Large-Margin Metric Learning for Constrained Partitioning Problems

Rémi Lajugie, Sylvain Arlot, Francis Bach

Large-Margin Metric Learning for Partitioning Problems

Rémi Lajugie*[1,2], Sylvain Arlot†[1,2], and Francis Bach‡[1,2]

[1]Département d'Informatique, Ecole Normale Supérieure, Paris, France
[2]INRIA, Equipe projet SIERRA

March 5, 2013

Abstract

In this paper, we consider unsupervised partitioning problems, such as clustering, image segmentation, video segmentation and other change-point detection problems. We focus on partitioning problems based explicitly or implicitly on the minimization of Euclidean distortions, which include mean-based change-point detection, K-means, spectral clustering and normalized cuts. Our main goal is to learn a Mahalanobis metric for these unsupervised problems, leading to feature weighting and/or selection. This is done in a supervised way by assuming the availability of several potentially partially labelled datasets that share the same metric. We cast the metric learning problem as a large-margin structured prediction problem, with proper definition of regularizers and losses, leading to a convex optimization problem which can be solved efficiently with iterative techniques. We provide experiments where we show how learning the metric may significantly improve the partitioning performance in synthetic examples, bioinformatics, video segmentation and image segmentation problems.

1 Introduction

Unsupervised partitioning problems are ubiquitous in machine learning and other data-oriented fields such as computer vision, bioinformatics or signal processing. They include (a) traditional *unsupervised clustering* problems, with the classical K-means algorithm, hierarchical linkage methods [14] and spectral clustering [22], (b) *unsupervised image segmentation* problems where two neighboring pixels are encouraged to be in the same cluster, with mean-shift techniques [9] or normalized cuts [25], and (c) *change-point detection* problems adapted to multivariate sequences (such as

*remi.lajugie@ens.fr
†sylvain.arlot@ens.fr
‡francis.bach@ens.fr

video) where segments are composed of contiguous elements, with typical window-based algorithms [11] and various methods looking for a change in the mean of the features (see, e.g., [8]).

All the algorithms mentioned above rely on a specific distance (or more generally a similarity measure) on the space of configurations. A good metric is crucial to the performance of these partitioning algorithms and its choice is heavily problem-dependent. While the choice of such a metric has been originally tackled manually (often by trial and error), recent work has considered learning such metric directly from data. Without any supervision, the problem is ill-posed and methods based on generative models may learn a metric or reduce dimensionality (see, e.g., [10]), but typically with no guarantees that they lead to better partitions. In this paper, we follow [4, 32, 3] and consider the goal of learning a metric for potentially several partitioning problems sharing the same metric, assuming that several fully or partially labelled partitioned datasets are available during the learning phase. While such labelled datasets are typically expensive to produce, there are several scenarios where these datasets have already been built, often for evaluation purposes. These occur in video segmentation tasks (see Section 6.1), image segmentation tasks (see Section 6.3) as well as change-point detection tasks in bioinformatics (see [15] and Section 5.3).

In this paper, we consider partitioning problems based explicitly or implicitly on the minimization of Euclidean distortions, which include K-means, spectral clustering and normalized cuts, and mean-based change-point detection. We make the following contributions:

- We review and unify several partitioning algorithms in Section 2, and cast them as the maximization of a linear function of a rescaled equivalence matrix, which can be solved by algorithms based on spectral relaxations or dynamic programming.
- Given fully labelled datasets, we cast in Section 4 the metric learning problem as a large-margin structured prediction problem, with proper definition of regularizers, losses and efficient loss-augmented inference.
- Given partially labelled datasets, we propose in Section 5 an algorithm, iterating between labelling the full datasets given a metric and learning a metric given the fully labelled datasets. We also consider in Section 5.3 extensions that allow changes in the full distribution of univariate time series (rather than changes only in the mean), with application to bioinformatics.
- We provide in Section 6 experiments where we show how learning the metric may significanty improve the partitioning performance in synthetic examples, video segmentation and image segmentation problems.

Related work.

The need for metric learning goes far beyond unsupervised partitionning problems. [30] proposed a large margin framework for learning a metric in nearest-neighbours algorithms based on sets of must-link/must not link constraints, while [13] considers a probability-based non-convex formulation. For these works, a single dataset is fully

labelled and the goal is to learn a metric leading to good testing performance on unseen data.

Some recent work [17] proved links between metric learning and kernel learning, permitting to kernelize any Mahalanobis distance learning problem.

Metric learning has also been considered in semi-supervised clustering of a single dataset, where some partial constraints are given. This includes the works of [4, 32], both based on efficient convex formulations. As shown in Section 6, these can be used in our settings as well by stacking several datasets into a single one. However, our discriminative large-margin approach outperforms these.

Moreover, the task of learning how to partition was tackled in [3] for spectral clustering. The problem set-up is the same (availability of several fully partitioned datasets), however, the formulation is non-convex and relies on the unstable optimization of eigenvectors. In Section 5.1, we propose a convex more stable large-margin approach.

