theory 37 (1991) 145–151.

[24] E. L. Lehmann, J. P. Romano, Testing Statistical Hypotheses, Springer, 3 edition, 2005.

[25] A. Fred, A. Jain, Cluster validation using a probabilistic attributed graph, in: Proc. of the 19th Int. Conf. on Pattern Recognition (ICPR 2008), pp. 1–4.

[26] D. J. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 5 edition, 2006.

[27] T. Su, J. G. Dy, In search of deterministic methods for initializing k-means and gaussian mixture clustering, Intelligent Data Analysis 11 (2007) 319–338.

[28] A. L. N. Fred, Finding consistent clusters in data partitions, in: Proc. of the 3rd Int. Workshop on Multiple Classifier Systems (MCS 2001), pp. 309–318.

[16] - (SIMBAD Technical Report n. 2011_09)

Aidos, H., Fred, A.L.N.: Hierarchical clustering with high order dissimilarities. In Perner, P., ed.: Machine Learning and Data Mining in Pattern Recognition. Volume 6871 of Lecture Notes in Computer Science. Springer (2011) 280–293 International Conference on Machine Learning and Data Mining - MLDM 2011, New York, NY, USA.

Hierarchical Clustering with High Order Dissimilarities

Helena Aidos and Ana Fred

Instituto de Telecomunicações, Instituto Superior Técnico, Lisbon, Portugal
{haidos, afred}@lx.it.pt

Abstract. This paper proposes a novel hierarchical clustering algorithm based on high order dissimilarities. These *dissimilarity increments* are measures computed over triplets of nearest neighbor points. Recently, the distribution of these dissimilarity increments was derived analytically. We propose to incorporate this distribution in a hierarchical clustering algorithm to decide whether two clusters should be merged or not. The proposed algorithm is parameter-free and can identify classes as the union of clusters following the dissimilarity increments distribution. Experimental results show that the proposed algorithm has excellent performance over well separated clusters, also providing a good hierarchical structure insight into touching clusters.

Keywords: dissimilarity increments, minimum description length, hierarchical clustering, single-link

1 Introduction

Clustering is used in various application areas, such as exploratory data analysis and data mining [7]. It is also known as unsupervised classification of patterns into groups (clusters). The aim is to find a data partition such that intra-cluster similarities are higher than inter-cluster similarities. Clustering can be performed through partitional or hierarchical approaches, and can use many different ways to measure the (dis)similarity of patterns [7, 9].

In the first of these two approaches, partitional methods, one assigns each data pattern to exactly one cluster; the number of clusters, K, is typically small and set beforehand as a design parameter. Otherwise, the choice of K may itself be addressed automatically, as a model selection problem. The most widespread partitional algorithm is also the most simple: K-means, using a centroid as cluster representative, minimizes a mean-square error criterion based on the Euclidean distance as measure of pairwise dissimilarity [9]. Also, algorithms which estimate probability density functions from the data, such as Gaussian mixture decomposition algorithms [2, 10], can also be used as clustering techniques.

The other class of clustering approaches, hierarchical methods, yields a set of nested partitions which is graphically shown as a dendrogram [6]. A data partition can be obtained by cutting the dendrogram at an appropriate level. A further subdivision of hierarchical methods is agglomerative and divisive algorithms [9]. Agglomerative methods start with a very high number of clusters (often equal to the number of data points, such that each point is one cluster), and produce a sequence of partitions with decreasing

number of clusters. Two widely used agglomerative methods are the single-link and the complete-link algorithms [7]. Divisive approaches are the opposite: they start with a very low number of clusters (often just one containing all the data points), and produce a sequence of partitions with increasing number of clusters. Divisive methods are less common than agglomerative ones.

Usually, clustering algorithms use a (dis)similarity measure between pairs of patterns, which is difficult to choose since one has no prior knowledge about cluster shapes in the data. The most typical dissimilarity measure is simply the Euclidean distance between two points, but many other measures can be used [9]. Fred and Leitão proposed recently [4] a new high order dissimilarity measure called *dissimilarity increments*, which is computed over triplets of nearest neighbor patterns. The fact that this measure uses three data points at a time, instead of two, gives more information about patterns belonging to the same cluster, since dissimilarity increments change smoothly if the patterns are in the same cluster and high values of increments correspond to patterns lying in different clusters [4]. In that paper, Fred and Leitão also proposed a hierarchical clustering algorithm based on dissimilarity increments.

Recently, the probability distribution for dissimilarity increments of a Gaussian cluster was derived analytically under mild approximations [1]. That distribution was used to create a partitional clustering algorithm, which used a Gaussian mixture decomposition as initialization. Using this initialization presents some problems, such as the tendency to capture more than one cluster into one Gaussian component if they are located near each other, even if they are separable.

In this paper, we propose a novel agglomerative hierarchical clustering algorithm to surpass the limitation mentioned above. The decision of which clusters to join is based on the dissimilarity increments distribution (DID) mentioned in the previous paragraph. This algorithm has the advantage of having no parameters. Furthermore, we define a class as the union of several clusters, where each cluster follows the DID. This definition of class (as opposed to defining each cluster as one class) allows the algorithm proposed in this paper to identify clusters belonging to the same class. We apply this algorithm to 8 synthetic data sets and 4 real data sets and compare it to other algorithms based on dissimilarity increments, as well as to some well-known clustering algorithms.

This paper is structured as follows: Section 2 defines dissimillarity increments and explains briefly the derivation of the dissimilarity increments distribution (DID). In Section 3, we propose the incorporation of the DID in a hierarchical approach and explain the differences between this new approach and the typical hierarchical algorithms like single-link. We present, in Section 4, the results of the proposed algorithm for 8 synthetic data sets with different characteristics and 4 real data sets from the UCI Machine Learning Repository. These results are compared with other two algorithms based on dissimilarity increments [1, 4] and with two more convencional hierarchical clustering algorithms: single-link and average-link. Conclusions are drawn in Section 5.

2 Dissimilarity Increments

Consider a set of patterns X, and let x_i represent an element in this set. The inter-pattern relationships can be measured by some dissimilarity measure, $d(\cdot,\cdot)$, which can be, for instance, the Euclidean distance if the patterns are represented in a feature space.

Let $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$ be the triplet of nearest neighbors belonging to the set X, where $\mathbf{x}_j$ is the nearest neighbor of $\mathbf{x}_i$ and $\mathbf{x}_k$ is the nearest neighbor of $\mathbf{x}_j$ different from $\mathbf{x}_i$. The *dissimilarity increment* [4] between the neighboring patterns is defined as

$$d_{inc}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) = |d(\mathbf{x}_i, \mathbf{x}_j) - d(\mathbf{x}_j, \mathbf{x}_k)| \,. \tag{1}$$

The use of this measure gives different information than the use of pairwise distances. Since a cluster is a set of patterns sharing some characteristics, the dissimilarity increments between neighboring patterns should not occur with abrupt changes and the dissimilarity increments between well separated clusters will have higher values.

A related concept which we will use is the *gap* between two clusters. The gap of cluster C_i with respect to cluster C_j is defined exactly by the same equation above, but with a particular triplet of points: x_i and x_j are the closest pair of points such that $x_i \in C_i$ and $x_j \in C_j$, and x_k is the nearest neighbor of x_i within C_i.

2.1 Dissimilarity Increments Distribution

Recently, Aidos and Fred [1] have derived the dissimilarity increments distribution (DID) under the hypothesis of Gaussian distribution for each cluster. We now briefly explain how to proceed to obtain the DID and show empirical evidence that this DID can be applied to higher-dimensional data sets with various cluster densities.

Assume that $X \in \mathbb{R}^2$, and that elements of X are independent and identically distributed, drawn from a normal distribution, $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ. In this paper, the Euclidean distance is used as the dissimilarity measure, and with no loss of generality, we can assume that $\boldsymbol{\mu}$ is zero and Σ is a diagonal matrix (this corresponds to a rotation and a translation of the data, which does not affect the Euclidean distances).

The square Euclidean distance in a sphered gaussian space is given by

$$(D^*)^2 = \sum_{i=1}^{2} \frac{(x_i - y_i)^2}{2\Sigma_{ii}} \tag{2}$$

and follows a chi-square distribution with 2 degrees of freedom, which is equivalent to an exponential distribution with parameter 1/2 [8]. To define the square Euclidean distance in the original space we need to do a transformation defined as (see [1] for details)

$$D^2 = \mathrm{tr}(\Sigma)(D^*)^2, \tag{3}$$

and the probability density function of $D = d(\mathbf{x}, \mathbf{y})$ is

$$p_{_D}(z) = \frac{z}{\mathrm{tr}(\Sigma)} \, \exp\left(-\frac{z^2}{2\,\mathrm{tr}(\Sigma)}\right), \; z \in [0, \infty). \tag{4}$$

The probability density function for the dissimilarity increments is obtained as a convolution between $d(\mathbf{x}, \mathbf{y})$ and $d(\mathbf{y}, \mathbf{z})$, and is given by

$$p_{d_{inc}}(w; \Sigma) = \frac{w}{2\,\mathrm{tr}(\Sigma)}\ \exp\left(-\frac{w^2}{2\,\mathrm{tr}(\Sigma)}\right) + \frac{\sqrt{\pi}}{4\left(\mathrm{tr}(\Sigma)\right)^{3/2}}\ \left(2\,\mathrm{tr}(\Sigma) - w^2\right) \times \\ \times \exp\left(-\frac{w^2}{4\,\mathrm{tr}(\Sigma)}\right) \mathrm{erfc}\left(\frac{w}{2\sqrt{\mathrm{tr}(\Sigma)}}\right), \tag{5}$$

where $\mathrm{erfc}(\cdot)$ is the complementary error function.

The DID in equation (5) requires explicit knowledge of the covariance matrix, Σ, in the original gaussian space. Thus, Aidos and Fred [1] proposed that the distribution is rewritten as a function of the mean value of the dissimilarity increments, $\lambda = \mathbb{E}[w]$, which is given by $\lambda = \frac{\sqrt{2\pi}}{2}\left(2-\sqrt{2}\right)^2 \mathrm{tr}(\Sigma)^{1/2}$. Therefore, we obtain an approximation for the dissimilarity increments distribution of a cluster that only depends of the mean of all the increments in that cluster:

$$p_{d_{inc}}(w; \lambda) = \frac{\pi\left(2-\sqrt{2}\right)^2}{4\lambda^2}\ w\ \exp\left(-\frac{\pi\left(2-\sqrt{2}\right)^2}{4\lambda^2}\ w^2\right) + \frac{\pi^2\left(2-\sqrt{2}\right)^3}{8\sqrt{2}\lambda^3}\ \times \\ \times \left(\frac{4\lambda^2}{\pi\left(2-\sqrt{2}\right)^2} - w^2\right) \exp\left(-\frac{\pi\left(2-\sqrt{2}\right)^2}{8\lambda^2}\ w^2\right) \mathrm{erfc}\left(\frac{\sqrt{\pi}\left(2-\sqrt{2}\right)}{2\sqrt{2}\lambda}\ w\right). \tag{6}$$

Although the underlying hypothesis of this distribuition is that the data comes from a cluster with a gaussian distribution, the distribution in equation (6) only depends of the mean of all the increments in that cluster. This means that we no longer need to have a gaussian cluster to use this distribution. In figure 1 we show this fact by presenting several data sets generated from several continuous and discrete distributions. For all the data sets we generate 2000 points in 50 dimensions. In all the distributions presented we see that the DID fits the histogram well.

3 Hierarchical Clustering Algorithm

3.1 Algorithm

The DID proposed in section 2.1 was used in [1] to create a partitional clustering algortihm, where the starting point was a gaussian mixture proposed by Figueiredo and Jain [2]. However, the gaussian mixture decomposition has some difficulties in identifying arbitrarily shaped clusters. Worse still is the fact, if this initial step is done wrongly, a gaussian component can overlap two or more classes in a way it is impossible for that algorithm to separate the classes correctly [1].

We propose an agglomerative hierarchical strategy to overcome the difficulties caused by the gaussian mixture decomposition. Recall that agglomerative methods start with a very high number of clusters and progressively join pairs of clusters. In our case, we

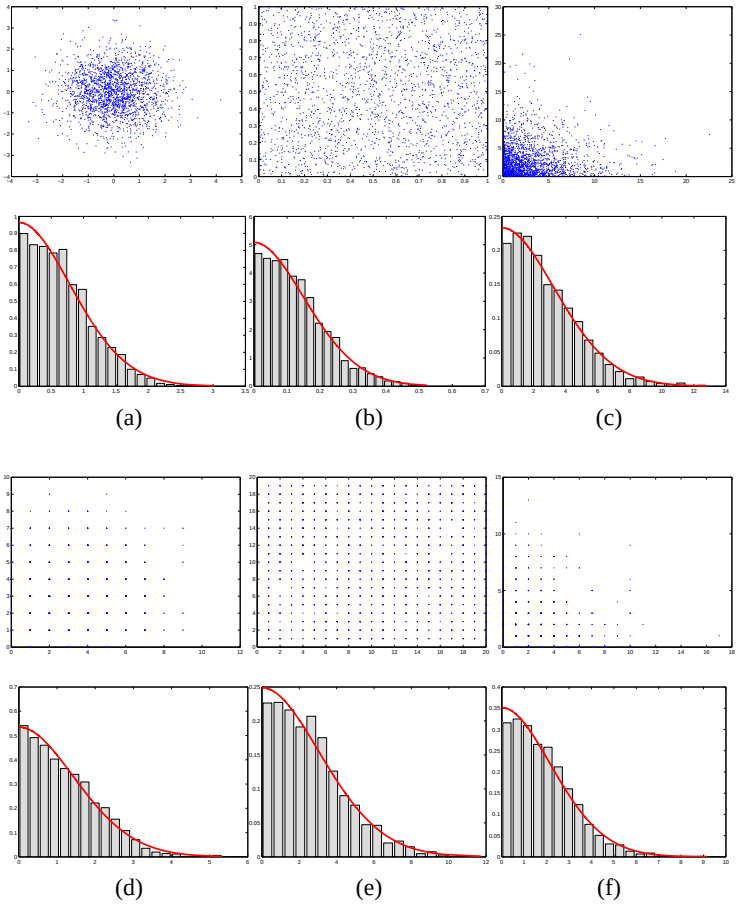


Fig. 1: Histograms (bar plots) and fitted dissimilarity increments distribution (solid line curves) computed over neighboring patterns. *First row:* (a) scatterplot of the first two dimensions of a Gaussian distribution; (b) scatterplot of the first two dimensions of an Uniform distribution; (c) scatterplot of the first two dimensions of an Exponential distribution. *Second row:* corresponding histograms of dissimilarity increments and fit of the DID. *Third row:* (d) scatterplot of the first two dimensions of a Poisson distribution; (e) scatterplot of the first two dimensions of an Uniform distribution; (f) scatterplot of the first two dimensions of a Geometric distribution. *Fourth row:* corresponding histograms of dissimilarity increments and fit of the DID.

will start with each point as a separate cluster (just like the single-link algorithm), and will decide whether to merge two clusters or not through one of the following four tests.

We automatically merge clusters if both of them have less than 6 points. We use a minimum of 6 points because we believe that it is the minimum number of points that allows us to compute a rough estimate of the DID. Therefore, when comparing clusters which both have less than 6 points, our algorithm behaves exactly like single-link.

If one of the clusters (C_i) has less than 6 points and one (C_j) has 6 or more points, we check whether the increments of C_i fall in the tail of the DID of C_j. More precisely, if the mean of the increments of C_i is smaller than 7 times the mean of the increments of C_j, we merge the two clusters. If it is larger, we keep them separate.

If both clusters have 6 or more points, we use a test similar to the algorithm proposed in [4]. If the gap of cluster C_i with respect to cluster C_j belongs to the tail of the DID of cluster C_i, we "freeze" C_i, which means that it is no longer tested with any other cluster. We do the same test for C_j relative to C_i. If none of the two clusters is "frozen", we use a minimum description length (MDL) criterion to decide whether they should be merged or not, described in the next section 3.2.

We continue this procedure until all the pairs of clusters have been tested. The computational complexity is $O(N^2)$, where N is the number of data points. An outline of all the procedure is in Table 1.

Table 1: Schematic description of the clustering algorithm proposed (SL-DID).

Algorithm: SL-DID
Input: data with N samples
Output: data partition
Initialization: Each pattern is a cluster
repeat
Choose the most similar pair of clusters (C_i, C_j) not yet tested
if # $C_i < 6$ and # $C_j < 6$
merge clusters C_i, C_j into a new cluster
if # $C_i \geq 6$ and # $C_j < 6$
if $inc(C_j)$ are not in the tail of the pdf of $inc(C_i)$
merge clusters C_i, C_j into a new cluster
else do not merge C_i, C_j
if # $C_i \geq 6$ and # $C_j \geq 6$
compute $gap(C_i)$ and $gap(C_j)$, and $DL(C_i)$, $DL(C_j)$ and $DL(C_i \cup C_j)$
if $gap(C_i)$ is in the tail of the pdf of $inc(C_i)$
freeze cluster C_i
elseif $gap(C_j)$ is in the tail of the pdf of $inc(C_j)$
freeze cluster C_j
elseif $DL(C_i \cup C_j) <= DL(C_i) + DL(C_j)$
merge clusters C_i, C_j into a new cluster
else do not merge C_i, C_j
until all pairs of clusters should not be merged

3.2 Minimum Description Length criterion

We now present the MDL criterion for our case. We consider a prior $p(\lambda) = 1/\lambda$ (a prior that favors small values of λ, which is appropriate since increments within a cluster are expected to be small) and the likelihood function $p_{d_{inc}}(W|\lambda)$ defined in equation (6), with W the set of increments computed inside a cluster. The description length for a cluster C_i can be shown to be, under some approximations [5]

$$DL(C_i) = \frac{1}{2}(1 - \log(12)) + \log\lambda + \frac{1}{2}\log(I(\lambda)) - \log p(W|\lambda), \tag{7}$$

where we used the expected Fisher information $I(\lambda) \equiv -\mathbb{E}[\frac{\partial^2 \log p(W|\lambda)}{\partial \lambda^2}]$.