Other approaches do not require any supervision [10], and perform dimensionality reduction and clustering at the same time, by iteratively alternating the computation of a low-rank matrix and a clustering of the data using the corresponding metric. However, they are unable to take advantage of the labelled information that we use.

Our approach can also be related to the one of [26]. Given a small set of labelled instances, they use a similar large-margin framework, inspired by [29] to learn parameters of Markov random fields, using graph cuts for solving the "loss-augmented inference problem" of structured prediction. However, their segmentation framework does not apply to unsupervised segmentation (which is the goal of this paper). In this paper, we present a supervised learning framework aiming at learning how to perform an unsupervised task.

Our approach to learn the metric is nevertheless slightly different of the ones mentioned above. Indeed, we cast this problem as the solution of a structured SVM as in [29, 27]. This make our paper shares many conceptual steps with works like [7, 21] where they use a structured SVM to learn in one case weights for graph matchings and a metric for ranking in the other case.

2 Partitioning through matrix factorization

In this section, we consider T multi-dimensional observations $x_1, \dots, x_T \in \mathbb{R}^P$, which may be represented in a matrix $X \in \mathbb{R}^{T \times P}$. Partitioning the T observations into K classes is equivalent to finding an *assignment matrix* $Y \in \{0,1\}^{T \times K}$, such that $Y_{ij} = 1$ if the i-th observation is affected to cluster j and 0 otherwise. For general partitioning problems, no additional constraints are used, but for change-point detection problems, it is assumed that the segments are contiguous and with increasing labels. That is, the matrix Y is of the form

$$Y = \begin{pmatrix} \mathbf{1}_{T_1} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{1}_{T_K} \end{pmatrix},$$

where $\mathbf{1}_D \in \mathbb{R}^D$ is the D-dimensional vector with constant components equal to one, and T_j is the number of elements in cluster j. For any partition, we may re-order (non uniquely) the data points so that the assignment matrix has the same form; this is typically useful for the understanding of partitioning problems.

2.1 Distortion measure

In this paper, we consider partitioning models where each data point in cluster j is modelled by a vector (often called a centroid or a mean) $c_j \in \mathbb{R}^p$, the overall goal being to find a partition and a set of means so that the distortion measure $\sum_{i=1}^T \sum_{j=1}^K Y_{ij} \|x_i - c_j\|^2$ is as small as possible, where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^P$. By considering the Frobenius norm defined through $\|A\|_F^2 = \sum_{i=1}^T \sum_{j=1}^P A_{ij}^2$, this is equivalent to minimizing

$$\|X - YC\|_F^2 \tag{1}$$

with respect to an assignment matrix Y and the centroid matrix $C \in \mathbb{R}^{K \times P}$.

2.2 Representing partitions

Following [3, 10], the quadratic minimization problem in Y can be solved in closed form, with solution $C = (Y^\top Y)^{-1} Y^\top X$ (it can be found by computing the matrix gradient and setting it to zero). Thus, the partitioning problem (with known number of clusters K) of minimizing the distortion in Eq. (1), is equivalent to:

$$\min_{Y \in \{0,1\}^{T \times K},\ Y\mathbf{1}_K = \mathbf{1}_P} \|X - Y(Y^\top Y)^{-1} Y^\top X\|_F^2 \ . \tag{2}$$

Thus, the problem is naturally parameterized by the $T \times T$-matrix $M = Y(Y^\top Y)^{-1} Y^\top$. This matrix, which we refer to as a *rescaled equivalence matrix*, has a specific structure. First the matrix $Y^\top Y$ is diagonal, with i-th diagonal element equal to the number of elements in the cluster containing the i-th data point. Thus $M_{ij} = 0$ if i and j are in different clusters and otherwise equal to $1/D$ where D is the number of elements in the cluster containing the i-th data point. Thus, if the points are re-ordered so that the segments are composed of contiguous elements, then we have the following form

$$M = \begin{pmatrix} \mathbf{1}\mathbf{1}^\top / T_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{1}\mathbf{1}^\top / T_K \end{pmatrix}.$$

In this paper, we use this representation of partitions. Note the difference with alternative representations $YY^\top$ which has values in $\{0, 1\}$, used in particular by [18].

We denote by $\mathcal{M}_K$ the set of rescaled equivalence matrices, i.e., matrices $M \in \mathbb{R}^{T \times T}$ such that there exists an assignment matrix $Y \in \mathbb{R}^{T \times K}$ such that $M = Y(Y^\top Y)^{-1} Y^\top$. For situations where the number of clusters is unspecified, we denote by $\mathcal{M}$ the union of all $\mathcal{M}_K$ for $K \in \{1, \dots, N\}$.

Note that the number of clusters may be obtained from the trace of M, since $\operatorname{Tr} M = \operatorname{Tr} Y(Y^\top Y)^{-1}Y^\top = \operatorname{Tr}(Y^\top Y)^{-1}Y^\top Y = K$. This can also be seen by noticing that $M^2 = Y(Y^\top Y)^{-1}Y^\top Y(Y^\top Y)^{-1}Y^\top = M$, i.e., M is a projection matrix, with eigenvalues in $\{0, 1\}$, and the number of eigenvalues equal to one is exactly the number of clusters. Thus, $\mathcal{M}_K = \big\{M \in \mathcal{M},\ \operatorname{Tr} M = K\big\}$.