The decision of merging two clusters is based on the minimum description length between two models: one composed by two separated clusters, M_2, and the other composed by two clusters merged, M_1. The description length of model M_2 consists of the sum of two description lengths (one for each cluster with parameters λ_1 and λ_2). The other model, M_1, has a description length with one parameter only. So, the minimum description length in this case is given by

$$\text{choose } M_i : i = \underset{i}{\operatorname{argmin}} \{DL(M_i)\}. \tag{8}$$

3.3 Algorithm Analysis

We claim that the proposed algorithm has two main features: it can adequately identify well separated clusters with arbitrarily shaped and densities, and it offers a deeper insight into the structure of touching clusters. In this section we will explain these two claims and present a few examples.

If one has data sets with separated clusters like the one presented in figure 2, single-link is not able to find a good partition of this data. Actually, single-link joins the two centered clusters into one cluster only and the outer cluster is split into two clusters, where one cluster has only four points (see figure 3).

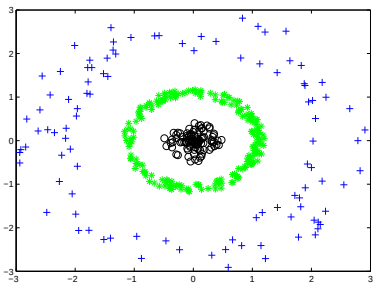


Fig. 2: Synthetic data set with three distinct clusters.

The new proposed algorithm, which we call SL-DID, can often find the correct number of clusters, in data sets where single-link fails. For example, in the dataset

presented in figure 2, SL-DID finds the three correct clusters, in spite of the clusters having different densities and arbitrary shapes (see figure 3).

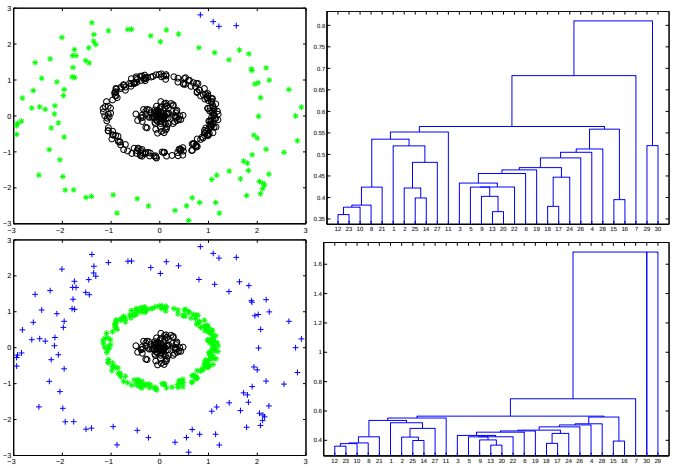


Fig. 3: *Top:* Partition produced by single-link in data set presented in figure 2 and dendrogram produced by single-link. *Bottom:* Partition produced by the proposed algorithm in the data set presented in figure 2 and corresponding dendrogram.

A disadvantage of typical hierarchical clustering algorithms is the fact that one often needs to know some *a priori* information about the true number of classes. This is, in practice, a parameter that needs to be fed into these algorithms, unless the user looks at the dendrogram and manually selects the appropriate number of clusters. But sometimes even the dendrogram does not give us good information on where to "cut". In the dendrogram presented on top of figure 3, the cut in the dendrogram can be made to obtain two or three clusters: the height of the corresponding lines of the dendrogram is similar, so one is uncertain whether there are two or three clusters present (this is in addition to the fact that the three clusters found by single-link are wrong). In SL-DID, on the other hand, the algorithm automatically stops at an appropriate number of clusters (which is 3 in the example above).

The use of a hierarchical clustering algorithm, like single-link or the proposed method, has some disadvantages when one has a dataset with touching clusters. Consider for example the Iris data set from the UCI Machine Learning Repository[1]. It is a data set with one cluster well separated from the other two which overlap each other. Hierarchical clustering techniques are able to find the well separated cluster, however the other two classes are combined into a single cluster. In this kind of data set it is better to use a gaussian mixture decomposition approach.

Furthermore, if we consider a data set with touching clusters like the one presented in figure 4, single-link is not able to find a good partition of this data. Actually, single-

[1] http://archive.ics.uci.edu/ml

link joins all the points into one cluster only. Even if we force single-link to stop in the true number of classes, it finds two clusters where one cluster has only one point, which means single-link mixed both classes in one single class (see figure 4) and cannot find the true structure of this data set.

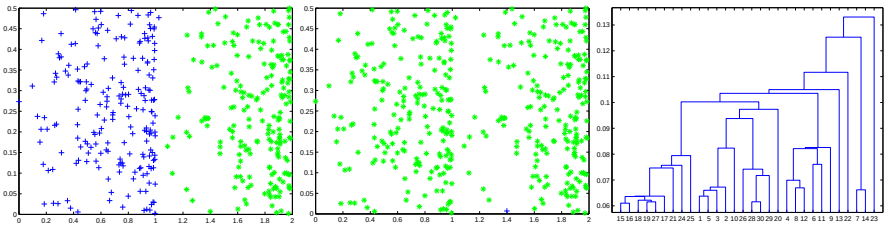


Fig. 4: *Left:* Synthetic data set with two distinct clusters. *Middle:* Partition produced by single-link in that data set. Note that all points are clustered together, except for one point with coordinates (1.4 , 0). *Right:* Dendrogram produced by single-link.

The proposed algorithm (SL-DID), in data sets with touching clusters, offers a deeper insight into the structure of touching clusters; it can often find a set of clusters such that each class is the union of a few clusters. For example, in the dataset presented in figure 4, SL-DID finds four clusters: two for each class (see figure 5). In this particular case, the increments in each class are not similar throughout the whole class, and SL-DID divides each class into two clusters. However, the gap between the two clusters of each class is much smaller than the gap between clusters of different classes. This information can be used to reconstruct the two original classes, as in figure 5. We can conclude that our algorithm is able to find classes that are composed by the union of small models based on dissimilarity increments.

4 Experimental Results

4.1 Datasets

To test the performance of the proposed method, we used 12 data sets: 8 synthetic data sets, and 4 real data sets from the UCI Machine Learning Repository[2]. The synthetic data sets were chosen to take into account a wide variety of situations: well-separated; gaussian and non-gaussian clusters; arbitrary shapes; and diverse cluster densities. These synthetic data sets are shown in figure 6.

We now describe the real data sets. The *Wisconsin Breast-Cancer* data set consists of 683 patterns represented by nine features and has two clusters. The *House Votes* data set consists of votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the Congressional Quarterly Almanac. It is composed by two clusters and only the patterns without missing values were considered, for a total

[2] http://archive.ics.uci.edu/ml

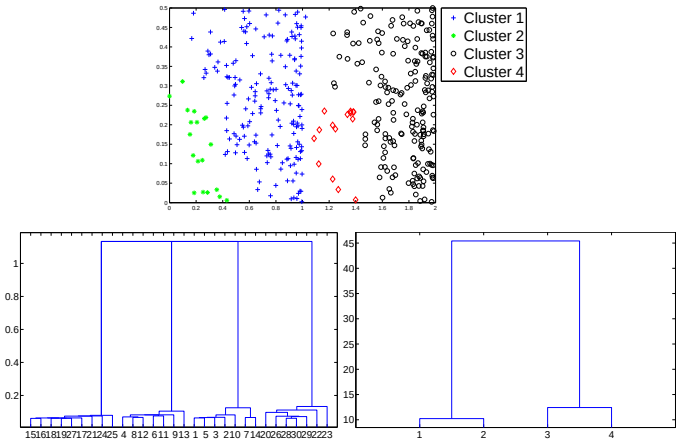


Fig. 5: *Top:* Partition produced by the proposed method in the synthetic data set presented in figure 4. *Bottom left:* Dendrogram produced during the successive mergings of SL-DID. Note that the presence of four clusters is obvious even with visual inspection, even though SL-DID stops automatically at four clusters. *Bottom right:* Dendrogram produced over the clusters presented in the top figure. As we can see clusters 1 and 2 should be considered as belonging to the same class and the same for clusters 3 and 4. However, the gap between clusters (1,2) and clusters (3,4) is huge, which indicates that we are in the presence of two classes: one formed by clusters 1 and 2, and the other formed by the clusters 3 and 4.

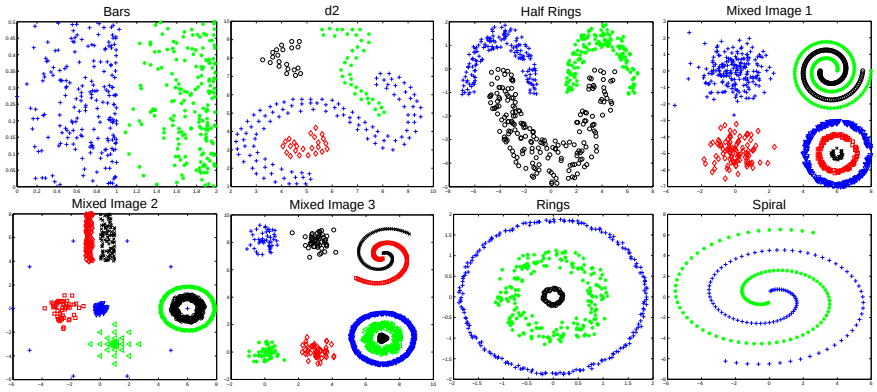


Fig. 6: Synthetic data sets

of 232 samples (125 democrats and 107 republicans). It was normalized to have unit variance. The *Iris* data set consists of three species of Iris plants (Setosa, Versicolor and Virginica). This data set is characterized by four features and 50 samples in each cluster. The *Wine* consists of the results of a chemical analysis of wines grown in the same region in Italy divided into three clusters with 59, 71 and 48 patterns described by 13 features.

4.2 Parameter Selection

Although our algorithm is parameter-free, we will compare it to other algorithms which have parameters that need to be chosen. To automatically select the parameters in those algorithms, we use the Graph-based Dissimilarity Increments Distribution index, or G-DID, which is a cluster validity index proposed by Fred and Jain [3]. G-DID is based on the minimum description length (MDL) of the graph-based representation of partition P. The selection among N partitions, produced by different values of parameters, is as follows

$$\text{Choose } P^i : i = \underset{j}{\operatorname{argmin}}\{\text{G-DID}(P^j)\}, \tag{9}$$

where G-DID$(P) = -\log \hat{f}(P) + \frac{k_P}{2}\log(n)$ is the graph description length, n is the number of samples (increments in our case), and $\hat{f}(P)$ is the probability of partition P, with k_P clusters, according to a Probabilistic Attributed Graph model taking into account the dissimilarity increments distribution (see [3] for details). In [3], the probability of the graph edges was estimated from an exponential model associated with each cluster.

4.3 Exprimental Results and Discussion

We compare the proposed method (SL-DID) to two algorithms also based in the dissimilarity increments distribution. The GMDID is a partitional clustering algorithm which uses a Gaussian mixture decomposition (GM) [2] as initialization and a likelihood ratio test to make the decision of whether to merge two clusters (see [1] for details). Another algorithm based on dissimilarity increments, SL-AGLO, was earlier proposed by Fred and Leitão [4]. We also compare our algorithm to the two most used hierarchical clustering algorithms: single-link (SL) and average-link (AL).

All the clustering methods using the dissimilarity increments concept find the number of clusters automatically and GM also finds the number of clusters automatically. However, GMDID and SL-AGLO have an additional parameter, so we compute partitions for several values of parameters and choose among them as described in section 4.2. GMDID has a significance level α to perform the likelihood ratio test and we used 1%, 5%, 10% and 15% to decide whether two clusters should be merged or not. The SL-AGLO algorithm has an isolation parameter which is a threshold set in the tail of the exponential distribution of the dissimilarity increments of a cluster (with parameter the inverse of the mean of the increments of that cluster). We used values ranging from the mean of the exponential distribution to 10 times this mean for this threshold, and the choice of the best value was also done using G-DID. For SL and AL we obtain several

partitions for different values of number of clusters and used also the G-DID to choose the best partition.

We assess the quality of each resulting partition P using the consistency index (CI), which is the percentage of agreement between P and the ground truth information. In other words, it is the percentage of correctly clustered points. Table 2 summarizes the results obtained with CI.

Table 2: Consistency index (CI) values of the partitions found by the algorithms. The values in parenthesis correspond to the number of clusters found by each algorithm. The first column contains the true number of clusters (Nc) of each data set.

	Nc	GM	GMDID	SL-AGLO	SL-DID	SL	AL
Bars	2	0.5000 (10)	0.7700 (3)	0.9100 (4)	0.9150 (4)	0.5075 (2)	**0.9925** (2)
d2	4	0.3400 (13)	0.9050 (5)	**1.0000** (4)	**1.0000** (4)	**1.0000** (4)	0.4900 (8)
Half Rings	3	0.2980 (17)	0.8620 (6)	**1.0000** (3)	**1.0000** (3)	0.9500 (4)	0.6000 (2)
Mixed Image 1	7	0.4420 (24)	0.5890 (15)	**0.9560** (8)	0.9190 (9)	0.8440 (11)	0.5810 (3)
Mixed Image 2	8	0.4709 (20)	**1.0000** (8)	0.9743 (10)	**1.0000** (8)	0.5142 (12)	0.5819 (12)
Mixed Image 3	9	0.4588 (26)	0.9141 (10)	**1.0000** (9)	0.9459 (11)	0.8188 (11)	0.5882 (6)
Rings	3	0.2444 (27)	**1.0000** (3)	**1.0000** (3)	**1.0000** (3)	**1.0000** (3)	0.5289 (2)
Spiral	2	0.1400 (27)	**1.0000** (2)	**1.0000** (2)	**1.0000** (2)	**1.0000** (2)	0.5200 (2)
Breast Cancer	2	0.5593 (5)	0.7467 (3)	0.5783 (24)	0.3748 (12)	0.6515 (3)	**0.9385** (5)
House Votes	2	0.8405 (2)	0.8405 (2)	0.5345 (3)	0.6336 (3)	0.5302 (2)	**0.8836** (2)
Iris	3	**0.8000** (4)	0.6667 (3)	0.4800 (6)	0.6667 (2)	0.6667 (2)	0.6667 (2)
Wine	3	0.5112 (10)	0.5169 (8)	0.2753 (11)	0.3034 (10)	0.5337 (7)	**0.6573** (4)

Overall, the use of dissimilarity increments incorporated in a clustering algorithm shows better results than the ones using only pairwise dissimilarities in synthetic data sets.

As mentioned above, our algorithm starts by merging points using the same criterion as single-link, until one of the clusters being tested has 6 or more points. Once that happens, we use a criterion to merge or not clusters using the DID presented in section 2.1. Table 2 shows that this criterion improves the results over single-link. It remains to be seen whether we can use a similar concept to improve the results of average-link. This is a topic we will research in the near future.

Also, this table shows that average-link is the best overall method for the four real data sets considered if one takes the produced clusters as the final partition. However, note that we forced single-link and average-link to stop at two clusters.

We alerted above to the fact that the true classes in the data can have varying internal structure, as in the data set of figure 4. In this case, even if the number of clusters is incorrect, one should test whether the true classes can be recovered as the union of some of the found clusters. For this purpose, we use another measure, which computes the percentage of correctly clustered points if one represents the classes as the union of clusters, where each cluster can only be used for one class. We denote this second measure by CI*, due to its similarity with the Consistency Index. Table 3 summarizes the results.

Table 3: CI* values of the partitions found by the algorithms. The first two columns correspond to the number of patterns (N) and the true number of clusters (Nc) of each data set.

	N	Nc	GM	GMDID	SL-AGLO	SL-DID	SL	AL
Bars	400	2	0.9775	0.9775	0.9625	**1.0000**	0.5075	0.9925
d2	200	4	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9400
Half Rings	500	3	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.6000
Mixed Image 1	1000	7	0.7850	0.7860	**0.9990**	**0.9990**	0.8490	0.5810
Mixed Image 2	739	8	**1.0000**	**1.0000**	0.9986	**1.0000**	0.5237	0.5900
Mixed Image 3	850	9	0.9459	0.9459	**1.0000**	**1.0000**	0.8235	0.5882
Rings	450	3	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.5289
Spiral	200	2	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.5200
Breast Cancer	683	2	0.7745	0.7980	0.9063	0.6691	0.6530	**0.9414**
House Votes	232	2	0.8405	0.8405	0.6983	0.7112	0.5345	**0.8836**
Iris	150	3	**0.9667**	0.6667	0.6933	0.6667	0.6667	0.6667
Wine	178	3	0.5618	0.5843	0.5169	0.4775	0.6517	**0.6910**

We noticed earlier that SL-DID is very similar to SL-AGLO in performance. However, if we take into account that a class can be composed by several models of increments, SL-DID is the best algorithm in terms of the synthetic data sets: see table 3.

In terms of real data sets, the average-link outperforms all the other algorithms. SL-DID has some problems in data sets with overlapped classes because it starts by assigning each pattern to a cluster and merges the closest clusters until we have a minimum number of samples to estimate the DID. If the classes have an overlap it is likely that the initial process of merging the closest clusters will merge together points belonging to different classes. In future work we will try to overcome this difficulty.

5 Conclusions

The dissimilarity increments distribution was recently derived and used to create a partitional clustering algorithm [1]. The initialization of that algorithm was a Gaussian mixture decomposition which has some difficulties in identify arbitrarily shaped clusters and sometimes it combines patterns that should be in different components. In this paper we propose to use the same distribution but in a hierarchical point of view.

The new proposed method is a parameter-free algorithm and identifies classes as the union of dissimilarity increments models. Experimental results show that it performs very well for well separated data sets in the identification of classes. However, in real data sets with overlapping classes, it tends to mix the classes and form a single model for the increments. In future work we will improve the approach presented in this paper, so it can work well in data sets with touching clusters.

6 Acknowledgments

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). This work was partially supported by

the Portuguese Foundation for Science and Technology (FCT), scholarship number SFRH/BD/39642/2007, and grant PTDC/EIA-CCO/103230/2008.

References

1. Aidos, H., Fred, A.: On the distribution of dissimilarity increments. In: IbPRIA 2011 (to appear) (2011)
2. Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of finite mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 381–396 (2002)
3. Fred, A., Jain, A.: Cluster validation using a probabilistic attributed graph. In: Proceedings of the 19th International Conference on Pattern Recognition (ICPR 2008). Tampa, Florida, USA (2008)
4. Fred, A., Leitão, J.: A new cluster isolation criterion based on dissimilarity increments. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 944–958 (2003)
5. Grünwald, P.D.: Advances in Minimum Description Length: Theory and Applications, chap. A Tutorial Introduction to the Minimum Description Length Principle. MIT Press (2005)
6. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, 2nd edn. (2009)
7. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31, 264–323 (1999)
8. Johnson, N.L., Kotz, S., Balakrishnan, N.: Continuous Univariate Distributions, Applied Probability and Statistics, vol. 1. John Wiley & Sons Ltd., 2 edn. (1994)
9. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Elsevier Academic Press, 2 edn. (2003)
10. Ueda, N., Nakano, R., Ghahramani, Z., Hinton, G.E.: Smem algorithm for mixture models. Neural Computation 12(9), 2109–2128 (2000)

[30] - (no technical report)

Martins, A., Smith, N., Xing, E., Aguiar, P., Figueiredo, M. Online Learning of Structured Predictors with Multiple Kernels, AISTATS (2011).

Online Learning of Structured Predictors with Multiple Kernels

André F. T. Martins[*†] **Noah A. Smith**[*] **Eric P. Xing**[*]

Pedro M. Q. Aguiar[‡] **Mário A. T. Figueiredo**[†]

[*]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

[‡]Instituto de Sistemas e Robótica, Instituto Superior Técnico, Lisboa, Portugal

[†]Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa, Portugal

Abstract

Training structured predictors often requires a considerable time selecting features or tweaking the kernel. Multiple kernel learning (MKL) sidesteps this issue by embedding the kernel learning into the training procedure. Despite the recent progress towards efficiency and scalability of MKL algorithms, the structured output case remains an open research front. We propose a new family of online proximal algorithms able to tackle many variants of MKL and group-LASSO, and for which we show regret, convergence, and generalization bounds. Experiments on handwriting recognition and dependency parsing illustrate the success of the approach.

1 INTRODUCTION

Structured prediction problems are characterized by strong interdependence among the output variables, usually with sequential, graphical, or combinatorial structure. Despite all the advances toward a unified formalism encompassing different learning objectives (Bakır et al., 2007), obtaining good predictors still requires a large effort in feature/kernel design and tuning (often done via cross-validation). Because discriminative training of structured predictors can be quite slow, especially in large-scale settings, it is appealing to learn the kernel function simultaneously.

In multiple kernel learning (MKL, Lanckriet et al. 2004; Bach et al. 2004), the kernel is learned as a linear combination of prespecified base kernels. This framework has been made scalable with the advent of wrapper-based methods, in which a standard learning problem (*e.g.*, an SVM) is repeatedly solved in an inner loop up to a prescribed accuracy (Sonnenburg et al., 2006; Rakotomamonjy et al., 2008; Kloft et al., 2010). Unfortunately, extending such methods to structured prediction raises practical hurdles: since the output space is large, so are the kernel matrices, and the number of support vectors. Moreover, since it is typically prohibitive to tackle the inner learning problem in its batch form, one often needs to resort to online algorithms (Ratliff et al., 2006; Vishwanathan et al., 2006; Collins et al., 2008); the latter are fast learners but slow optimizers (Bottou and Bousquet, 2007), hence using them in the inner loop with early stopping can misguide the overall MKL optimization.

In this paper, we overcome the above difficulty by proposing a stand-alone online MKL algorithm which iterates between subgradient and proximal steps. The algorithm, which when applied to structured prediction is termed SPOM (*Structured Prediction by Online MKL*), has important advantages: *(i)* it is simple, flexible, and compatible with sparse and non-sparse MKL, *(ii)* it is adequate for structured prediction, *(iii)* it offers regret, convergence, and generalization guarantees. Our approach extends and kernelizes the forward-backward splitting scheme FOBOS (Duchi and Singer, 2009), whose regret bound we improve.

Our paper is organized as follows: after reviewing structured prediction and MKL (§2), we present a class of online proximal algorithms which handle composite regularizers with multiple proximal steps (§3). We derive convergence rates and show how these algorithms are applicable in MKL, group-LASSO, and other structural sparsity formalisms. In §4, we apply this algorithm for structured prediction (yielding SPOM) in two experimental testbeds: sequence labeling for handwritten text recognition, and natural language dependency parsing. We show the potential of SPOM by learning combinations of kernels from tens of thousands of training instances, with encouraging results in terms of runtimes and accuracy.

Appearing in Proceedings of the 14^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15.

2 STRUCTURED PREDICTION AND MULTIPLE KERNEL LEARNING

2.1 Inference and Learning with Structured Outputs

We denote by $\mathcal{X}$ and $\mathcal{Y}$ the input and output sets, respectively. Given an input $x \in \mathcal{X}$, we let $\mathcal{Y}(x) \subseteq \mathcal{Y}$ be its set of admissible outputs; in structured prediction, this set is assumed to be structured and exponentially large. Two important examples of structured prediction problems are: sequence labeling, in which x is an observed sequence and each $y \in \mathcal{Y}(x)$ is a corresponding sequence of labels; and natural language parsing, where x is a string and each $y \in \mathcal{Y}(x)$ is a possible parse tree that spans that string.

Let $\mathcal{U} \triangleq \{(x, y) \mid x \in \mathcal{X}, y \in \mathcal{Y}(x)\}$ denote the set of admissible input-output pairs. In supervised learning, we are given a labeled dataset $\mathcal{D} \triangleq \{(x_1, y_1), \ldots, (x_N, y_N)\} \subseteq \mathcal{U}$, and the goal is to learn a compatibility function $f_{\boldsymbol{\theta}} : \mathcal{U} \to \mathbb{R}$ that allows to make predictions on unseen data via

$$x \mapsto \hat{y}(x) \triangleq \arg \max_{y \in \mathcal{Y}(x)} f_{\boldsymbol{\theta}}(x, y). \tag{1}$$

Problem (1) is called *inference* (*decoding*) and, in strutured prediction, often involves combinatorial optimization (*e.g.*, dynamic programming). In this paper, we consider linear functions, $f_{\boldsymbol{\theta}}(x, y) \triangleq \langle \boldsymbol{\theta}, \boldsymbol{\phi}(x, y) \rangle$, where $\boldsymbol{\theta}$ is a parameter vector and $\boldsymbol{\phi}(x, y)$ a feature vector. To be general, we let these vectors live in a Hilbert space $\mathcal{H}$, with reproducing kernel $K : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$. In the sequel, features will be used implicitly or explicitly, as convenience determines.

We want $f_{\boldsymbol{\theta}}$ to generalize well, *i.e.*, given a cost function $c : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ (with $c(y, y) = 0$, $\forall y$) we want the corresponding inference criterion to have low expected risk $\mathbb{E}_{X,Y}\, c(Y, \hat{y}(X))$. To achieve this, one casts learning as the minimization of a regularized empirical risk functional,

$$\min_{f_{\boldsymbol{\theta}} \in \mathcal{H}} \lambda\, \Omega(f_{\boldsymbol{\theta}}) + \frac{1}{N} \sum_{i=1}^{N} L(f_{\boldsymbol{\theta}}; x_i, y_i), \tag{2}$$

where $\Omega : \mathcal{H} \to \mathcal{R}_+$ is a regularizer, $\lambda \geq 0$ is the regularization constant, and $L : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a convex loss function. Common choices are the logistic loss, in *conditional random fields* (CRF, Lafferty et al. 2001), and the structured hinge loss, in structural *support vector machines* (SVM, Taskar et al. 2003; Tsochantaridis et al. 2004):

$$L_{\text{CRF}}(f_{\boldsymbol{\theta}}; x, y) \triangleq \log \textstyle\sum_{y' \in \mathcal{Y}(x)} \exp(\delta f_{\boldsymbol{\theta}}(x, y', y)), \tag{3}$$

$$L_{\text{SVM}}(f_{\boldsymbol{\theta}}; x, y) \triangleq \max_{y' \in \mathcal{Y}(x)} \delta f_{\boldsymbol{\theta}}(x, y', y) + c(y', y), \tag{4}$$

where $\delta f_{\boldsymbol{\theta}}(x, y', y) \triangleq f_{\boldsymbol{\theta}}(x, y') - f_{\boldsymbol{\theta}}(x, y)$.

If the regularizer is $\Omega(f_{\boldsymbol{\theta}}) = \frac{1}{2}\|\boldsymbol{\theta}\|^2$ (ℓ_2-regularization), the solution of (2) can be expressed as a kernel expansion (structured version of the representer theorem, Hofmann et al. 2008, Corollary 13). We next discuss alternative forms of regularization that take into consideration another level of structure—now in the feature space.

2.2 Block-Norm Regularization and Kernel Learning

Selecting relevant features or picking a good kernel are ubiquitous problems in statistical learning, both of which have been addressed with *sparsity-promoting regularizers*. We first illustrate this point for the explicit feature case. Often, features exhibit a natural block structure: for example, many models in NLP consider *feature templates*—these are binary features indexed by each word w in the vocabulary, by each part-of-speech tag t, by each pair (w, t), etc. Each of these templates correspond to a *block* (also called a *group*) in the feature space $\mathcal{H}$. Thus, $\mathcal{H}$ is endowed with a block structure, where each block (indexed by $m = 1, \ldots, M$) is itself a "smaller" feature space $\mathcal{H}_m$; formally, $\mathcal{H}$ is a direct sum of Hilbert spaces: $\mathcal{H} = \bigoplus_{m=1}^{M} \mathcal{H}_m$.

Group-LASSO. Consider the goal of learning a model in the presence of many irrelevant feature templates. A well-known criterion is the group-LASSO (Bakin, 1999; Yuan and Lin, 2006), which uses the following block-norm regularizer: $\Omega_{\text{GL}}(f_{\boldsymbol{\theta}}) = \sum_{m=1}^{M} \|\boldsymbol{\theta}_m\|$. This can be seen as the ℓ_1-norm of the ℓ_2-norms: it promotes sparsity w.r.t. the number of templates (groups) that are selected. When Ω_{GL} is used in (2), the following happens within the mth group: either the optimal $\boldsymbol{\theta}_m^*$ is identically zero—in which case the entire group is discarded—or it is non-sparse.

Sparse MKL. In MKL (Lanckriet et al. 2004), we learn a kernel as a linear combination $K = \sum_{m=1}^{M} \beta_m K_m$ of M prespecified base kernels $\{K_1, \ldots, K_M\}$, where the coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_M)$ are constrained to the simplex $\Delta^M \triangleq \{\boldsymbol{\beta} \geq \mathbf{0} \mid \|\boldsymbol{\beta}\|_1 = 1\}$. This is formulated as an outer minimization of (2) w.r.t. $\boldsymbol{\beta} \in \Delta^M$:

$$\min_{\boldsymbol{\beta} \in \Delta^M} \min_{f_{\boldsymbol{\theta}} \in \mathcal{H}} \frac{\lambda}{2} \sum_{m=1}^{M} \beta_m \|\boldsymbol{\theta}_m\|^2 + \frac{1}{N} \sum_{i=1}^{N} L\left(\sum_{m=1}^{M} \beta_m f_{\boldsymbol{\theta}_m}; x_i, y_i \right). \tag{5}$$

Remarkably, as shown by Bach et al. (2004) and Rakotomamonjy et al. (2008), this joint optimization over $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ can be transformed into a single optimization of the form (2) with a block-structured regularizer $\Omega_{\text{MKL}}(f_{\boldsymbol{\theta}}) = \frac{1}{2}(\sum_{m=1}^{M} \|\boldsymbol{\theta}_m\|)^2$. Note that this coincides with the square of the group-LASSO regularizer; in fact, the two problems are equivalent up to a change of λ (Bach, 2008a). Hence, this MKL formulation promotes sparsity in the number of selected kernels (*i.e.*, only a few nonzero entries in $\boldsymbol{\beta}$).

Non-Sparse MKL. A more general MKL formulation (not necessarily sparse) was recently proposed by Kloft et al. (2010). Define, for $p \geq 1$, the set $\Delta_p^M \triangleq \{\boldsymbol{\beta} \geq \mathbf{0} \mid \|\boldsymbol{\beta}\|_p = 1\}$. Then, by modifying the constraint in (5) to $\boldsymbol{\beta} \in \Delta_p^M$, the resulting problem can be again transformed

in one of the form (2) with the block-structured regularizer:

$$\Omega_{\text{MKL},q}(f_{\boldsymbol{\theta}}) = \frac{1}{2}\left(\sum_{m=1}^{M} \|\boldsymbol{\theta}_m\|^q\right)^{2/q} \triangleq \frac{1}{2}\|\boldsymbol{\theta}\|_{2,q}^2, \quad (6)$$

where $q = 2p/(p+1)$. The function $\|.\|_{2,q}$ satisfies the axioms of a norm, and is called the $\ell_{2,q}$ *mixed norm*. Given a solution $\boldsymbol{\theta}^*$, the optimal kernel coefficients can be computed as $\beta_m^* \propto \|\boldsymbol{\theta}_m^*\|^{2-q}$. Note that the case $p = q = 1$ corresponds to sparse MKL and that $p = \infty, q = 2$ corresponds to standard ℓ_2-regularization with a sum-of-kernels.

2.3 Learning the Kernel in Structured Prediction

Up to now, all formulations of MKL apply to classification problems with small numbers of classes. The algorithmic challenges that prevent the straightforward application of these formulations to structured prediction will be discussed in §2.4; here, we formally present our MKL formulation for structured prediction.

In structured prediction, model factorization assumptions are needed to make the inference problem (1) tractable. This can be accomplished by defining a set of *parts* over which the model decomposes. Suppose, for example, that outputs correspond to vertex labelings in a Markov network (V, E). Then, we may let each part be either a vertex or an edge, and write the feature vector as $\boldsymbol{\phi}(x, y) = (\boldsymbol{\phi}_V(x, y), \boldsymbol{\phi}_E(x, y))$, with

$$\boldsymbol{\phi}_V(x, y) = \sum_{i \in V} \boldsymbol{\psi}_V(x, i) \otimes \boldsymbol{\zeta}_V(y_i) \quad (7)$$

$$\boldsymbol{\phi}_E(x, y) = \sum_{ij \in E} \boldsymbol{\psi}_E(x, ij) \otimes \boldsymbol{\zeta}_E(y_i, y_j), \quad (8)$$

where $\otimes$ denotes the Kronecker product between feature vectors, $\boldsymbol{\psi}_V$ and $\boldsymbol{\psi}_E$ are input feature vectors (which may depend globally on x), and $\boldsymbol{\zeta}_V$ and $\boldsymbol{\zeta}_E$ are *local* output feature vectors which only look, respectively, at a single vertex and a single edge. A common simplifying assumption is to let $\boldsymbol{\zeta}_V$ be the identity feature mapping, $\boldsymbol{\psi}_E \equiv 1$, and to define $\boldsymbol{\zeta}_E$ as the identity feature mapping scaled by $\beta_0 > 0$. We can then learn the kernel $K_{X,V}((x, i), (x', i')) \triangleq \langle \boldsymbol{\psi}_V(x, i), \boldsymbol{\psi}_V(x', i') \rangle$ as a combination of basis kernels $\{K_{X,V,m}\}_{m=1}^M$. This yields a kernel decomposition of the form

$$K((x, y), (x', y')) = \beta_0 \cdot |\{ij, i'j' \in E : y_i = y'_{i'}, y_j = y'_{j'}\}| + \sum_{i,i' \in V : y_i = y'_{i'}} \sum_{m=1}^{M} \beta_m K_{X,V,m}((x, i), (x', i')).$$

In our sequence labeling experiments (§4), vertices and edges correspond to label unigrams and bigrams. We explore two strategies: learning $\beta_1, \ldots, \beta_M$, with $\beta_0 = 1$ fixed, or also learning β_0.

2.4 Existing MKL Algorithms

Early approaches to MKL (Lanckriet et al., 2004; Bach et al., 2004) considered the dual of (5) in the form of a second order cone program, thus were limited to small/medium scale problems. Subsequent work focused on scalability: Sonnenburg et al. (2006) proposes a semi-infinite LP formulation and a cutting plane algorithm; Rakotomamonjy et al. (2008) proposes a gradient-based method (*SimpleMKL*) for optimizing the kernel coefficients $\boldsymbol{\beta}$; Kloft et al. (2010) performs Gauss-Seidel alternate optimization of $\boldsymbol{\beta}$ and of SVM instances; Xu et al. (2009) proposes an extended level method.

The methods mentioned in the previous paragraph are all *wrapper-based algorithms*: they repeatedly solve problems of the form (2) (or smaller chunks of it, as in Kloft et al. 2010). Although warm-starting may offer considerable speed-ups, convergence relies on the exactness (or prescribed accuracy in the dual) of these solutions, which constitutes a serious obstacle when using such algorithms for structured prediction. Large-scale solvers for structured SVMs lack strong convergence guarantees; the best methods require $O(\frac{1}{\epsilon})$ rounds to converge to ϵ-accuracy. Sophisticated second-order methods are intractable, since the kernel matrix is exponentially large and hard to invert; furthermore, there are typically many support vectors, since they are indexed by elements of $\mathcal{Y}(x_i)$.

In contrast, we tackle (5) in *primal* form. Rather than repeatedly calling off-the-shelf solvers for (2), we propose a stand-alone online algorithm with runtime comparable to that of solving a *single* instance of (2) by fast online methods. This paradigm shift paves the way for extending MKL to structured prediction, a vast unexplored territory.