Learning the number of clusters K. Given the number of clusters K, we have seen from Eq. (2) that the partitioning problem is equivalent to

$$\min_{M \in \mathcal{M}_K} \|X - MX\|_F^2 = \min_{M \in \mathcal{M}_K} \operatorname{Tr}\big[XX^\top(I - M)\big]. \tag{3}$$

In change-point detection problems, an extra constraint of contiguity of segments is added.

In the common situation when the number of clusters K is unknown, then it may be estimated directly from data by penalizing the distortion measure by a term proportional to the number of clusters, as usually done for instance in change-point detection [19]. This is a classical idea that can be traced back to the AIC criterion [1] for instance. Given that the number of clusters for a rescaled equivalence matrix M is $\operatorname{Tr} M$, this leads to the following formulation:

$$\min_{M \in \mathcal{M}} \operatorname{Tr}\big[XX^\top(I - M)\big] + \lambda \operatorname{Tr} M \tag{4}$$

Note that our metric learning algorithm also learns this extra parameter λ.

Thus, the two types of partitioning problems (with fixed or unknown number of clusters) can be cast as the problem of maximizing a linear function of the form $\operatorname{Tr}(AM)$ with respect to $M \in \mathcal{M}$, with the potential constraint that $\operatorname{Tr} M = K$. In general, such optimization problems may not be solved in polynomial time. In Section 2.3, we show how adding contiguity constraints makes it possible to obtain a solution in polynomial time through dynamic programming. For general situations, the K-means algorithm, although not exact, can be used to get good partitioning in polynomial time. In Section 2.4, we provide a spectral relaxation, which we use within our large-margin framework in Section 4.

2.3 Change-point detection by dynamic programming

The change-point detection problem is a restriction of the general partitioning problem where the segments are composed of contiguous elements. We denote by $\mathcal{M}^{\text{seq}}$ the set of partition matrices for the change-point detection problem, and $\mathcal{M}_K^{\text{seq}}$, its restriction to partitions with K segments.

The problem is thus of solving Eq. (4) (known number of clusters) or Eq. (3) (unknown number of clusters) with the extra constraint that $M \in \mathcal{M}^{\text{seq}}$. In these two situations, the contiguity constraint leads to *exact* polynomial-time algorithms based on dynamic programming. See, e.g., [24]. This leads to algorithms for maximizing $\operatorname{Tr}(AM)$, when A is positive semi-definite in $O(T^2)$. When the number of segments K is known the running time complexity is $O(KT^2)$.

We now describe a reformulation that can solve $\max_{M \in \mathcal{M}} \operatorname{Tr}(AM)$ for any matrix A (potentially with negative eigenvalues, as from Eq. (4)). This algorithm is presented

in Algorithm 1. It only requires some preprocessing of the input matrix A, namely computing its summed area table I (or image integral), defined to have the same size as A and with $I_{ij} = \sum_{i' \leq i,\ j' \leq j} A_{i'j'}$. In words it is the sum of the elements of A which are above and to the left of respectively i and j. A similar algorithm can be derived in the case where $M \in \mathcal{M}_K$.

Algorithm 1 Dynamic programming for maximizing $\mathrm{Tr}(AM)$ such that $M \in \mathcal{M}$

Require: $T \times T$ matrix A
  Compute I, image integral (summed area table) of A
  Initialize $C(1,:) =$diag(I)
  for $t = 1 : T-1$ **do**
    $C(t+1, t+1) = \max(C(1:t,t)) + I(t+1, t+1)$
    for u=t+1 ... T **do**
      $\beta = \frac{I(s,s)+I(t+1,t+1)-I(s,t+1)-I(t+1,s)}{(u-t)}$
      $C(t+1, u) = \max(C(1:t,t)) + \beta$
    end for
  end for
  Backtracking steps: $t_c = T$, $Y = \emptyset$
  while $t_c \geqslant 1$ **do**
    $t_c^{\mathrm{old}} = t_c$, $t_c = \mathrm{argmax}\,\{C(t_c, :)\}$
    $s = t_c^{\mathrm{old}} - t_c + 1$, $Y = \begin{pmatrix} Y & 0 \\ 0 & \mathbf{1}_s \end{pmatrix}$
  end while
  return Matrix $M = Y(Y^\top Y)^{-1} Y^\top$.

2.4 K-means clustering and spectral relaxation

For a known number of clusters K, K-means is an iterative algorithm aiming at minimizing the distortion measure in Eq. (1): it iterates between (a) optimizing with respect to C, i.e., $C = (Y^\top Y)^{-1} Y^\top X$, and (b) minimizing with respect to Y (by assigning points to the closest centroids). Note that this algorithm only converges to a local minimum and there is no known algorithm to perform an exact decoding in polynomial time in high dimensions P. Moreover, the K-means algorithm cannot be readily applied to approximately maximize any linear function $\mathrm{Tr}\, AM$ with respect to $M \in \mathcal{M}$, i.e., when A is not positive-definite or the number of clusters is not known.