3 ONLINE PROXIMAL ALGORITHMS FOR KERNEL LEARNING

3.1 An Online Proximal Gradient Scheme

The general algorithmic scheme that we propose and analyze in this paper is presented as Alg. 1. It deals (in an online fashion[1]) with problems of the form

$$\min_{\boldsymbol{\theta} \in \Theta} \lambda \Omega(\boldsymbol{\theta}) + \tfrac{1}{N} \textstyle\sum_{i=1}^{N} L(\boldsymbol{\theta}; x_i, y_i), \quad (9)$$

where $\Theta \subseteq \mathcal{H}$ is a convex set and the regularizer Ω has a composite form $\Omega(\boldsymbol{\theta}) = \sum_{j=1}^{J} \Omega_j(\boldsymbol{\theta})$. This encompasses all formulations described in §2.1–2.2: standard ℓ_2-regularized SVMs and CRFs, group LASSO, and sparse and non-sparse variants of MKL. For all these, we have $\Theta = \mathcal{H}$ and $J = 1$. Moreover, with $J > 1$ it allows new variants of block-norm regularization, as we will discuss in §3.2.

[1] For simplicity, we focus on the pure online setting, *i.e.*, each parameter update uses a single observation; analogous algorithms may be derived for the batch and mini-batch cases.

Alg. 1 is similar to stochastic gradient descent (SGD, Bottou 1991), in that it also performs "noisy" (sub-)gradient steps[2] by looking only at a single instance (line 4). Hence, as SGD, it is also suitable for problems with large N. The difference is that these subgradients are only w.r.t. the loss function L, *i.e.*, they ignore the regularizer Ω.

In turn, each round of Alg. 1 makes J proximal steps, one per each term Ω_j (line 7). Given a function $\Phi : \mathcal{H} \to \mathbb{R}$, the Φ-proximity operator (Moreau, 1962) is defined as:

$$\text{prox}_{\Phi}(\boldsymbol{\theta}) \quad = \quad \arg\min_{\boldsymbol{\xi}\in\mathcal{H}} \frac{1}{2}\|\boldsymbol{\xi}-\boldsymbol{\theta}\|^2 + \Phi(\boldsymbol{\xi}). \quad (10)$$

For sparsity-promoting regularizers, such as $\Phi(\boldsymbol{\xi}) = \|\boldsymbol{\xi}\|_1$ or $\Phi(\boldsymbol{\xi}) = \|\boldsymbol{\xi}\|_{2,1}$, prox_{Φ} has a shrinkage and thresholding effect on the parameter vector, pushing Alg. 1 to return sparse solutions. This does not usually happen in standard SGD: since those regularizers are non-differentiable at the origin, each subgradient step in SGD causes oscillation and the final solution is rarely sparse. Contrarily, Alg. 1 is particularly appropriate for learning sparse models.

Finally, the projection step (line 9) can be used as a trick to accelerate convergence: for example, if we take the unconstrained version of (9) and we know beforehand that the optimum lies in a convex set Θ, then the constraint $\boldsymbol{\theta} \in \Theta$ is "vacuous," but line 9 ensures that each iterate $\boldsymbol{\theta}_t$ is confined to a bounded region containing the optimum. This idea is also used in PEGASOS (Shalev-Shwartz et al., 2007).

Before analyzing Alg. 1, we remark that it includes, as particular cases, many well-known online learners:

- if $\Omega = 0$ and $\eta_t \propto \frac{1}{\sqrt{t}}$, Alg. 1 is the online projected subgradient algorithm of Zinkevich (2003);
- if $\Omega = 0$, $L = L_{\text{SVM}} + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2$, and $\eta_t = \frac{1}{\lambda t}$, Alg. 1 becomes PEGASOS, a popular algorithm for learning SVMs that has been extended to structured prediction (Shalev-Shwartz et al., 2007, 2010);
- If $\Omega(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1$, Alg. 1 was named truncated gradient descent and studied by Langford et al. (2009);
- If $J = 1$, Alg. 1 coincides with FOBOS (Duchi and Singer, 2009), which was used for learning SVMs and also for group-LASSO (but not for structured prediction).

In §3.4, we show how to kernelize Alg. 1 and apply it to sparse MKL. The case $J > 1$ has applications in variants of MKL or group-LASSO with composite regularizers (Tomioka and Suzuki, 2010; Friedman et al., 2010; Bach, 2008b; Zhao et al., 2008). In some of those cases, the proximity operators of $\Omega_1, ..., \Omega_J$ are easier to compute than their sum Ω, making Alg. 1 more suitable than FOBOS.

[2] Given a convex function $\Phi : \mathcal{H} \to \mathbb{R}$, its *subdifferential* at $\boldsymbol{\theta}$ is the set $\partial\Phi(\boldsymbol{\theta}) \triangleq \{\boldsymbol{g} \in \mathcal{H} \mid \forall \boldsymbol{\theta}' \in \mathcal{H},\ \Phi(\boldsymbol{\theta}') - \Phi(\boldsymbol{\theta}) \geq \langle \boldsymbol{g}, \boldsymbol{\theta}' - \boldsymbol{\theta}\rangle\}$, the elements of which are the *subgradients*.

Algorithm 1 Online Proximal Algorithm

1: **input:** dataset $\mathcal{D}$, parameter λ, number of rounds T, learning rate sequence $(\eta_t)_{t=1,...,T}$
2: initialize $\boldsymbol{\theta}_1 = \mathbf{0}$; set $N = |\mathcal{D}|$
3: **for** $t = 1$ **to** T **do**
4: take training pair (x_t, y_t) and obtain a subgradient $\boldsymbol{g} \in \partial L(\boldsymbol{\theta}_t; x_t, y_t)$
5: $\tilde{\boldsymbol{\theta}}_t = \boldsymbol{\theta}_t - \eta_t \boldsymbol{g}$ (gradient step)
6: **for** $j = 1$ **to** J **do**
7: $\tilde{\boldsymbol{\theta}}_{t+\frac{j}{J}} = \text{prox}_{\eta_t \lambda \Omega_j}(\tilde{\boldsymbol{\theta}}_{t+\frac{j-1}{J}})$ (proximal step)
8: **end for**
9: $\boldsymbol{\theta}_{t+1} = \Pi_{\Theta}(\tilde{\boldsymbol{\theta}}_{t+1})$ (projection step)
10: **end for**
11: **output:** the last model $\boldsymbol{\theta}_{T+1}$ or the averaged model $\bar{\boldsymbol{\theta}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\theta}_t$

3.2 Proximity Operators of Block-Norm Regularizers

For Alg. 1 to handle the MKL and group-LASSO problems (described in §2.2), it needs to compute the proximal steps for block-norm regularizers. The following proposition (proved in App. A) is crucial for this purpose.

Proposition 1 *If $\Phi(\boldsymbol{\theta}) = \varphi((\|\boldsymbol{\theta}_m\|)_{m=1}^M)$ depends on each block only through its ℓ_2-norm, where $\varphi : \mathbb{R}^M \to \mathbb{R}$, then, $[\text{prox}_{\Phi}(\boldsymbol{\theta})]_m = [\text{prox}_{\varphi}(\|\boldsymbol{\theta}_1\|, \ldots, \|\boldsymbol{\theta}_M\|)]_m(\boldsymbol{\theta}_m/\|\boldsymbol{\theta}_m\|)$.*

Hence, any $\ell_{2,q}^r$-proximity operator can be reduced to an ℓ_q^r one: its effect is to scale the weights of each block by an amount that depends on the latter. Examples follow.

Group-LASSO. This corresponds to $q = r = 1$, so we are left with the problem of computing the $\tau\|\cdot\|_1$-proximity operator, which has a well-known closed form solution: the soft-threshold function (Donoho and Johnstone, 1994),

$$\text{prox}_{\tau\|\cdot\|_1}(\boldsymbol{b}) = \text{soft}(\boldsymbol{b}, \tau), \quad (11)$$

where $[\text{soft}(\boldsymbol{b}, \tau)]_k \triangleq \text{sgn}(b_k) \cdot \max\{0, |b_k| - \tau\}$.

Sparse MKL. This corresponds to $q = 1$, $r = 2$, and there are two options: one is to transform the problem back into group-LASSO, by removing the square from Ω_{MKL} (as pointed out in §2, these two problems are equivalent in the sense that they have the same regularization path); the other option (that we adopt) is to tackle Ω_{MKL} directly.[3] Prop. 1 enables reducing the evaluation of a $\|\cdot\|_{2,1}^2$-proximity operator to that of a *squared* ℓ_1. However the squared ℓ_1 is not separable (unlike ℓ_1), hence the proximity operator cannot be evaluated coordinatewise as in (11). This apparent difficulty has led some authors (*e.g.*, Suzuki and Tomioka 2009) to stick with the first option. However, despite the non-separability of ℓ_1^2, this proximal step can still be efficiently computed, as shown in Alg. 2. This algorithm requires

[3] This makes possible the comparison with other MKL algorithms, for the same values of λ, as reported in §4.

Algorithm 2 Proximity Operator of ℓ_1^2

input: vector $\mathbf{x} \in \mathbb{R}^M$ and parameter $\lambda > 0$
sort the entries of $|\mathbf{x}|$ into $\mathbf{y}$ (yielding $y_1 \geq \ldots \geq y_M$)
set $\rho = \max \left\{ j \in \{1, ..., M\} \,|\, y_j - \frac{\lambda}{1+j\lambda} \sum_{r=1}^{j} y_r > 0 \right\}$
output: $\mathbf{z} = \text{soft}(\mathbf{x}, \tau)$, where $\tau = \frac{\lambda}{1+\rho\lambda} \sum_{r=1}^{\rho} y_r$

sorting the weights of each group, which has $O(M \log M)$ cost. Correctness of this algorithm is shown in App. E.[4]

Non-Sparse MKL. For the case $q \geq 1$ and $r = 2$, a direct evaluation of $\text{prox}_{\tau\|\cdot\|_{2,q}^2}$ is more involved. It seems advantageous to transform this problem into an equivalent one, which uses a separable $\ell_{2,q}^q$ regularizer instead (the two problems are also equivalent up to a change in the regularization constant). The resulting proximal step amounts to solving (on b) M scalar equations of the form $b - b_0 + \tau q b^{q-1} = 0$, also valid for $q \geq 2$ (unlike the method described by Kloft et al. 2010). This can be done very efficiently using Newton's method.

Other variants. Many other variants of MKL and group-LASSO can be handled by Alg. 1, with $J > 1$. For example, the elastic net MKL (Tomioka and Suzuki, 2010) uses a sum of two regularizers, $\frac{\sigma}{2}\|.\|^2 + \frac{1-\sigma}{2}\|.\|_{2,1}^2$. In hierarchical LASSO and group-LASSO with overlaps (Bach, 2008b; Zhao et al., 2008; Jenatton et al., 2009), each feature may appear in more than one group. Alg. 1 handles these problems seamlessly by enabling a proximal step for each group.[5] Sparse group-LASSO (Friedman et al., 2010) simultaneously promotes group-sparsity and sparsity *within* each group, via $\sigma\|.\|_{2,1} + (1-\sigma)\|.\|_1$ regularization; Alg. 1 can handle this regularizer by using two proximal steps, both involving simple soft-thresholding: one at the group level, and another within each group.

3.3 Regret, Convergence, and Generalization Bounds

We next show that, for a convex loss L and under standard assumptions, Alg. 1 converges up to ϵ precision, with high confidence, in $O(1/\epsilon^2)$ iterations. If L or Ω are strongly convex,[6] this bound is improved to $\tilde{O}(1/\epsilon)$, where $\tilde{O}$ hides logarithmic terms. Our proofs combine tools of online convex programming (Zinkevich, 2003; Hazan et al., 2007) and classical results about proximity operators (Moreau, 1962). The key is the following lemma (proved in App. B).

Lemma 2 *Assume that L is convex and G-Lipschitz[7] on Θ, and that $\Omega = \sum_{j=1}^{J} \Omega_j$ satisfies the following conditions:* **(i)** *each Ω_j is convex;* **(ii)** *$\forall \boldsymbol{\theta} \in \Theta$, $\forall j' < j$, $\Omega_{j'}(\boldsymbol{\theta}) \geq \Omega_{j'}(\text{prox}_{\lambda\Omega_j}(\boldsymbol{\theta}))$ (each proximity operator $\text{prox}_{\lambda\Omega_j}$ does not increase the previous $\Omega_{j'}$);* **(iii)** *$\Omega(\boldsymbol{\theta}) \geq \Omega(\Pi_\Theta(\boldsymbol{\theta}))$ (projecting the argument onto Θ does not increase Ω). Then, for any $\bar{\boldsymbol{\theta}} \in \Theta$, at each round t of Alg. 1,*

$$L(\boldsymbol{\theta}_t) + \lambda\Omega(\boldsymbol{\theta}_{t+1}) \leq L(\bar{\boldsymbol{\theta}}) + \lambda\Omega(\bar{\boldsymbol{\theta}}) + \epsilon, \tag{12}$$

where $\epsilon = \frac{\eta_t}{2}G^2 + \frac{\|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_t\|^2 - \|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_{t+1}\|^2}{2\eta_t}$.

If L is σ-strongly convex this bound can be strengthened to

$$L(\boldsymbol{\theta}_t) + \lambda\Omega(\boldsymbol{\theta}_{t+1}) \leq L(\bar{\boldsymbol{\theta}}) + \lambda\Omega(\bar{\boldsymbol{\theta}}) + \epsilon', \tag{13}$$

where $\epsilon' = \epsilon - \frac{\sigma}{2}\|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_t\|^2$.

A related, but less tight, bound for $J = 1$ was derived by Duchi and Singer (2009); instead of our term $\frac{\eta}{2}G^2$ in ϵ, theirs is $7\frac{\eta}{2}G^2$.[8] When $\Omega = \|\cdot\|_1$, FOBOS becomes the truncated gradient algorithm of Langford et al. (2009) and our bound matches the one therein derived, closing the gap between Duchi and Singer (2009) and Langford et al. (2009). Finally, note that the conditions **(i)–(iii)** are not restrictive: they hold whenever the proximity operators are shrinkage functions—*e.g.*, if $\Omega_j(\boldsymbol{\theta}) = \|\boldsymbol{\theta}_{G_j}\|_{2,p_j}^{q_j}$, with $p_j, q_j \geq 1$ and $G_j \subseteq \{1, \ldots, M\}$, which also covers the overlapping group case where $\bigcap_{j=1}^{J} G_j \neq \varnothing$.

We next use Lemma 2 to characterize Alg. 1 in terms of its cumulative regret w.r.t. the best fixed hypothesis, *i.e.*,

$$\text{Reg}_T \triangleq \sum_{t=1}^{T} \left(\lambda\Omega(\boldsymbol{\theta}_t) + L(\boldsymbol{\theta}_t; x_t, y_t)\right) - \min_{\boldsymbol{\theta}\in\Theta} \sum_{t=1}^{T} \left(\lambda\Omega(\boldsymbol{\theta}) + L(\boldsymbol{\theta}; x_t, y_t)\right). \tag{14}$$

Proposition 3 (regret bounds) *Assume the conditions of Lemma 2, along with $\Omega \geq 0$ and $\Omega(\mathbf{0}) = 0$. Then:*

1. *Running Alg. 1 with fixed learning rate η yields*

$$\text{Reg}_T \leq \frac{\eta T}{2}G^2 + \frac{\|\boldsymbol{\theta}^*\|^2}{2\eta}, \tag{15}$$

where $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\Theta} \sum_{t=1}^{T} \left(\lambda\Omega(\boldsymbol{\theta}) + L(\boldsymbol{\theta}; x_t, y_t)\right)$. Setting $\eta = \|\boldsymbol{\theta}^*\|/(G\sqrt{T})$ yields a sublinear regret of $\|\boldsymbol{\theta}^*\|G\sqrt{T}$. (Note that this requires knowing in advance $\|\boldsymbol{\theta}^*\|$ and the number of rounds T.)

2. *Assume that Θ is bounded with diameter F (i.e., $\forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$, $\|\boldsymbol{\theta} - \boldsymbol{\theta}'\| \leq F$). Let the learning rate be $\eta_t = \eta_0/\sqrt{t}$, with arbitrary $\eta_0 > 0$. Then,*

$$\text{Reg}_T \leq \left(\frac{F^2}{2\eta_0} + G^2\eta_0\right)\sqrt{T}. \tag{16}$$

[4] A similar algorithm was proposed independently by Kowalski and Torrésani (2009) in a different context.

[5] Recently, a lot of effort has been placed on ways for computing the proximal step for regularizers with overlapping groups (Liu and Ye, 2010a,b; Mairal et al., 2010). Alg. 1 suggests an alternative approach: split the regularizer into several non-overlapping parts and apply sequential proximal steps. Although in general $\text{prox}_{\Omega_J} \circ \ldots \circ \text{prox}_{\Omega_1} \neq \text{prox}_{\Omega_J \circ \ldots \circ \Omega_1}$, Alg. 1 is still applicable, as we will see in §3.3.

[6] Φ is *σ-strongly convex* in $\mathcal{S}$ if $\forall \boldsymbol{\theta} \in \mathcal{S}, \forall \boldsymbol{g} \in \partial\Phi(\boldsymbol{\theta}), \forall \boldsymbol{\theta}' \in \mathcal{H}$, $\Phi(\boldsymbol{\theta}') \geq \Phi(\boldsymbol{\theta}) + \langle \boldsymbol{g}, \boldsymbol{\theta}' - \boldsymbol{\theta} \rangle + \frac{\sigma}{2}\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|^2$.

[7] Φ is *G-Lipschitz* in $\mathcal{S}$ if $\forall \boldsymbol{\theta} \in \mathcal{S}, \forall \boldsymbol{g} \in \partial\Phi(\boldsymbol{\theta}), \|\boldsymbol{g}\| \leq G$.

[8] This can be seen from their Eq. 9 with $A = 0$ and $\eta_t = \eta_{t+\frac{1}{2}}$.