Following [25, 22, 3], we now present a spectral relaxation of this problem. This is done by relaxing the set $\mathcal{M}$ to the set of matrices that satisfy $M^2 = M$ (i.e., removing the constraint that M takes a finite number of distinct values). When the number of clusters is known, this leads to the classical spectral relaxation, i.e.,

$$\max_{M \in \mathcal{M},\ \mathrm{Tr}\, M = K} \mathrm{Tr}(AM) \leqslant \max_{M^2 = M,\ \mathrm{Tr}\, M = K} \mathrm{Tr}(AM),$$

which is equal to the sum of the K largest eigenvalues of A; the optimal matrix M of the spectral relaxation is the orthogonal projector on the eigenvectors of A with K largest eigenvalues.

When the number of clusters is unknown, we have:

$$\max_{M \in \mathcal{M}} \operatorname{Tr}(AM) \leqslant \max_{M^2 = M} \operatorname{Tr}(AM) = \operatorname{Tr}(A)_+,$$

where $\operatorname{Tr}(A)_+$ is the sum of positive eigenvalues of A. The optimal matrix M of the spectral relaxation is the orthogonal projector on the eigenvectors of A with positive eigenvalues. Note that in the formulation from Eq. (4), this corresponds to thresholding all eigenvalues of $XX^\top$ which are less than λ.

We denote by $\mathcal{M}^{\text{spec}} = \{M \in \mathbb{R}^{P \times P},\ M^2 = M\}$ and $\mathcal{M}_K^{\text{spec}} = \{M \in \mathbb{R}^{P \times P},\ M^2 = M,\ \operatorname{Tr} M = K\}$ the relaxed set of rescaled equivalence matrices.

2.5 Metric learning

In this paper, we consider learning a *Mahalanobis metric*, which may be parameterized by a positive definite matrix $B \in \mathbb{R}^{P \times P}$. This corresponds to replacing dot-products $x_i^\top x_j$ by $x_i^\top B x_j$, and $XX^\top$ by $XBX^\top$. Thus, when the number of cluster is known, this corresponds to

$$\min_{M \in \mathcal{M}_K} \operatorname{Tr}\left[XBX^\top(I - M)\right] \tag{5}$$

or, when the number of clusters is unknown, to:

$$\min_{M \in \mathcal{M}} \operatorname{Tr}\left[BX^\top(I - M)X\right] + \lambda \operatorname{Tr} M. \tag{6}$$

Note that by replacing B by $B\lambda$ and dividing the equation by λ, we may use an equivalent formulation of Eq. (6) with $\lambda = 1$, that is:

$$\min_{M \in \mathcal{M}} \operatorname{Tr}\left[XBX^\top(I - M)\right] + \operatorname{Tr} M. \tag{7}$$

The key aspect of the partitioning problem is that it is formulated as optimizing with respect to M a function *linearly* parameterized by B. The linear parametrization in M will be useful when defining proper losses and efficient loss-augmented inference in Section 4.

Note that we may allow B to be just positive semi-definite. In that case, the zero-eigenvalues of the pseudo-metric corresponds to irrelevant dimensions. That means in particular we have performed dimensionality reduction on the input data. We propose a simple way to encourage this desirable property in Section 4.3.

3 Loss between partitions

Before going further and apply the framework of Structured prediction [29] in the context of metric learning, we need to find a loss on the output space of possible partitioning which is well suited to our context. To avoid any notation conflict, we will refer in that section to $\mathcal{P}$ as a general set of partition (it can corresponds for instance to $\mathcal{M}^{\text{seq}}$).

3.1 Some standard loss

The Rand index When comparing partitions [16], a standard way to measure how different two of them are is to use the Rand [23] index which is defined, for two partitions of the same set of T elements S $P_1 = \{P_1^1, \dots, P_1^{K_1}\}$ and $P_2\{P_2^1, \dots, P_2^{K_2}\}$ as the sum of concordant pairs over the number of possible pairs. More precisely, if we consider all the possible pairs of elements of S, the concordant pairs are defined as the sum of the pairs of elements which both belong to the same set in P_1 and P_2 and of the pairs which are not in the same set both in P_1 and P_2. In matricial terms, it is linked to the Frobenius distance between the equivalence matrices representing P_1 and P_2 (these matrices are binary matrices of size $T \times T$ which are 1 if and only if the element i and the element j belong to the same set of the partition).
This loss is not necessarily very well suited to our problem, since intuitively one can see that it doesn't take into account the size of each subset inside the partition, whereas our concern is to optimize intra class variance which is a rescaled indicator.

Hausdorff distance In the change-point detection litterature, a very common way to measure dissimilarities between partitions is the so-called Hausdorff distance [6] on the elements of the frontier of the elements of the partitions (the need for a frontier makes it inapplicable directly to the case of general clustering). Let's consider two partitions of a finite set S of T elements. We assume that the elements have a sequential order and thus elements of partitions P_1 and P_2 have to be contiguous. It is then possible to define the frontier (or set of ruptures) of P_1 as the collection of indexes $\partial P_1 = \{\inf P_1^2, \dots, \inf P_1^K\}$. Then, by embedding the set S into $[0,1]$ (it corresponds just to normalize the time indexes so that they are in $[0,1]$), we can consider a distance d on $[0,1]$, (typically the absolute value) and then define the associated Hausdorff distance $d_H(P_1, P_2) = \max\{\sup_{x\in\partial P_1} \inf_{y\in\partial P_2} d(x,y), \sup_{y\in\partial P_2} \inf_{x\in\partial P_1} d(x,y)\}$

The loss considered in our context In this paper, we consider the following loss, which was originated proposed in a slightly different form by [16] and has then been widely used in the field of clustering [3]. This loss is a variation of the χ^2 association in a $K_1 \times K_2$ contingency table (see [16]). More precisely, if we consider the contingency table associated to P_1 (partition of a set of size T) with K_1 elements and P_2 with K_2 elements (the contingency table being the $K_1 \times K_2$ table C such that $C_{i,j} = n_{ij}$ the number of elements in element i of P_1 and in element j of P_2), we have that $\|M - N\|_F^2 = K_1 + K_2 - \frac{\chi^2(C)+T}{T}$.

$$\ell(M,N) = \frac{1}{T}\|M-N\|_F^2 = \frac{1}{T}\big(\operatorname{Tr}(M) + \operatorname{Tr}(N) - 2\operatorname{Tr}(MN)\big). \tag{8}$$

Moreover, if the partitions encoded by M and N have clusters $P_1^1, \dots, P_1^{K_1}$ and $P_2^1, \dots, P_2^{K_2}$, then $T\ell(M,N) = K_1 + K_2 - 2\sum_{k,l} \frac{|A_k \cap B_l|^2}{|A_k|\cdot|B_l|}$. This loss is equal to zero if the partitions are equal, and always less than $\frac{1}{T}(K+L-2)$. Another equivalent interpretation of this index is given by, with the usual convention that for the element

of S indexed by i $P_1(i)$ is the subset of P_1 where i belongs:

$$T\ell(M,N) = K_1 + K_2 - 2\sum_{i=1}^{T} \frac{|P_1(i) \cap P_2(i)|}{|P_1(i)| \times |P_2(i)|}.$$

This index seems intuitively much more suited to the study of the problem of variance minimization since it involves the rescaled equivalence matrices which parametrize naturally these kind of problems. We examine in the Appendix more facts about these losses and their links, especially about the asymptotic behaviour of the loss we use in the paper. We also show a link between this loss and the Hausdorff in the case of change-point detection.

4 Structured prediction for metric learning

As shown in the previous section, our goal is to learn a positive definite matrix B, in order to improve the performance of structured output algorithm that minimizes with respect to $M \in \mathcal{M}$, the following cost function of Eq. 7. Using the change of variable described in the table below, the partitioning problem may be cast as

$$\max_{M \in \mathcal{M}} \langle w, \varphi(X, M) \rangle \text{ or } \max_{M \in \mathcal{M}_K} \langle w, \varphi(X, M) \rangle.$$

where $\langle A, B \rangle$ is the Frobenius dot product.

Number of clusters	$\varphi(X, M)$	w
Known $(\operatorname{Tr} M = K)$	$X^\top M X$	B
Unknown	$\frac{1}{T}\begin{pmatrix} X^\top M X & 0 \\ 0 & M \end{pmatrix}$	$\begin{pmatrix} B & 0 \\ 0 & -I \end{pmatrix}$

We denote by $\mathcal{F}$ the vector space where the vector w defined above belongs to. Our goal is thus to estimate $w \in \mathcal{F}$ from N pairs of observations $(X_i, M_i) \in \mathcal{X} \times \mathcal{M}$. This is exactly the goal of large-margin structured prediction [29], which we now present. We denote by $\mathcal{N}$ a generic set of matrices, which may either be $\mathcal{M}$, $\mathcal{M}^{\text{spec}}$, $\mathcal{M}^{\text{seq}}$, $\mathcal{M}_K$, $\mathcal{M}_K^{\text{spec}}$, $\mathcal{M}_K^{\text{seq}}$, depending on the situation (see Section 4.2 for specific cases).