With $\eta_0 = F/(\sqrt{2}G)$, we obtain $\mathrm{Reg}_T \leq FG\sqrt{2T}$.

3. *If L is σ-strongly convex, and $\eta_t = 1/(\sigma t)$, we obtain a logarithmic regret bound:*

$$\mathrm{Reg}_T \leq G^2(1+\log T)/(2\sigma). \tag{17}$$

Proof: See App. C. ■

Similarly to other analyses of online learning algorithms, once an online-to-batch conversion is specified, regret bounds allow us to obtain PAC bounds on optimization and generalization errors. The following proposition can be proved using the techniques of Cesa-Bianchi et al. (2004) and Shalev-Shwartz et al. (2007).

Proposition 4 (optimization and estimation error) *If the assumptions of Prop. 3 hold and $\eta_t = \eta_0/\sqrt{t}$ as in item 2 in Prop. 3, then the version of Alg. 1 that returns the averaged model solves (9) with ϵ-accuracy[9] in $T = O((F^2G^2 + \log(1/\delta))/\epsilon^2)$ iterations with probability at least $1-\delta$. If L is also σ-strongly convex and $\eta_t = 1/(\sigma t)$ as in item 3 of Prop. 3, then, for the version of Alg. 1 that returns $\boldsymbol{\theta}_{T+1}$, we get $T = \tilde{O}(G^2/(\sigma\delta\epsilon))$. The generalization bounds are of the same orders.*

We now pause to examine some concrete cases. The requirement that the loss is G-Lipschitz holds for the hinge and logistic losses, where $G = 2\max_{u\in\mathcal{U}} \|\boldsymbol{\phi}(u)\|$ (see App. D). These losses are not strongly convex, and therefore Alg. 1 has only $O(1/\epsilon^2)$ convergence. If the *regularizer* Ω is σ-strongly convex, a possible workaround to obtain $\tilde{O}(1/\epsilon)$ convergence is to let L "absorb" that strong convexity by redefining $\tilde{L}(\boldsymbol{\theta}; x_t, y_t) = L(\boldsymbol{\theta}; x_t, y_t) + \sigma\|\boldsymbol{\theta}\|^2/2$. Since neither the $\ell_{2,1}$-norm nor its square are strongly convex, we cannot use this trick for the MKL case, but it *does* apply for non-sparse MKL ($\ell^2_{2,q}$-norms are strongly convex for $q > 1$) and for elastic MKL. Still, the $O(1/\epsilon^2)$ rate for MKL is competitive with the best batch algorithms that tackle the dual; *e.g.*, the method of Xu et al. (2009) achieves ϵ primal-dual gap in $O(1/\epsilon^2)$ iterations.[10] Some losses of interest (*e.g.*, the squared loss, or the modified loss $\tilde{L}$ above) are G-Lipschitz in any compact subset of $\mathcal{H}$ but not in $\mathcal{H}$. However, if the optimal solution is known to lie in some compact set Θ, we can run Alg. 1 with the projection step, making the analysis still applicable.

3.4 SPOM: Structured Prediction with Online MKL

The instantiation of Alg. 1 for structured prediction and $\Omega_{\mathrm{MKL}}(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{\theta}\|^2_{2,1}$ yields the SPOM algorithm (Alg. 3). We consider $L = L_{\mathrm{SVM}}$; adapting to any generalized linear model (*e.g.*, $L = L_{\mathrm{CRF}}$) is straightforward. As discussed in the last paragraph of §3.3, the inclusion of a vacuous projection may accelerate convergence. Hence, an optional upper bound γ on $\|\boldsymbol{\theta}\|$ is accepted as input. Suitable values of γ for the SVM and CRF case are given in App. D.

Algorithm 3 SPOM

1: **input:** $\mathcal{D}$, λ, T, radius γ, learning rates $(\eta_t)_{t=1}^T$
2: initialize $\boldsymbol{\theta}^1 \leftarrow \mathbf{0}$
3: **for** $t = 1$ **to** T **do**
4: sample an instance x_t, y_t
5: compute scores for $m = 1, \ldots, M$:
$$f_m(x_t, y'_t) = \langle \boldsymbol{\theta}^t_m, \boldsymbol{\phi}_m(x_t, y'_t)\rangle$$
6: decode: $\hat{y}_t \in \operatorname{argmax}_{y'_t \in \mathcal{Y}(x)} \sum_{m=1}^M f_m(x_t, y'_t) + c(y'_t, y_t)$
7: Gradient step for $m = 1, \ldots, M$:
$$\tilde{\boldsymbol{\theta}}^t_m = \boldsymbol{\theta}^t_m - \eta_t(\boldsymbol{\phi}_m(x_t, \hat{y}_t) - \boldsymbol{\phi}_m(x_t, y_t))$$
8: compute weights for $m = 1, \ldots, M$: $\tilde{b}^t_m = \|\tilde{\boldsymbol{\theta}}^t_m\|$
9: shrink weights $\mathbf{b}^t = \mathrm{prox}_{\eta_t\lambda\|\cdot\|^2_{2,1}}(\tilde{\mathbf{b}}^t)$ with Alg. 2
10: Proximal step for $m = 1, \ldots, M$: $\tilde{\boldsymbol{\theta}}^{t+1}_m = b^t_m/\tilde{b}^t_m \cdot \tilde{\boldsymbol{\theta}}^t_m$
11: Projection step: $\boldsymbol{\theta}^{t+1} = \tilde{\boldsymbol{\theta}}^{t+1} \cdot \min\{1, \gamma/\|\tilde{\boldsymbol{\theta}}^{t+1}\|\}$
12: **end for**
13: compute $\beta_m \propto \|\boldsymbol{\theta}^{T+1}_m\|$ for $m = 1, \ldots, M$
14: return $\boldsymbol{\beta}$, and the last model $\boldsymbol{\theta}^{T+1}$

In line 5, the scores of candidate outputs are computed blockwise; as described in §2.3, a factorization over parts is assumed and the scores are for partial output assignments. Line 6 gathers all these scores and decodes (loss-augmented inference for the SVM case, or marginal inference for the CRF case). Line 10 is where the block structure is taken into account, by applying a proximity operator which corresponds to a blockwise shrinkage/thresholding, with some blocks eventually being set to zero.

Although Alg. 3 is described with explicit features, it can be kernelized, as shown next (one can also use explicit features in some groups, and implicit in others). Observe that the parameters of the mth block after round t can be written as $\boldsymbol{\theta}^{t+1}_m = \sum_{s=1}^t \alpha^{t+1}_{ms}(\boldsymbol{\phi}_m(x_s, y_s) - \boldsymbol{\phi}_m(x_s, \hat{y}_s))$, where

$$\begin{aligned}\alpha^{t+1}_{ms} &= \eta_s \prod_{r=s}^{t} \left((b^r_m/\tilde{b}^r_m)\min\{1, \gamma/\|\tilde{\boldsymbol{\theta}}^{r+1}\|\}\right) \\ &= \begin{cases} \eta_t(b^t_m/\tilde{b}^t_m)\min\{1, \gamma/\|\tilde{\boldsymbol{\theta}}^{t+1}\|\} & \text{if } s = t \\ \alpha^t_{ms}(b^t_m/\tilde{b}^t_m)\min\{1, \gamma/\|\tilde{\boldsymbol{\theta}}^{t+1}\|\} & \text{if } s < t. \end{cases}\end{aligned}$$

Therefore, the inner products in line 5 can be kernelized. The cost of this step is $O(\min\{N, t\})$, instead of the $O(d_m)$ (where d_m is the dimension of the mth block) for the explicit feature case. After the decoding step (line 6), the supporting pair $(x_t, \hat{y}_t)$ is stored. Lines 9, 11 and 13 require the *norm* of each group, which can be manipulated using kernels: indeed, after each gradient step (line 7), we

[9] *I.e.*, it returns a feasible solution whose objective value is less than ϵ apart from the optimum.

[10] On the other hand, batch proximal gradient methods for smooth losses can be accelerated to achieve $O(1/\sqrt{\epsilon})$ convergence in the primal objective (Beck and Teboulle, 2009).

have (denoting $u_t = (x_t, y_t)$ and $\hat{u}_t = (x_t, \hat{y}_t)$)

$$\begin{aligned}\|\tilde{\boldsymbol{\theta}}_m^t\|^2 &= \|\boldsymbol{\theta}_m^t\|^2 - 2\eta_t\langle\boldsymbol{\theta}_m^t, \boldsymbol{\phi}_m(u_t)\rangle + \\ &\quad \eta_t^2\|\boldsymbol{\phi}_m(\hat{u}_t) - \boldsymbol{\phi}_m(u_t)\|^2 \\ &= \|\boldsymbol{\theta}_m^t\|^2 - 2\eta_t f_m(\hat{u}_t) + \\ &\quad \eta_t^2(K_m(u_t, u_t) + K_m(\hat{u}_t, \hat{u}_t) - 2K_m(u_t, \hat{u}_t));\end{aligned}$$

and the proximal and projection steps merely scale these norms. When the algorithm terminates, it returns the kernel coefficients $\boldsymbol{\beta}$ and the sequence (α_{mt}^{T+1}).

In case of sparse explicit features, an implementation trick analogous to the one used by Shalev-Shwartz et al. (2007) (where each $\boldsymbol{\theta}_m$ is represented by its norm and an unnormalized vector) can substantially reduce the amount of computation. In the case of implicit features with a sparse kernel matrix, a sparse storage of this matrix can also significantly speed up the algorithm, eliminating its dependency on N in line 5. We do that in the experiments (§4). Note that all steps involving block-specific computation can be carried out in parallel using multi-core machines, making Alg. 3 capable of handling many kernels (large M).

4 EXPERIMENTS

We evaluate SPOM (Alg. 3) on two structured prediction tasks: a sequence labeling task (handwriting recognition) and a natural language parsing task (dependency parsing).

4.1 Handwriting Recognition

We use the OCR dataset of Taskar et al. (2003), which has a total of 6,877 words and 52,152 characters.[11] Each character (the input) is a 16×8 binary image, with one of 26 labels (`a-z`, the output to predict). As Taskar et al. (2003), we address this sequence labeling problem with a structural SVM; however, we use the SPOM algorithm to *learn* the kernel from the data. We use an indicator basis function to represent the correlation between consecutive outputs.

MKL versus average kernel. Our first experiment (upper part of Table 1; solid lines in Figure 1) compares linear, quadratic, and Gaussian kernels, either used individually, combined via a simple average, or with MKL (via SPOM). The results show that MKL outperforms the others by $\geq 2\%$, and that learning the bigram weight β_0 (§2.3) does not make any difference. Figure 1 shows that the MKL approach is able to achieve an accurate model sooner.

Feature and kernel sparsity. The second experiment aims at showing SPOM's ability to exploit both *feature* and *kernel* sparsity. We learn a combination of a linear kernel (explicit features) with a generalized B_1-spline kernel,

[11] Available at `www.cis.upenn.edu/~taskar/ocr`.

Table 1: Results for handwriting recognition. Averages over 10 runs (same folds as Taskar et al. (2003), training on one and testing on the others). The linear and quadratic kernels are normalized to unit diagonal. In all cases, 20 epochs were used, with η_0 in (16) picked from $\{0.01, 0.1, 1, 10\}$ by selecting the one that most decreases the objective after 5 epochs. In all cases, the regularization coefficient $C = 1/(\lambda N)$ was chosen with 5-fold cross-validation from $\{0.1, 1, 10, 10^2, 10^3, 10^4\}$.

Kernel	Training Runtimes	Test Acc. (per char.)
Linear (L)	6 sec.	$71.8 \pm 3.9\%$
Quadratic (Q)	116 sec.	$85.5 \pm 0.3\%$
Gaussian (G) ($\sigma^2 = 5$)	123 sec.	$84.1 \pm 0.4\%$
Average $(L + Q + G)/3$	118 sec.	$84.3 \pm 0.3\%$
MKL $\beta_1 L + \beta_2 Q + \beta_3 G$	**279 sec.**	$\mathbf{87.5 \pm 0.3\%}$
MKL $\beta_0, \beta_1 L + \beta_2 Q + \beta_3 G$	**282 sec.**	$\mathbf{87.5 \pm 0.4\%}$
B_1-Spline (B_1)	8 sec.	$75.4 \pm 0.9\%$
Average $(L + B_1)/2$	15 sec.	$83.0 \pm 0.3\%$
MKL $\beta_1 L + \beta_2 B_1$	**15 sec.**	$\mathbf{85.2 \pm 0.3\%}$
MKL $\beta_0, \beta_1 L + \beta_2 B_1$	**16 sec.**	$\mathbf{85.2 \pm 0.3\%}$

given by $K(\mathbf{x}, \mathbf{x}') = \max\{0, 1 - \|\mathbf{x} - \mathbf{x}'\|/h\}$, with h chosen so that the kernel matrix has $\sim 95\%$ zeros. The rationale is to combine the strength of a simple feature-based kernel with that of one depending only on a few nearest neighbors. The results (bottom part of Tab. 1) show that the MKL outperforms by $\sim 10\%$ the individual kernels, and by more than 2% the averaged kernel. Perhaps more importantly, the accuracy is not much worse than the best one obtained in the previous experiment, while the runtime is much faster (15 versus 279 seconds). Figure 1 (dashed lines) is striking in showing the ability of producing a reasonable model very fast.

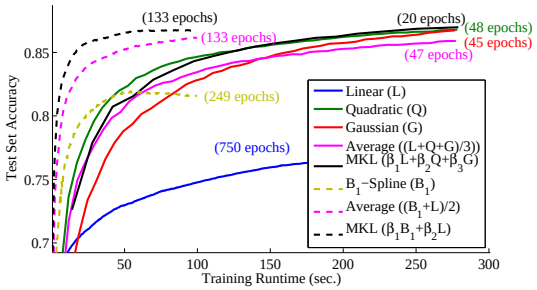


Figure 1: Test set accuracies of single kernel and multiple kernel methods as a function of the training stopping times.

SPOM versus wrapper-based methods. To assess the effectiveness of SPOM as a kernel learning algorithm, we compare it with two wrapper-based MKL algorithms: a Gauss-Seidel method alternating between optimizing the SVM and the kernel coefficients (see, e.g., Kloft et al. 2010), and a gradient method (*SimpleMKL*, Rakotoma-

monjy et al. 2008).[12] In both cases, the SVMs were tackled with structured PEGASOS. Despite the fact that each SVM is strongly convex and has $O(\frac{1}{\epsilon})$ convergence, its combination with a outer loop becomes time-consuming, even if we warm-start each SVM. This is worse when regularization is weak (small λ). In contrast, SPOM, with its overall $O(\frac{1}{\epsilon^2})$ convergence, is stable and very fast to converge to a near-optimal region, as attested in Fig. 2. This suggests its usefulness in settings where each epoch is costly.

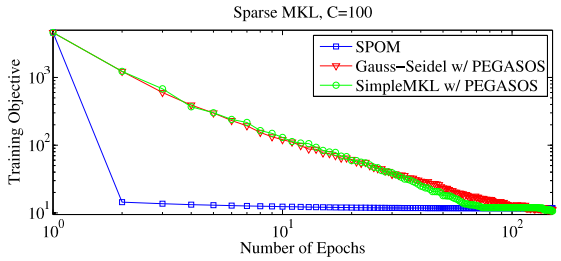


Figure 2: Comparison between SPOM (Alg. 3) and two wrapper based methods in the OCR dataset, with $C = 100$. The wrapper-based methods run 20 epochs of PEGASOS in their first SVM call; subsequent calls run 3 epochs with warm-starting. With only 20–30 passes over the data, SPOM approaches a region very close to the optimum; the wrapper-based methods need about 100 epochs.

4.2 Dependency Parsing

We trained non-projective dependency parsers for English, using the CoNLL-2008 shared task dataset (Surdeanu et al., 2008), in a total of 39,278 training and 2,399 test sentences. The output to be predicted from each input sentence is the set of dependency arcs, linking *heads* to *modifiers*, that must define a spanning tree (see example in Fig. 3). We use arc-factored models, for which exact inference is tractable via minimum spanning tree algorithms (McDonald et al., 2005). We defined $M = 507$ feature templates for each candidate arc by conjoining the words, lemmas, and parts-of-speech of the head and the modifier, as well as the parts-of-speech of the surrounding words, and the distance and direction of attachment. This instantiates > 50 million features. The feature vectors associated with each candidate arc, however, are very sparse and this is exploited in the implementation. We ran SPOM with explicit features, with each group standing for a feature template. MKL (via SPOM) did not outperform a standard SVM in this experiment (90.67% against 90.92%); however, it showed good performance at pruning irrelevant feature templates (see Fig. 3, bottom right). Besides *interpretability*, which may be useful for the understanding of the syntax of natural languages, and memory efficiency (creating a smaller model), this pruning is also appealing in a two-stage architecture,

[12] We used the code in `http://asi.insa-rouen.fr/enseignants/~arakotom/code/mklindex.html`.

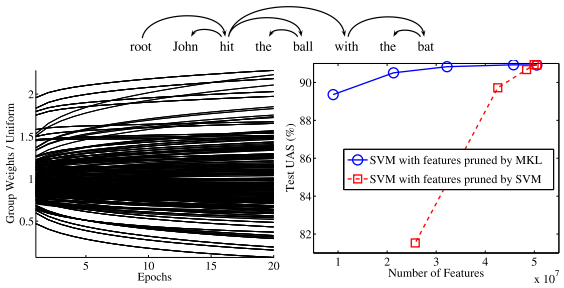


Figure 3: Top: a dependency parse tree (adapted from McDonald et al. 2005). Bottom left: group weights along the epochs of Alg. 3. Bottom right: results of standard SVMs trained on sets of feature templates of sizes $\{107, 207, 307, 407, 507\}$, either selected via a standard SVM or by MKL, via SPOM (the UAS—*unlabeled attachment score*—is the fraction of non-punctuation words whose head was correctly assigned.)

where a learner at a second stage will only need to handle a small fraction of the templates initially hypothesized.

5 RELATED WORK

Discriminative learning of structured predictors has been an active area of research since the seminal works of Lafferty et al. (2001); Collins (2002); Altun et al. (2003); Taskar et al. (2003); Tsochantaridis et al. (2004).

Following the introduction of MKL by Lanckriet et al. (2004), a string of increasingly efficient algorithms were proposed (Sonnenburg et al., 2006; Zien and Ong, 2007; Rakotomamonjy et al., 2008; Chapelle and Rakotomamonjy, 2008; Xu et al., 2009; Suzuki and Tomioka, 2009; Kloft et al., 2010), although none was applied to structured prediction. Group LASSO is due to Bakin (1999); Yuan and Lin (2006), after which many variants and algorithms appeared, all working in batch form: Bach (2008b); Zhao et al. (2008); Jenatton et al. (2009); Friedman et al. (2010).

Independently from us, Jie et al. (2010) recently proposed an online algorithm for multi-class MKL (called OM-2), which differs from ours in that, rather than subgradient and proximal steps, online updates perform coordinate descent in the dual. Our algorithm is more flexible: while OM-2 is limited to $\ell^2_{2,q}$-regularization, with $q > 1$, and becomes slow when $q \to 1$, we efficiently handle the $\ell^2_{2,1}$ case as well as arbitrary composite regularizers. Jie et al. (2010) also have not addressed structured prediction.

Proximity operators are well known in convex analysis and optimization (Moreau, 1962; Lions and Mercier, 1979) and have recently seen wide use in signal processing; see Combettes and Wajs (2006), Wright et al. (2009), and references therein. Specifically, the theory of proximity operators (see App. A) underlies the proofs of our regret bounds (Prop. 3).

6 CONCLUSIONS

We proposed a new method for kernel learning and feature template selection of structured predictors. To accomplish this, we introduced a class of online proximal algorithms applicable to many variants of MKL and group-LASSO. We studied its convergence rate and used the algorithm for learning the kernel in structured prediction tasks.

Our work may impact other problems. In structured prediction, the ability to promote structural sparsity can be useful for learning simultaneously the structure and parameters of graphical models. We will explore this in future work.

Acknowledgements

A. M. was supported by a grant from FCT/ICTI through the CMU-Portugal Program, and also by Priberam Informática. N. S. was supported by NSF IIS-0915187 and NSF CAREER IIS-1054319. E. X. was supported by AFOSR FA9550010247, ONR N000140910758, NSF CAREER DBI-0546594, NSF IIS-0713379, and an Alfred P. Sloan Fellowship. This work was partially supported by the FET programme (EU FP7), under the SIMBAD project (contract 213250), and by a FCT grant PTDC/EEA-TEL/72572/2006.

References

Altun, Y., Tsochantaridis, I., and Hofmann, T. (2003). Hidden Markov support vector machines. In *Proc. of ICML*.

Bach, F. (2008a). Consistency of the group Lasso and multiple kernel learning. *JMLR*, 9:1179–1225.

Bach, F. (2008b). Exploring large feature spaces with hierarchical multiple kernel learning. *NIPS*, 21.

Bach, F., Lanckriet, G., and Jordan, M. (2004). Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML*.

Bakin, S. (1999). *Adaptive regression and model selection in data mining problems*. Australian National University.

Bakır, G., Hofmann, T., Schölkopf, B., Smola, A., Taskar, B., and Vishwanathan, S. (2007). *Predicting Structured Data*. The MIT Press.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202.

Bottou, L. (1991). Stochastic gradient learning in neural networks. In *Proc. of Neuro-Nîmes*.

Bottou, L. and Bousquet, O. (2007). The tradeoffs of large scale learning. *NIPS*, 20.

Cesa-Bianchi, N., Conconi, A., and Gentile, C. (2004). On the generalization ability of on-line learning algorithms. *IEEE Trans. on Inf. Theory*, 50(9):2050–2057.

Chapelle, O. and Rakotomamonjy, A. (2008). Second order optimization of kernel parameters. In *Proc. of the NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*.

Collins, M. (2002). Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In *Proc. of EMNLP*.

Collins, M., Globerson, A., Koo, T., Carreras, X., and Bartlett, P. (2008). Exponentiated gradient algorithms for conditional random fields and max-margin Markov networks. *JMLR*.

Combettes, P. and Wajs, V. (2006). Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200.

Donoho, D. and Johnstone, J. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425.

Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. (2008). Efficient projections onto the L1-ball for learning in high dimensions. In *ICML*.

Duchi, J. and Singer, Y. (2009). Efficient online and batch learning using forward backward splitting. *JMLR*, 10:2873–2908.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). A note on the group lasso and a sparse group lasso.

Hazan, E., Agarwal, A., and Kale, S. (2007). Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192.

Hofmann, T., Scholkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171.

Jenatton, R., Audibert, J.-Y., and Bach, F. (2009). Structured variable selection with sparsity-inducing norms. Technical report, arXiv:0904.3523.

Jie, L., Orabona, F., Fornoni, M., Caputo, B., and Cesa-Bianchi, N. (2010). OM-2: An online multi-class Multi-Kernel Learning algorithm. In *Proc. of the 4th IEEE Online Learning for Computer Vision Workshop*, pages 43–50. IEEE.

Kloft, M., Brefeld, U., Sonnenburg, S., and Zien, A. (2010). Non-Sparse Regularization and Efficient Training with Multiple Kernels. *Arxiv preprint arXiv:1003.0079*.

Kowalski, M. and Torrésani, B. (2009). Structured sparsity: From mixed norms to structured shrinkage. In *Workshop on Signal Processing with Adaptive Sparse Structured Representations*.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E., and Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *JMLR*, 5:27–72.

Langford, J., Li, L., and Zhang, T. (2009). Sparse online learning via truncated gradient. *JMLR*, 10:777–801.

Lions, P. and Mercier, B. (1979). Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979.

Liu, J. and Ye, J. (2010a). Fast Overlapping Group Lasso. *Arxiv preprint arXiv:1009.0306*.

Liu, J. and Ye, J. (2010b). Moreau-yosida regularization for grouped tree structure learning. In Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R., and Culotta, A., editors, *NIPS 23*, pages 1459–1467.

Mairal, J., Jenatton, R., Obozinski, G., and Bach, F. (2010). Network flow algorithms for structured sparsity. In Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R., and Culotta, A., editors, *NIPS 23*, pages 1558–1566.

McDonald, R. T., Pereira, F., Ribarov, K., and Hajic, J. (2005). Non-projective dependency parsing using spanning tree algorithms. In *Proc. of HLT-EMNLP*.

Moreau, J. (1962). Fonctions convexes duales et points proximaux dans un espace hilbertien. *CR Acad. Sci. Paris Sér. A Math*, 255:2897–2899.

Rakotomamonjy, A., Bach, F., Canu, S., and Grandvalet, Y. (2008). SimpleMKL. *JMLR*, 9:2491–2521.

Ratliff, N., Bagnell, J., and Zinkevich, M. (2006). Subgradient methods for maximum margin structured learning. In *ICML Workshop on Learning in Structured Outputs Spaces*.

Shalev-Shwartz, S., Singer, Y., and Srebro, N. (2007). Pegasos: Primal estimated sub-gradient solver for svm. In *ICML*.

Shalev-Shwartz, S., Singer, Y., Srebro, N., and Cotter, A. (2010). Pegasos: primal estimated sub-gradient solver for SVM. *Mathematical Programming*, 125.

Sonnenburg, S., Rätsch, G., Schäfer, C., and Schölkopf, B. (2006). Large scale MKL. *JMLR*, 7:1565.

Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L., and Nivre, J. (2008). The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. *Proc. of CoNLL*.

Suzuki, T. and Tomioka, R. (2009). SpicyMKL. *Arxiv preprint arXiv:0909.5026*.

Taskar, B., Guestrin, C., and Koller, D. (2003). Max-margin Markov networks. In *NIPS*.

Tomioka, R. and Suzuki, T. (2010). Sparsity-accuracy trade-off in MKL. *Arxiv*.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *ICML*.

Vishwanathan, S. V. N., Schraudolph, N. N., Schmidt, M. W., and Murphy, K. P. (2006). Accelerated training of conditional random fields with stochastic gradient methods. In *Proc. of ICML*.

Wright, S., Nowak, R., and Figueiredo, M. (2009). Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493.

Xu, Z., Jin, R., King, I., and Lyu, M. (2009). An extended level method for efficient multiple kernel learning. *NIPS*, 21:1825–1832.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 68(1):49.

Zhao, P., Rocha, G., and Yu, B. (2008). Grouped and hierarchical model selection through composite absolute penalties. *Annals of Statistics*.

Zien, A. and Ong, C. (2007). Multiclass multiple kernel learning. In *ICML*.

Zinkevich, M. (2003). Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *ICML*.

Supplementary Material

A PROXIMITY OPERATORS AND MOREAU PROJECTIONS

Throughout, we let $\varphi : \mathbb{R}^p \to \bar{\mathbb{R}}$ (where $\bar{\mathbb{R}} \triangleq \mathbb{R} \cup \{+\infty\}$) be a convex, lower semicontinuous (lsc) (the epigraph $\text{epi}\varphi \triangleq \{(x,t) \in \mathbb{R}^p \times \mathbb{R} \,|\, \varphi(x) \leq t\}$ is closed in $\mathbb{R}^p \times \mathbb{R}$), and proper ($\exists \mathbf{x} : \varphi(\mathbf{x}) \neq +\infty$) function. The *Fenchel conjugate* of φ is $\varphi^\star : \mathbb{R}^p \to \bar{\mathbb{R}}$, $\varphi^\star(\mathbf{y}) \triangleq \sup_{\mathbf{x}} \mathbf{y}^\top \mathbf{x} - \varphi(\mathbf{x})$. Let:

$$M_\varphi(\mathbf{y}) \triangleq \inf_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2 + \varphi(\mathbf{x}), \quad \text{and} \quad \text{prox}_\varphi(\mathbf{y}) = \arg\inf_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2 + \varphi(\mathbf{x});$$

the function $M_\varphi : \mathbb{R}^p \to \bar{\mathbb{R}}$ is called the *Moreau envelope* of φ, and the map $\text{prox}_\varphi : \mathbb{R}^p \to \mathbb{R}^p$ is the *proximity operator* of φ (Combettes and Wajs, 2006; Moreau, 1962). Proximity operators generalize Euclidean projectors: consider the case $\varphi = \iota_{\mathcal{C}}$, where $\mathcal{C} \subseteq \mathbb{R}^p$ is a convex set and $\iota_{\mathcal{C}}$ denotes its indicator (*i.e.*, $\varphi(\mathbf{x}) = 0$ if $\mathbf{x} \in \mathcal{C}$ and $+\infty$ otherwise). Then, prox_φ is the Euclidean projector onto $\mathcal{C}$ and M_φ is the residual. Two other important examples of proximity operators follow:

- if $\varphi(\mathbf{x}) = (\lambda/2)\|\mathbf{x}\|^2$, then $\text{prox}_\varphi(\mathbf{y}) = \mathbf{y}/(1+\lambda)$;
- if $\varphi(\mathbf{x}) = \tau\|\mathbf{x}\|_1$, then $\text{prox}_\varphi(\mathbf{y}) = \text{soft}(\mathbf{y}, \tau)$ is the *soft-threshold* function (Wright et al., 2009), defined as $[\text{soft}(\mathbf{y},\tau)]_k = \text{sgn}(y_k) \cdot \max\{0, |y_k| - \tau\}$.

If $\varphi : \mathbb{R}^{d_1} \times \ldots \times \mathbb{R}^{d_p} \to \bar{\mathbb{R}}$ is (group-)separable, *i.e.*, $\varphi(\mathbf{x}) = \sum_{k=1}^p \varphi_k(\mathbf{x}_k)$, where $\mathbf{x}_k \in \mathbb{R}^{d_k}$, then its proximity operator inherits the same (group-)separability: $[\text{prox}_\varphi(\mathbf{x})]_k = \text{prox}_{\varphi_k}(\mathbf{x}_k)$ (Wright et al., 2009). For example, the proximity operator of the mixed $\ell_{2,1}$-norm, which is group-separable, has this form. The following proposition extends this result by showing how to compute proximity operators of functions (maybe not separable) that only depend on the ℓ_2-norms of groups of components; *e.g.*, the proximity operator of the squared $\ell_{2,1}$-norm reduces to that of squared ℓ_1.

Proposition 5 *Let $\varphi : \mathbb{R}^{d_1} \times \ldots \times \mathbb{R}^{d_p} \to \bar{\mathbb{R}}$ be of the form $\varphi(\mathbf{x}_1, \ldots, \mathbf{x}_p) = \psi(\|\mathbf{x}_1\|, \ldots, \|\mathbf{x}_p\|)$ for some $\psi : \mathbb{R}^p \to \bar{\mathbb{R}}$. Then, $M_\varphi(\mathbf{x}_1, \ldots, \mathbf{x}_p) = M_\psi(\|\mathbf{x}_1\|, \ldots, \|\mathbf{x}_p\|)$ and $[\text{prox}_\varphi(\mathbf{x}_1, \ldots, \mathbf{x}_p)]_k = [\text{prox}_\psi(\|\mathbf{x}_1\|, \ldots, \|\mathbf{x}_p\|)]_k(\mathbf{x}_k/\|\mathbf{x}_k\|)$.*

Proof: We have respectively:

$$\begin{aligned}
M_\varphi(\mathbf{x}_1, \ldots, \mathbf{x}_p) &= \min_{\mathbf{y}} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|^2 + \varphi(\mathbf{y}) \\
&= \min_{\mathbf{y}_1,\ldots,\mathbf{y}_p} \frac{1}{2}\sum_{k=1}^p \|\mathbf{y}_k - \mathbf{x}_k\|^2 + \psi(\|\mathbf{y}_1\|, \ldots, \|\mathbf{y}_p\|) \\
&= \min_{\mathbf{u}\in\mathbb{R}^p_+} \psi(u_1, \ldots, u_p) + \min_{\mathbf{y}:\|\mathbf{y}_k\|=u_k, \forall k} \frac{1}{2}\sum_{k=1}^p \|\mathbf{y}_k - \mathbf{x}_k\|^2 \\
&= \min_{\mathbf{u}\in\mathbb{R}^p_+} \psi(u_1, \ldots, u_p) + \frac{1}{2}\sum_{k=1}^p \min_{\mathbf{y}_k:\|\mathbf{y}_k\|=u_k} \|\mathbf{y}_k - \mathbf{x}_k\|^2 \quad (*) \\
&= \min_{\mathbf{u}\in\mathbb{R}^p_+} \psi(u_1, \ldots, u_p) + \frac{1}{2}\sum_{k=1}^p \left\| \frac{u_k}{\|\mathbf{x}_k\|}\mathbf{x}_k - \mathbf{x}_k \right\|^2 \\
&= \min_{\mathbf{u}\in\mathbb{R}^p_+} \psi(u_1, \ldots, u_p) + \frac{1}{2}\sum_{k=1}^p (u_k - \|\mathbf{x}_k\|)^2 \\
&= M_\psi(\|\mathbf{x}_1\|, \ldots, \|\mathbf{x}_p\|),
\end{aligned} \tag{18}$$

where the solution of the innermost minimization problem in $(*)$ is $\mathbf{y}_k = \frac{u_k}{\|\mathbf{x}_k\|}\mathbf{x}_k$, and therefore $[\text{prox}_\varphi(\mathbf{x}_1, \ldots, \mathbf{x}_p)]_k = [\text{prox}_\psi(\|\mathbf{x}_1\|, \ldots, \|\mathbf{x}_p\|)]_k \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|}$. ■

Finally, we recall the *Moreau decomposition*, relating the proximity operators of Fenchel conjugate functions (Combettes and Wajs, 2006) and present a corollary that is the key to our regret bound in §3.3.