4.1 Large-margin structured output learning

In the margin-rescaling framework of [29], using a certain loss $\ell : \mathcal{N} \times \mathcal{N} \to \mathbb{R}_+$ between elements of $\mathcal{N}$ (here partitions), the goal is to minimize with respect to $w \in \mathcal{F}$,

$$\frac{1}{N}\sum_{i=1}^{N} \ell\big(\operatorname{argmax}_{M \in \mathcal{N}} \langle w, \varphi(X_i, M) \rangle, M_i\big) + \Omega(w),$$

where Ω is any (typically convex) regularizer. This framework is standard in machine learning in general and metric learning in particular (see e.g, [17]). This loss function

$w \mapsto \ell\big(\operatorname{argmax}_{M\in\mathcal{N}}\langle w, \varphi(X_i, M)\rangle, M_i\big)$ is not convex in M, and may be replaced by the convex surrogate

$$L_i(w) = \max_{M\in\mathcal{N}} \big\{\ell(M, M_i) + \langle w, \varphi(X_i, M) - \varphi(X_i, M_i)\rangle\big\},$$

leading to the minimization of

$$\frac{1}{N}\sum_{i=1}^{N} L_i(w) + \Omega(w). \tag{9}$$

In order to apply this framework, several elements are needed: (a) a regularizer Ω, (b) a loss function ℓ, and (c) the associated efficient algorithms for computing L_i, i.e., solving the *loss-augmented inference* problem $\max_{M\in\mathcal{N}} \big\{\ell(M, M_i)+\langle w, \varphi(X_i, M)-\varphi(X_i, M_i)\rangle\big\}$.

As discussed in Section 3, a natural loss on our output space is given by the Frobenius norm of the rescaled equivalence matrices associated to partitions.

4.2 Loss-augmented inference problem

Efficient minimization is key to the applicability of large-margin structured prediction and this problem is a classical computational bottleneck. In our situation the cardinality of $\mathcal{N}$ is exponential, but the choice of loss between partitions lead to the problem $\max_{M\in\mathcal{N}} \operatorname{Tr}(A_i M)$ where:

- $A_i = \frac{1}{T}(X_i B X_i^\top - 2M_i + \mathrm{Id})$ if the number of clusters is known.
- $A_i = \frac{1}{T}(X_i B X_i^\top - 2M_i)$ otherwise.

Thus, the loss-augmented problem may be performed for the change-point problems exactly (see Section 2.3) or through a spectral relaxation otherwise (see Section 2.4). Namely, for change-point detection problems, $\mathcal{N}$ is either $\mathcal{M}^{\mathrm{seq}}$ or $\mathcal{M}_K^{\mathrm{seq}}$, while for general partitioning problems, it is either $\mathcal{M}^{\mathrm{spec}}$ or $\mathcal{M}_K^{\mathrm{spec}}$.

4.3 Regularizer

We may consider several parametrizations/regularizers for our positive semidefinite matrix B. We may classically (see e.g, [17]) penalize $\operatorname{Tr} B^2 = \|B\|_F^2$, which is the classical squared Euclidean norm. However, two variants of our algorithm are often needed for practical problems.

Diagonal metric. To limit the number of parameters, we may be interested in only reweighting the different dimensions of the input data, i.e., we can impose the metric to be diagonal, i.e, $B = \operatorname{Diag}(b)$ where $b \in \mathbb{R}^P$. Then, the constraint is $b \geqslant 0$, and we may penalize by $\|b\|_1 = \mathbf{1}_P^\top b$ or $\|b\|_2^2$, depending whether we want to promote zeros in b (i.e., to do feature selection).

Low-rank metric. Another potentially desirable property is the interpretability of the obtained metric in terms of its eigenvectors. Ideally we want to have a pseudo-metric with a small rank. As it is classically done, we relaxed it into the sum of singular values. Here, since the matrix B is symmetric positive definite, this is simply the trace $\operatorname{Tr}(B)$.

4.4 Optimization

In order to optimize the objective function of Eq. (9), we can use several optimization techniques. This objective present the drawback of being non-smooth and thus the convergence speed that we can expect are not very fast.
In the structured prediction litterature, the most common solvers are based on cutting-plane methods (see [29]) which can be used in our case for small dimensional-problem (i.e., low P). Otherwise we use a projected subgradient method, which leads to more numerous but cheaper iterations. Cutting plane and Bundle methods [28] shows the best speed performances when the dimension of the feature space of the data to partition is low, but were empirically outperformed by a subgradient in the very high dimensional setting.

5 Extensions

We now present extensions which make our metric learning more generally applicable.

5.1 Spectral clustering and normalized cuts

Normalized cut segmentation is a graph-based formulation for clustering aiming at finding roughly balanced cuts in graphs [25]. The input data X is now replaced by a similarity matrix $W \in \mathbb{R}_+^{T\times T}$ and, for a known number of clusters K, as shown by [22, 3], it is exactly equivalent to

$$\max_{M\in\mathcal{M}_K} \operatorname{Tr}\left[M\widetilde{W}\right],$$

where $\widetilde{W} = \operatorname{Diag}(W\mathbf{1})^{-1/2} W \operatorname{Diag}(W\mathbf{1})^{-1/2}$ is the normalized similarity matrix.

Parametrization of the similarity matrix W. Typically, given data points $x_1, \ldots, x_T \in \mathbb{R}^P$ (in image segmentation problem, these are often the concatenation of the positions in the image and local feature vectors), the similarity matrix is computed as

$$(W_B)_{ij} = \exp\big(-(x_i - x_j)^\top B(x_i - x_j)\big), \tag{10}$$

where B is a positive semidefinite matrix. Learning the matrix B is thus of key practical importance.