Proposition 6 (Moreau (1962)) *For any convex, lsc, proper function* $\varphi : \mathbb{R}^p \to \bar{\mathbb{R}}$,

$$\mathbf{x} = \mathrm{prox}_{\varphi}(\mathbf{x}) + \mathrm{prox}_{\varphi^\star}(\mathbf{x}) \quad \textit{and} \quad \|\mathbf{x}\|^2/2 = M_{\varphi}(\mathbf{x}) + M_{\varphi^\star}(\mathbf{x}). \tag{19}$$

Corollary 7 *Let* $\varphi : \mathbb{R}^p \to \bar{\mathbb{R}}$ *be as in Prop. 6, and* $\bar{\mathbf{x}} \triangleq \mathrm{prox}_{\varphi}(\mathbf{x})$. *Then, any* $\mathbf{y} \in \mathbb{R}^p$ *satisfies*

$$\|\mathbf{y} - \bar{\mathbf{x}}\|^2 - \|\mathbf{y} - \mathbf{x}\|^2 \leq 2(\varphi(\mathbf{y}) - \varphi(\bar{\mathbf{x}})). \tag{20}$$

Proof: We start by stating and proving the following lemma:

Lemma 8 *Let* $\varphi : \mathbb{R}^p \to \bar{\mathbb{R}}$ *be as in Prop. 6, and let* $\bar{\mathbf{x}} \triangleq \mathrm{prox}_{\varphi}(\mathbf{x})$. *Then, any* $\mathbf{y} \in \mathbb{R}^p$ *satisfies*

$$(\bar{\mathbf{x}} - \mathbf{y})^\top (\bar{\mathbf{x}} - \mathbf{x}) \leq \varphi(\mathbf{y}) - \varphi(\bar{\mathbf{x}}) \tag{21}$$

Proof (of the Lemma): From (19), we have that

$$\begin{aligned}
\frac{1}{2}\|\mathbf{x}\|^2 &= \frac{1}{2}\|\bar{\mathbf{x}} - \mathbf{x}\|^2 + \varphi(\bar{\mathbf{x}}) + \frac{1}{2}\|\bar{\mathbf{x}}\|^2 + \varphi^*(\mathbf{x} - \bar{\mathbf{x}}) \\
&= \frac{1}{2}\|\bar{\mathbf{x}} - \mathbf{x}\|^2 + \varphi(\bar{\mathbf{x}}) + \frac{1}{2}\|\bar{\mathbf{x}}\|^2 + \sup_{\mathbf{u}\in\mathbb{R}^p} \left(\mathbf{u}^\top(\mathbf{x} - \bar{\mathbf{x}}) - \varphi(\mathbf{u})\right) \\
&\geq \frac{1}{2}\|\bar{\mathbf{x}} - \mathbf{x}\|^2 + \varphi(\bar{\mathbf{x}}) + \frac{1}{2}\|\bar{\mathbf{x}}\|^2 + \mathbf{y}^\top(\mathbf{x} - \bar{\mathbf{x}}) - \varphi(\mathbf{y}) \\
&= \frac{1}{2}\|\mathbf{x}\|^2 + \bar{\mathbf{x}}^\top(\bar{\mathbf{x}} - \mathbf{x}) + \mathbf{y}^\top(\mathbf{x} - \bar{\mathbf{x}}) - \varphi(\mathbf{y}) + \varphi(\bar{\mathbf{x}}) \\
&= \frac{1}{2}\|\mathbf{x}\|^2 + (\bar{\mathbf{x}} - \mathbf{y})^\top(\bar{\mathbf{x}} - \mathbf{x}) - \varphi(\mathbf{y}) + \varphi(\bar{\mathbf{x}}),
\end{aligned}$$

from which (21) follows. ∎

Now, take Lemma 8 and bound the left hand side as:

$$\begin{aligned}
(\bar{\mathbf{x}} - \mathbf{y})^\top(\bar{\mathbf{x}} - \mathbf{x}) &\geq (\bar{\mathbf{x}} - \mathbf{y})^\top(\bar{\mathbf{x}} - \mathbf{x}) - \frac{1}{2}\|\bar{\mathbf{x}} - \mathbf{x}\|^2 \\
&= (\bar{\mathbf{x}} - \mathbf{y})^\top(\bar{\mathbf{x}} - \mathbf{x}) - \frac{1}{2}\|\bar{\mathbf{x}}\|^2 - \frac{1}{2}\|\mathbf{x}\|^2 + \bar{\mathbf{x}}^\top\mathbf{x} \\
&= \frac{1}{2}\|\bar{\mathbf{x}}\|^2 - \mathbf{y}^\top(\bar{\mathbf{x}} - \mathbf{x}) - \frac{1}{2}\|\mathbf{x}\|^2 \\
&= \frac{1}{2}\|\mathbf{y} - \bar{\mathbf{x}}\|^2 - \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|^2.
\end{aligned}$$

This concludes the proof of Corollary 7. ∎

Note that although the Fenchel dual $\varphi^\star$ does not show up in (20), it has a crucial role in this proof.

B PROOF OF LEMMA 2

Let $u(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}) \triangleq \lambda\Omega(\bar{\boldsymbol{\theta}}) - \lambda\Omega(\boldsymbol{\theta})$. We have successively:

$$\begin{aligned}
\|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_{t+1}\|^2 &\leq^{(\mathrm{i})} \|\bar{\boldsymbol{\theta}} - \tilde{\boldsymbol{\theta}}_{t+1}\|^2 \\
&\leq^{(\mathrm{ii})} \|\bar{\boldsymbol{\theta}} - \tilde{\boldsymbol{\theta}}_{t}\|^2 + 2\eta_t\lambda\textstyle\sum_{j=1}^{J}(\Omega_j(\bar{\boldsymbol{\theta}}) - \Omega_j(\tilde{\boldsymbol{\theta}}_{t+j/J})) \\
&\leq^{(\mathrm{iii})} \|\bar{\boldsymbol{\theta}} - \tilde{\boldsymbol{\theta}}_{t}\|^2 + 2\eta_t u(\bar{\boldsymbol{\theta}}, \tilde{\boldsymbol{\theta}}_{t+1}) \\
&\leq^{(\mathrm{iv})} \|\bar{\boldsymbol{\theta}} - \tilde{\boldsymbol{\theta}}_{t}\|^2 + 2\eta_t u(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}_{t+1}) \\
&= \|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_{t}\|^2 + \|\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}_{t}\|^2 + 2(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_t)^\top(\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}_t) + 2\eta_t u(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}_{t+1}) \\
&= \|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_{t}\|^2 + \eta_t^2\|\boldsymbol{g}\|^2 + 2\eta_t(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_t)^\top\boldsymbol{g} + 2\eta_t u(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}_{t+1}) \\
&\leq^{(\mathrm{v})} \|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_{t}\|^2 + \eta_t^2\|\boldsymbol{g}\|^2 + 2\eta_t(L(\bar{\boldsymbol{\theta}}) - L(\boldsymbol{\theta}_t)) + 2\eta_t u(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}_{t+1}) \\
&\leq \|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_{t}\|^2 + \eta_t^2 G^2 + 2\eta_t(L(\bar{\boldsymbol{\theta}}) - L(\boldsymbol{\theta}_t)) + 2\eta_t u(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}_{t+1}),
\end{aligned} \tag{22}$$

where the inequality (i) is due to the nonexpansiveness of the projection operator, (ii) follows from applying Corollary 7 J times, (iii) follows from applying the inequality $\Omega_j(\tilde{\boldsymbol{\theta}}_{t+l/J}) \geq \Omega_j(\tilde{\boldsymbol{\theta}}_{t+(l+1)/J})$ for $l = j, \ldots, J-1$, (iv) results from the fact that $\Omega(\tilde{\boldsymbol{\theta}}_{t+1}) \geq \Omega(\Pi_\Theta(\tilde{\boldsymbol{\theta}}_{t+1}))$, and (v) results from the subgradient inequality of convex functions, which has an extra term $\frac{\sigma}{2}\|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_t\|^2$ if L is σ-strongly convex.

C PROOF OF PROPOSITION 3

Invoke Lemma 2 and sum for $t = 1, \ldots, T$, which gives

$$\begin{aligned}
\sum_{t=1}^{T} (L(\boldsymbol{\theta}_t; x_t, y_t) + \lambda\Omega(\boldsymbol{\theta}_t)) &= \sum_{t=1}^{T} (L(\boldsymbol{\theta}_t; x_t, y_t) + \lambda\Omega(\boldsymbol{\theta}_{t+1})) - \lambda(\Omega(\boldsymbol{\theta}_{T+1}) - \Omega(\boldsymbol{\theta}_1)) \\
&\leq^{\text{(i)}} \sum_{t=1}^{T} (L(\boldsymbol{\theta}_t; x_t, y_t) + \lambda\Omega(\boldsymbol{\theta}_{t+1})) \\
&\leq \sum_{t=1}^{T} (L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda\Omega(\boldsymbol{\theta}^*)) + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t + \sum_{t=1}^{T}\frac{\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_t\|^2 - \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_{t+1}\|^2}{2\eta_t} \\
&= \sum_{t=1}^{T} (L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda\Omega(\boldsymbol{\theta}^*)) + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t + \frac{1}{2}\sum_{t=2}^{T}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right)\cdot\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_t\|^2 \\
&\quad + \frac{1}{2\eta_1}\cdot\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_1\|^2 - \frac{1}{2\eta_T}\cdot\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_{T+1}\|^2 \qquad (23)
\end{aligned}$$

where the inequality (i) is due to the fact that $\boldsymbol{\theta}_1 = \mathbf{0}$. Noting that the third term vanishes for a constant learning rate and that the last term is non-positive suffices to prove the first part. For the second part, we continue as:

$$\begin{aligned}
\sum_{t=1}^{T} (L(\boldsymbol{\theta}_t; x_t, y_t) + \lambda\Omega(\boldsymbol{\theta}_t)) &\leq \sum_{t=1}^{T} (L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda\Omega(\boldsymbol{\theta}^*)) + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t + \frac{F^2}{2}\sum_{t=2}^{T}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) + \frac{F^2}{2\eta_1} \\
&= \sum_{t=1}^{T} (L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda\Omega(\boldsymbol{\theta}^*)) + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t + \frac{F^2}{2\eta_T} \\
&\leq^{\text{(ii)}} \sum_{t=1}^{T} (L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda\Omega(\boldsymbol{\theta}^*)) + G^2\eta_0(\sqrt{T} - 1/2) + \frac{F^2\sqrt{T}}{2\eta_0} \\
&\leq \sum_{t=1}^{T} (L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda\Omega(\boldsymbol{\theta}^*)) + \left(G^2\eta_0 + \frac{F^2}{2\eta_0}\right)\sqrt{T}, \qquad (24)
\end{aligned}$$

where equality (ii) is due to the fact that $\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq 2\sqrt{T} - 1$. For the third part, continue after inequality (i) as:

$$\begin{aligned}
\sum_{t=1}^{T} (L(\boldsymbol{\theta}_t; x_t, y_t) + \lambda\Omega(\boldsymbol{\theta}_t)) &\leq \sum_{t=1}^{T} (L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda\Omega(\boldsymbol{\theta}^*)) + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t + \frac{1}{2}\sum_{t=2}^{T}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \sigma\right)\cdot\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_t\|^2 \\
&\quad + \frac{1}{2}\left(\frac{1}{\eta_1} - \sigma\right)\cdot\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_1\|^2 - \frac{1}{2\eta_T}\cdot\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_{T+1}\|^2 \\
&= \sum_{t=1}^{T} (L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda\Omega(\boldsymbol{\theta}^*)) + \frac{G^2}{2\sigma}\sum_{t=1}^{T}\frac{1}{t} - \frac{\sigma T}{2}\cdot\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_{T+1}\|^2 \\
&\leq \sum_{t=1}^{T} (L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda\Omega(\boldsymbol{\theta}^*)) + \frac{G^2}{2\sigma}\sum_{t=1}^{T}\frac{1}{t} \\
&\leq^{\text{(iii)}} \sum_{t=1}^{T} (L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda\Omega(\boldsymbol{\theta}^*)) + \frac{G^2}{2\sigma}(1 + \log T), \qquad (25)
\end{aligned}$$

where the equality (iii) is due to the fact that $\sum_{t=1}^{T}\frac{1}{t} \leq 1 + \log T$.

D LIPSCHITZ CONSTANTS FOR SOME LOSS FUNCTIONS

Let $\boldsymbol{\theta}^*$ be a solution of the problem (9) with $\Theta = \mathcal{H}$. For certain loss functions, we may obtain bounds of the form $\|\boldsymbol{\theta}^*\| \leq \gamma$ for some $\gamma > 0$, as the next proposition illustrates. Therefore, we may redefine $\Theta = \{\boldsymbol{\theta} \in \mathcal{H} \mid \|\boldsymbol{\theta}\| \leq \gamma\}$ (a vacuous constraint) without affecting the solution of (9).

Proposition 9 *Let $\Omega(\boldsymbol{\theta}) = \frac{1}{2}(\sum_{m=1}^{M} \|\boldsymbol{\theta}_m\|)^2$. Let L_{SVM} and L_{CRF} be the structured hinge and logistic losses (4). Assume that the average cost function (in the SVM case) or the average entropy (in the CRF case) are bounded by some $\Lambda \geq 0$, i.e.,*[13]

$$\frac{1}{N}\sum_{i=1}^{N} \max_{y_i' \in \in \mathcal{Y}(x_t)} c(y_i'; y_i) \leq \Lambda \quad \textit{or} \quad \frac{1}{N}\sum_{i=1}^{N} H(Y_i) \leq \Lambda. \tag{26}$$

Then:

1. *The solution of (9) with $\Theta = \mathcal{H}$ satisfies $\|\boldsymbol{\theta}^*\| \leq \sqrt{2\Lambda/\lambda}$.*
2. *L is G-Lipschitz on $\mathcal{H}$, with $G = 2\max_{u\in\mathcal{U}} \|\boldsymbol{\phi}(u)\|$.*
3. *Consider the following problem obtained from (9) by adding a quadratic term:*

$$\min_{\boldsymbol{\theta}} \frac{\sigma}{2}\|\boldsymbol{\theta}\|^2 + \lambda\Omega(\boldsymbol{\theta}) + \frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{\theta}; x_i, y_i). \tag{27}$$

   The solution of this problem satisfies $\|\boldsymbol{\theta}^*\| \leq \sqrt{2\Lambda/(\lambda+\sigma)}$.
4. *The modified loss $\tilde{L} = L + \frac{\sigma}{2}\|.\|^2$ is $\tilde{G}$-Lipschitz on $\left\{\boldsymbol{\theta} \mid \|\boldsymbol{\theta}\| \leq \sqrt{2\Lambda/(\lambda+\sigma)}\right\}$, where $\tilde{G} = G + \sqrt{2\sigma^2\Lambda/(\lambda+\sigma)}$.*

Proof: Let $F_{\mathrm{SVM}}(\boldsymbol{\theta})$ and $F_{\mathrm{CRF}}(\boldsymbol{\theta})$ be the objectives of (9) for the SVM and CRF cases. We have

$$F_{\mathrm{SVM}}(\mathbf{0}) = \lambda\Omega(\mathbf{0}) + \frac{1}{N}\sum_{i=1}^{N} L_{\mathrm{SVM}}(\mathbf{0}; x_i, y_i) = \frac{1}{N}\sum_{i=1}^{N} \max_{y_i' \in \mathcal{Y}(x_i)} c(y_i'; y_i) \leq \Lambda_{\mathrm{SVM}} \tag{28}$$

$$F_{\mathrm{CRF}}(\mathbf{0}) = \lambda\Omega(\mathbf{0}) + \frac{1}{N}\sum_{i=1}^{N} L_{\mathrm{CRF}}(\mathbf{0}; x_i, y_i) = \frac{1}{N}\sum_{i=1}^{N} \log|\mathcal{Y}(x_i)| \leq \Lambda_{\mathrm{CRF}} \tag{29}$$

Using the facts that $F(\boldsymbol{\theta}^*) \leq F(\mathbf{0})$, that the losses are non-negative, and that $(\sum_i |x_i|)^2 \geq \sum_i x_i^2$, we obtain $\frac{\lambda}{2}\|\boldsymbol{\theta}^*\|^2 \leq \lambda\Omega(\boldsymbol{\theta}^*) \leq F(\boldsymbol{\theta}^*) \leq F(\mathbf{0})$, which proves the first statement.

To prove the second statement for the SVM case, note that a subgradient of L_{SVM} at $\boldsymbol{\theta}$ is $\boldsymbol{g}_{\mathrm{SVM}} = \boldsymbol{\phi}(x,\hat{y}) - \boldsymbol{\phi}(x,y)$, where $\hat{y} = \arg\max_{y'\in\mathcal{Y}(x)} \boldsymbol{\theta}^\top(\boldsymbol{\phi}(x,y') - \boldsymbol{\phi}(x,y)) + c(y';y)$; and that the gradient of L_{CRF} at $\boldsymbol{\theta}$ is $\boldsymbol{g}_{\mathrm{CRF}} = \mathbb{E}_{\boldsymbol{\theta}}\boldsymbol{\phi}(x,Y) - \boldsymbol{\phi}(x,y)$. Applying Jensen's inequality, we have that $\|\boldsymbol{g}_{\mathrm{CRF}}\| \leq \mathbb{E}_{\boldsymbol{\theta}}\|\boldsymbol{\phi}(x,Y) - \boldsymbol{\phi}(x,y)\|$. Therefore, both $\|\boldsymbol{g}_{\mathrm{SVM}}\|$ and $\|\boldsymbol{g}_{\mathrm{CRF}}\|$ are upper bounded by $\max_{x\in\mathcal{X},y,y'\in\mathcal{Y}(x)} \|\boldsymbol{\phi}(x,y') - \boldsymbol{\phi}(x,y)\| \leq 2\max_{u\in\mathcal{U}}\|\boldsymbol{\phi}(u)\|$.

The same rationale can be used to prove the third and fourth statements. ■

E COMPUTING THE PROXIMITY OPERATOR OF THE (NON-SEPARABLE) SQUARED ℓ_1

We present an algorithm (Alg. 4) that computes the Moreau projection of the *squared*, *weighted* ℓ_1-norm. Denote by $\odot$ the Hadamard product, $[\mathbf{a}\odot\mathbf{b}]_k = a_k b_k$. Letting $\lambda, \mathbf{d} \geq 0$, and $\phi_{\mathbf{d}}(\mathbf{x}) \triangleq \frac{1}{2}\|\mathbf{d}\odot\mathbf{x}\|_1^2$, the underlying optimization problem is:

$$M_{\lambda\phi_{\mathbf{d}}}(\mathbf{x}_0) \triangleq \min_{\mathbf{x}\in\mathbb{R}^M} \frac{1}{2}\|\mathbf{x}-\mathbf{x}_0\|^2 + \frac{\lambda}{2}\left(\sum_{m=1}^{M} d_m|x_m|\right)^2. \tag{30}$$

[13] In sequence binary labeling, we have $\Lambda = \bar{P}$ for the CRF case and for the SVM case with a Hamming cost function, where $\bar{P}$ is the average sequence length. Observe that the entropy of a distribution over labelings of a sequence of length P is upper bounded by $\log 2^P = P$.

Algorithm 4 Moreau projection for the squared weighted ℓ_1-norm

Input: A vector $\mathbf{x}_0 \in \mathbb{R}^M$, a weight vector $\mathbf{d} \geq 0$, and a parameter $\lambda > 0$
Set $u_{0m} = |x_{0m}|/d_m$ and $a_m = d_m^2$ for each $m = 1, \ldots, M$
Sort $\mathbf{u}_0$: $u_{0(1)} \geq \ldots \geq u_{0(M)}$
Find $\rho = \max \left\{ j \in \{1, \ldots, M\} \mid u_{0(j)} - \frac{\lambda}{1+\lambda \sum_{r=1}^{j} a_{(r)}} \sum_{r=1}^{j} a_{(r)} u_{0(r)} > 0 \right\}$
Compute $\mathbf{u} = \text{soft}(\mathbf{u}_0, \tau)$, where $\tau = \frac{\lambda}{1+\lambda \sum_{r=1}^{\rho} a_{(r)}} \sum_{r=1}^{\rho} a_{(r)} u_{0(r)}$
Output: $\mathbf{x}$ s.t. $x_r = \text{sign}(x_{0r}) d_r u_r$.