However, our formulation would lead to efficiently learning (as a convex optimization problem) parameters only for a linear parametrization of $\widetilde{W}$. While the linear combination is attractive computationally, we follow the experience from the supervised setting where learning linear combinations of kernels, while formulated as a convex problem, does not significantly improve on methods that learn the metric within a Gaussian kernel with non-convex approaches (see, e.g., [12, 20]).

We thus stick to the parametrization of Eq. (10). In order to make the problem simpler and more tractable, we consider spectral clustering directly with W and not with its normalized version, i.e., our partitioning problem becomes

$$\max_{M\in\mathcal{M}} \operatorname{Tr} WM \text{ or } \max_{M\in\mathcal{M}_K} \operatorname{Tr} WM.$$

In order to solve the previous problem, the spectral relaxation outlined in Section 2.4 may be used, and corresponds to computing the eigenvectors of W (the first K ones if K is known, and the ones corresponding to eigenvalues greater than a certain threshold otherwise).

Non-convex optimization. In our structured output prediction formulation, the loss function for the i-th observation becomes (for the case where the number of clusters is known):

$$\begin{aligned} & \max_{M \in \mathcal{M}_K^{\text{spec}}} \left\{ \ell(M, M_i) + \operatorname{Tr} W_B (M - M_i) \right\} \\ = & -\operatorname{Tr} W_B M_i + \max_{M \in \mathcal{M}_K^{\text{spec}}} \left\{ \ell(M, M_i) + \operatorname{Tr} W_B M \right\}. \end{aligned}$$

It is not a convex function of B, however, it is a difference of a concave and a convex function, which can be dealt with using majorization-minimization algorithm [33]. The idea of this algorithm is simply to upper-bound the concave part $-\operatorname{Tr} W_B M_i$ by its linear tangent. Then the problem becomes convex and can be optimized using one of the method proposed in Section 4.4 We then iterate the process, which is known to be converging to a stationary point.

5.2 Partial labellings

The large-margin convex optimization framework relies on fully labelled datasets, i.e., pairs (X_i, M_i) where X_i is a dataset and M_i the corresponding rescaled equivalence matrix. In many situations however, only partial information is available. In these situations, starting from the PCA metric, we propose to iterate between (a) label all datasets using the current metric and respecting the constraints imposed by the partial labels and (b) learn the metric using Section 4 from the fully labelled datasets. See an application in Section 6.1.

5.3 Detecting changes in distribution of temporal signals

In sequential problems, for now, we are just able to detect changes in the mean of the distribution of time series but not to detect change-points in the whole distribution (e.g., the mean may be constant but the variance piecewise constant). Let us consider a temporal series X in which some breakpoints occur in the distribution of the data. From this single series, we build several series permitting to detect these changes, by considering features built from X, in which the change of distribution appears as a change in mean. A naive way would be to consider the moments of the data $X, X^2, X^3, \ldots, X^r$ but unfortunately as r grows these moments explode. A way to prevent them from exploding is to use the robust Hermite moments [31]. These moments are computed using the Hermite functions and permit to consider the p-dimensional series $H_1(X), H_2(X), \ldots$, where $H_i(X)$ is the i-th Hermite function $H_i(x) = 2\sqrt{2^i \pi i!} e^{-\frac{x^2}{2\sigma^2}} (-1)^i 2^{i/2} e^{\frac{x^2}{2}} \frac{d^i}{dx^i} \left(e^{\frac{-x^2}{2}} \right)$.

Bioinformatics application. Detection of change-points in DNA sequences for cancer prognosis provides a natural testbed for this approach. Indeed, in this field, researchers face data which are linked to the number of copies of each gene along the DNA (a-CGH data as used in [15]). The presence of such changes are generally related to the development of certain types of cancers. On the data from the Neuroblastoma dataset [15], some caryotypes with changes of distribution were manually annotated. Without any metric learning, the global error rate in change-point identification is 12%. By considering the first 5 Hermite moments and learning a metric, we reach a rate of 6.9%, thus improving significantly the performance.

6 Experiments

We have conducted a series of experiments showing improvements of our large-margin metric learning methods over previous metric learning techniques.

6.1 Change point detection

Synthetic examples and robustness to lack of information. We consider 300-dimensional time series of length $T = 600$ with an unknown number of breakpoints. Among these series only 10 are relevant to the problem of change-point detection, i.e., 290 series have abrupt changes which should be discarded. Since the identity of the 10 relevant time series is unknown, by learning a metric we hope to obtain high weights on the relevant series and small weights on the others. The number of segments is not assumed to be known and is learned automatically.

Moreover, in this experiment we progressively remove information, in the sense that as input of the algorithm we only give a fraction of the original time series (and we measure the amount of information given through the ratio of the given temporal series compared to the original one). Results are presented in Figure 1. As expected, the performance without metric learning is bad, while it is improved with PCA. Techniques such as RCA [4] which use the labels improve even more (all datasets were stacked into a single one with the corresponding supervision); however, it is not directly adapted to change-point detection, it requirse dimensionality reduction to work and the performance is not robust to the choice of the number of dimensions. Note also that all methods except ours are given the exact number of change-points. Our large-margin approach outperforms the other metric, in the convex setting (i.e., extreme right of the curves), but also in partially-supervised setting where we use the alternative approach describe in Section 5.2.