This includes the squared ℓ_1-norm as a particular case, when $\mathbf{d} = \mathbf{1}$ (the case addressed in Alg. 2). The proof is somewhat technical and follows the same procedure employed by Duchi et al. (2008) to derive an algorithm for projecting onto the ℓ_1-ball. The runtime is $O(M \log M)$ (the amount of time that is necessary to sort the vector), but a similar trick as the one described by (Duchi et al., 2008) can be employed to yield $O(M)$ runtime.

Lemma 10 *Let* $\mathbf{x}^* = \text{prox}_{\lambda \phi_{\mathbf{d}}}(\mathbf{x}_0)$ *be the solution of* (30)*. Then:*

1. $\mathbf{x}^*$ *agrees in sign with* $\mathbf{x}_0$, i.e.*, each component satisfies* $x_{0i} \cdot x_i^* \geq 0$.
2. *Let* $\boldsymbol{\sigma} \in \{-1, 1\}^M$. *Then* $\text{prox}_{\lambda \phi_{\mathbf{d}}}(\boldsymbol{\sigma} \odot \mathbf{x}_0) = \boldsymbol{\sigma} \odot \text{prox}_{\lambda \phi_{\mathbf{d}}}(\mathbf{x}_0)$, i.e.*, flipping a sign in* $\mathbf{x}_0$ *produces a* $\mathbf{x}^*$ *with the same sign flipped.*

Proof: Suppose that $x_{0i} \cdot x_i^* < 0$ for some i. Then, $\mathbf{x}$ defined by $x_j = x_j^*$ for $j \neq i$ and $x_i = -x_i^*$ achieves a lower objective value than $\mathbf{x}^*$, since $\phi_{\mathbf{d}}(\mathbf{x}) = \phi_{\mathbf{d}}(\mathbf{x}^*)$ and $(x_i - x_{0i})^2 < (x_i^* - x_{0i})^2$; this contradicts the optimality of $\mathbf{x}^*$. The second statement is a simple consequence of the first one and that $\phi_{\mathbf{d},\lambda}(\boldsymbol{\sigma} \odot \mathbf{x}) = \phi_{\mathbf{d},\lambda}(\boldsymbol{\sigma} \odot \mathbf{x}^*)$. ■

Lemma 10 enables reducing the problem to the non-negative orthant, by writing $\mathbf{x}_0 = \boldsymbol{\sigma} \cdot \tilde{\mathbf{x}}_0$, with $\tilde{\mathbf{x}}_0 \geq \mathbf{0}$, obtaining a solution $\tilde{\mathbf{x}}^*$ and then recovering the true solution as $\mathbf{x}^* = \boldsymbol{\sigma} \cdot \tilde{\mathbf{x}}^*$. It therefore suffices to solve (30) with the constraint $\mathbf{x} \geq \mathbf{0}$, which in turn can be transformed into:

$$\min_{\mathbf{u} \geq \mathbf{0}} F(\mathbf{u}) \triangleq \frac{1}{2} \sum_{m=1}^{M} a_m (u_m - u_{0m})^2 + \frac{\lambda}{2} \left(\sum_{m=1}^{M} a_m u_m \right)^2, \tag{31}$$

where we made the change of variables $a_m \triangleq d_m^2$, $u_{0m} \triangleq x_{0m}/d_m$ and $u_m \triangleq x_m/d_m$.

The Lagrangian of (31) is $\mathcal{L}(\mathbf{u}, \boldsymbol{\xi}) = \frac{1}{2} \sum_{m=1}^{M} a_m (u_m - u_{0m})^2 + \frac{\lambda}{2} \left(\sum_{m=1}^{M} a_m u_m \right)^2 - \boldsymbol{\xi}^\top \mathbf{u}$, where $\boldsymbol{\xi} \geq \mathbf{0}$ are Lagrange multipliers. Equating the gradient (w.r.t. $\mathbf{u}$) to zero gives

$$\mathbf{a} \odot (\mathbf{u} - \mathbf{u}_0) + \lambda \sum_{m=1}^{M} a_m u_m \mathbf{a} - \boldsymbol{\xi} = \mathbf{0}. \tag{32}$$

From the complementary slackness condition, $u_j > 0$ implies $\xi_j = 0$, which in turn implies

$$a_j (u_j - u_{0j}) + \lambda a_j \sum_{m=1}^{M} a_m u_m = 0. \tag{33}$$

Thus, if $u_j > 0$, the solution is of the form $u_j = u_{0j} - \tau$, with $\tau = \lambda \sum_{m=1}^{M} a_m u_m$. The next lemma shows the existence of a split point below which some coordinates vanish.

Lemma 11 *Let* $\mathbf{u}^*$ *be the solution of* (31)*. If* $u_k^* = 0$ *and* $u_{0j} < u_{0k}$*, then we must have* $u_j^* = 0$.

Proof: Suppose that $u_j^* = \epsilon > 0$. We will construct a $\tilde{\mathbf{u}}$ whose objective value is lower than $F(\mathbf{u}^*)$, which contradicts the optimality of $\mathbf{u}^*$: set $\tilde{u}_l = u_l^*$ for $l \notin \{j, k\}$, $\tilde{u}_k = \epsilon c$, and $\tilde{u}_j = \epsilon\,(1 - c a_k/a_j)$, where $c = \min\{a_j/a_k, 1\}$. We have $\sum_{m=1}^{M} a_m u_m^* = \sum_{m=1}^{M} a_m \tilde{u}_m$, and therefore

$$\begin{aligned} 2(F(\tilde{\mathbf{u}}) - F(\mathbf{u}^*)) &= \sum_{m=1}^{M} a_m (\tilde{u}_m - u_{0m})^2 - \sum_{m=1}^{M} a_m (u_m^* - u_{0m})^2 \\ &= a_j (\tilde{u}_j - u_{0j})^2 - a_j (u_j^* - u_{0j})^2 + a_k (\tilde{u}_k - u_{0k})^2 - a_k (u_k^* - u_{0k})^2. \end{aligned} \tag{34}$$

Consider the following two cases: (i) if $a_j \leq a_k$, then $\tilde{u}_k = \epsilon a_j/a_k$ and $\tilde{u}_j = 0$. Substituting in (34), we obtain $2(F(\tilde{\mathbf{u}}) - F(\mathbf{u}^*)) = \epsilon^2 \left(a_j^2/a_k - a_j\right) \leq 0$, which leads to the contradiction $F(\tilde{\mathbf{u}}) \leq F(\mathbf{u}^*)$. If (ii) $a_j > a_k$, then $\tilde{u}_k = \epsilon$ and $\tilde{u}_j = \epsilon\left(1 - a_k/a_j\right)$. Substituting in (34), we obtain $2(F(\tilde{\mathbf{u}}) - F(\mathbf{u}^*)) = a_j\epsilon^2\left(1 - a_k/a_j\right)^2 + 2a_k\epsilon u_{0j} - 2a_k\epsilon u_{0k} + a_k\epsilon^2 - a_j\epsilon^2 < a_k^2/a_j\epsilon^2 - 2a_k\epsilon^2 + a_k\epsilon^2 = \epsilon^2\left(a_k^2/a_j - a_k\right) < 0$, which also leads to a contradiction. ■

Let $u_{0(1)} \geq \ldots \geq u_{0(M)}$ be the entries of $\mathbf{u}_0$ sorted in decreasing order, and let $u^*_{(1)}, \ldots, u^*_{(M)}$ be the entries of $\mathbf{u}^*$ under the same permutation. Let ρ be the number of nonzero entries in $\mathbf{u}^*$, i.e., $u^*_{(\rho)} > 0$, and, if $\rho < M$, $u^*_{(\rho+1)} = 0$. Summing (33) for $(j) = 1, \ldots, \rho$, we get

$$\sum_{r=1}^{\rho} a_{(r)} u^*_{(r)} - \sum_{r=1}^{\rho} a_{(r)} u_{0(r)} + \left(\sum_{r=1}^{\rho} a_{(r)}\right) \lambda \sum_{r=1}^{\rho} a_{(r)} u^*_{(r)} = 0, \tag{35}$$

which implies

$$\sum_{m=1}^{M} u^*_m = \sum_{r=1}^{\rho} u^*_{(r)} = \frac{1}{1 + \lambda \sum_{r=1}^{\rho} a_{(r)}} \sum_{r=1}^{\rho} a_{(r)} u_{0(r)}, \tag{36}$$

and therefore $\tau = \frac{\lambda}{1+\lambda \sum_{r=1}^{\rho} a_{(r)}} \sum_{r=1}^{\rho} a_{(r)} u_{0(r)}$. The complementary slackness conditions for $r = \rho$ and $r = \rho + 1$ imply

$$u^*_{(\rho)} - u_{0(\rho)} + \lambda \sum_{r=1}^{\rho} a_{(r)} u^*_{(r)} = 0 \quad \text{and} \quad -u^*_{0(\rho+1)} + \lambda \sum_{r=1}^{\rho} a_{(r)} u^*_{(r)} = \xi_{(\rho+1)} \geq 0; \tag{37}$$

therefore $u_{0(\rho)} > u_{0(\rho)} - u^*_{(\rho)} = \tau \geq u_{0(\rho+1)}$. This implies that ρ is such that

$$u_{0(\rho)} > \frac{\lambda}{1 + \lambda \sum_{r=1}^{\rho} a_{(r)}} \sum_{r=1}^{\rho} a_{(r)} u_{0(r)} \geq u_{0(\rho+1)}. \tag{38}$$

The next proposition goes farther by exactly determining ρ.

Proposition 12 *The quantity ρ can be determined via:*

$$\rho = \max\left\{ j \in \{1, \ldots, M\} \;\middle|\; u_{0(j)} - \frac{\lambda}{1 + \lambda \sum_{r=1}^{j} a_{(r)}} \sum_{r=1}^{j} a_{(r)} u_{0(r)} > 0 \right\}. \tag{39}$$

Proof: Let $\rho^* = \max\{j | u^*_{(j)} > 0\}$. We have that $u^*_{(r)} = u_{0(r)} - \tau^*$ for $r \leq \rho^*$, where $\tau^* = \frac{\lambda}{1+\lambda \sum_{r=1}^{\rho^*} a_{(r)}} \sum_{r=1}^{\rho^*} a_{(r)} u_{0(r)}$, and therefore $\rho \geq \rho^*$. We need to prove that $\rho \leq \rho^*$, which we will do by contradiction. Assume that $\rho > \rho^*$. Let $\mathbf{u}$ be the vector induced by the choice of ρ, i.e., $u_{(r)} = 0$ for $r > \rho$ and $u_{(r)} = u_{0(r)} - \tau$ for $r \leq \rho$, where $\tau = \frac{\lambda}{1+\lambda \sum_{r=1}^{\rho} a_{(r)}} \sum_{r=1}^{\rho} a_{(r)} u_{0(r)}$. From the definition of ρ, we have $u_{(\rho)} = u_{0(\rho)} - \tau > 0$, which implies $u_{(r)} = u_{0(r)} - \tau > 0$ for each $r \leq \rho$. In addition,

$$\begin{aligned}
\sum_{r=1}^{M} a_r u_r &= \sum_{r=1}^{\rho} a_{(r)} u_{0(r)} - \sum_{r=1}^{\rho} a_{(r)} \tau = \left(1 - \frac{\lambda \sum_{r=1}^{\rho} a_{(r)}}{1 + \lambda \sum_{r=1}^{\rho} a_{(r)}}\right) \sum_{r=1}^{\rho} a_{(r)} u_{0(r)} \\
&= \frac{1}{1 + \lambda \sum_{r=1}^{\rho} a_{(r)}} \sum_{r=1}^{\rho} a_{(r)} u_{0(r)} = \frac{\tau}{\lambda}, \qquad (40) \\
\sum_{r=1}^{M} a_r (u_r - u_{0r})^2 &= \sum_{r=1}^{\rho^*} a_{(r)} \tau^2 + \sum_{r=\rho^*+1}^{\rho} a_{(r)} \tau^2 + \sum_{r=\rho+1}^{M} a_{(r)} u^2_{0(r)} \\
&< \sum_{r=1}^{\rho^*} a_{(r)} \tau^2 + \sum_{r=\rho^*+1}^{M} a_{(r)} u^2_{0(r)}. \qquad (41)
\end{aligned}$$

We next consider two cases:

1. $\boxed{\tau^* \geq \tau.}$ From (41), we have that $\sum_{r=1}^{M} a_r(u_r - u_{0r})^2 < \sum_{r=1}^{\rho^*} a_{(r)}\tau^2 + \sum_{r=\rho^*+1}^{M} a_{(r)}u_{0(r)}^2 \leq \sum_{r=1}^{\rho^*} a_{(r)}(\tau^*)^2 + \sum_{r=\rho^*+1}^{M} a_{(r)}u_{0(r)}^2 = \sum_{r=1}^{M} a_r(u_r^* - u_{0r})^2$. From (40), we have that $\left(\sum_{r=1}^{M} a_r u_r\right)^2 = \tau^2/\lambda^2 \leq (\tau^*)^2/\lambda^2$. Summing the two inequalities, we get $F(\mathbf{u}) < F(\mathbf{u}^*)$, which leads to a contradiction.

2. $\boxed{\tau^* < \tau.}$ We will construct a vector $\tilde{\mathbf{u}}$ from $\mathbf{u}^*$ and show that $F(\tilde{\mathbf{u}}) < F(\mathbf{u}^*)$. Define

$$\tilde{u}_{(r)} = \begin{cases} u^*_{(\rho^*)} - \frac{2a_{(\rho^*+1)}}{a_{(\rho^*)}+a_{(\rho^*+1)}}\epsilon, & \text{if } r = \rho^* \\ \frac{2a_{(\rho^*)}}{a_{(\rho^*)}+a_{(\rho^*+1)}}\epsilon, & \text{if } r = \rho^*+1 \\ u^*_{(r)} & \text{otherwise,} \end{cases} \tag{42}$$

where $\epsilon = (u_{0(\rho^*+1)} - \tau^*)/2$. Note that $\sum_{r=1}^{M} a_r\tilde{u}_r = \sum_{r=1}^{M} a_r u_r^*$. From the assumptions that $\tau^* < \tau$ and $\rho^* < \rho$, we have that $u^*_{(\rho^*+1)} = u_{0(\rho^*+1)} - \tau > 0$, which implies that $\tilde{u}_{(\rho^*+1)} = \frac{a_{(\rho^*)}(u_{0(\rho^*+1)}-\tau^*)}{a_{(\rho^*)}+a_{(\rho^*+1)}} > \frac{a_{(\rho^*)}(u_{0(\rho^*+1)}-\tau)}{a_{(\rho^*)}+a_{(\rho^*+1)}} = \frac{a_{(\rho^*)}u^*_{(\rho^*+1)}}{a_{(\rho^*)}+a_{(\rho^*+1)}} > 0$, and that $u^*_{(\rho^*)} = u_{0(\rho^*)} - \tau^* - \frac{a_{(\rho^*+1)}(u_{0(\rho^*+1)}-\tau^*)}{a_{(\rho^*)}+a_{(\rho^*+1)}} = u_{0(\rho^*)} - \frac{a_{(\rho^*+1)}u_{0(\rho^*+1)}}{a_{(\rho^*)}+a_{(\rho^*+1)}} - \left(1 - \frac{a_{(\rho^*+1)}}{a_{(\rho^*)}+a_{(\rho^*+1)}}\right)\tau^* >^{\text{(i)}} \left(1 - \frac{a_{(\rho^*+1)}}{a_{(\rho^*)}+a_{(\rho^*+1)}}\right)(u_{0(\rho^*+1)} - \tau) = \left(1 - \frac{a_{(\rho^*+1)}}{a_{(\rho^*)}+a_{(\rho^*+1)}}\right)(u^*_{(\rho^*+1)}) > 0$, where inequality (i) is justified by the facts that $u_{0(\rho^*)} \geq u_{0(\rho^*+1)}$ and $\tau > \tau^*$. This ensures that $\tilde{\mathbf{u}}$ is well defined. We have:

$$\begin{aligned} 2(F(\mathbf{u}^*) - F(\tilde{\mathbf{u}})) &= \sum_{r=1}^{M} a_r(u_r^* - u_{0r})^2 - \sum_{r=1}^{M} a_r(\tilde{u}_r - u_{0r})^2 \\ &= a_{(\rho^*)}(\tau^*)^2 + a_{(\rho^*+1)}u_{0(\rho^*+1)}^2 - a_{(\rho^*)}\left(\tau^* + \frac{2a_{(\rho^*+1)}\epsilon}{a_{(\rho^*)}+a_{(\rho^*+1)}}\right)^2 \\ &\quad -a_{(\rho^*+1)}\left(u_{0(\rho^*+1)} - \frac{2a_{(\rho^*)}\epsilon}{a_{(\rho^*)}+a_{(\rho^*+1)}}\right)^2 \\ &= -\frac{4a_{(\rho^*)}a_{(\rho^*+1)}\epsilon}{a_{(\rho^*)}+a_{(\rho^*+1)}}\underbrace{(\tau^* - u_{0(\rho^*+1)})}_{-2\epsilon} - \frac{4a_{(\rho^*)}a_{(\rho^*+1)}^2\epsilon^2}{\left(a_{(\rho^*)}+a_{(\rho^*+1)}\right)^2} - \frac{4a_{(\rho^*)}^2a_{(\rho^*+1)}\epsilon^2}{\left(a_{(\rho^*)}+a_{(\rho^*+1)}\right)^2} \\ &= \frac{4a_{(\rho^*)}a_{(\rho^*+1)}\epsilon^2}{a_{(\rho^*)}+a_{(\rho^*+1)}} \geq 0, \end{aligned} \tag{43}$$

which leads to a contradiction and completes the proof.

■