Video segmentation. We applied our method to data coming from old TV shows (the length of the time series in that case is about 5400, with 60 to 120 change-points) where some speaking passages alternate with singing ones. The videos are from 1h up to 1h30 long. We aim at recovering the segmentation induced by the speaking parts and the musical ones. Following [2], we use GIST features for the video part and MFCC features for the audio. The features were aggregated every second so that the temporal series we are considering are about several thousands vectors long, which is

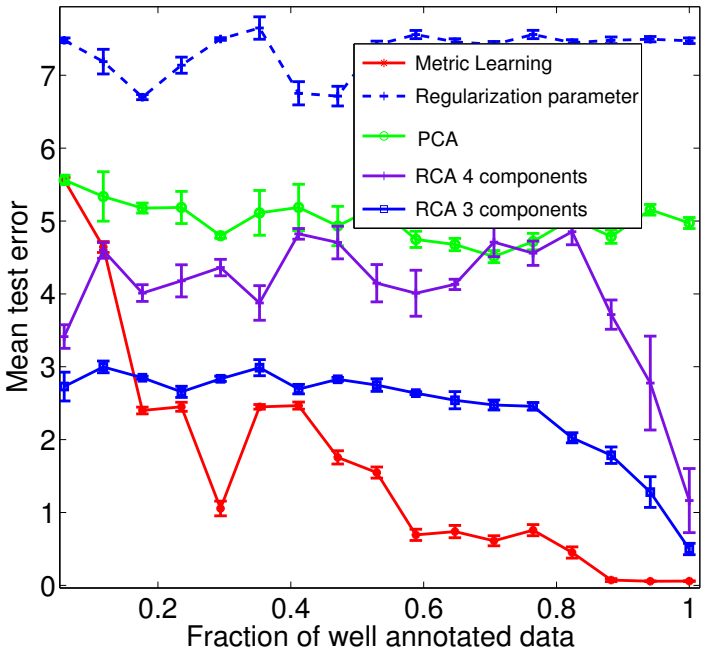

Figure 1: Performances on synthetic data vs. the quantity of information available in the time series. Note the small error bars. We compare ourselves against a metric learned by RCA (with 3 or 4 components), an exhaustive search for one regularization parameter, and PCA.

Table 1: Empirical performance on each of the three TV shows used for testing. Each subcolumn stands for a different TV show. The smaller the loss is, the better the segmentation is.

Method	Audio			Video			Both		
PCA	23	41	34	40	55	25	29	53	37
Reg. parameter	29	48	33	59	55	47	40	48	36
Metric learning	**6.1**	**9.3**	**7**	**10**	**14**	**11**	**8.7**	**9.6**	**7.8**

Table 2: Performance of the metric learning versus the Euclidean distance, and other metric learning algorithms such as RCA or [32]. We use the loss from Eq. (8).

Dataset	Ours	Euclidean	RCA	[32]
Iris	$\mathbf{0.18} \pm 0.01$	0.55 ± 10^{-11}	0.43 ± 0.02	0.30 ± 0.01
Wine	1.03 ± 0.04	3.4 ± 3.10^{-4}	$\mathbf{0.88} \pm 0.14$	3.08 ± 0.1
Letters	34.5 ± 0.1	41.62 ± 0.2	34.8 ± 0.5	35.26 ± 0.1
Mov. Libras	14 ± 1	15 ± 0.2	22 ± 2	15.07 ± 1

still computationally tractable using the dynamic programming of Algorithm 1. We used 4 shows for train, 3 for validation, 3 for test. The running times of our Matlab implementation were in order of a few hours.

The results are described in Table 1. We consider three different settings: using only the image stream, only the audio stream or both. In these three cases, we consider using the existing metric (no learning), PCA, or our approach. In all settings, metric learning improves performance. Note that the performance is best with only the audio stream and our metric learning, given both streams, manages to do almost as well as with only the audio stream, thus illustrating the robustness of using metric learning in this context.

6.2 K-means clustering

Using the partition induced by the classes as ground truth, we tested our algorithm on some classification datasets from the UCI machine learning repository, using the classification information as partitions, following the methodology proposed by [32]. This application of our framework is a little extreme in the sense that we assume only one partitioning as training point (i.e., $N = 1$). The results are presented in Table 2. For the "Letters" and "Mov. Libras" datasets, there are no significant differences, while for the "Wine" dataset, RCA is the best, and for the "Iris" dataset, our large-margin approach is best: even in this extreme case, we are competitive with existing techniques.

Table 3: Performance of the metric learned in the context of image segmentation, comparing the result of a learned metric vs. the results of an exhaustive grid search (Grid). σ is the standard deviation of the difference between the loss with our metric and the grid search. To assess the significance of our results, we perform t-tests whose p-values are respectively 2.10^{-9} and 4.10^{-9}.

Loss used	Learned metric	Grid	σ
Loss of Eq. (8)	1.54	1.77	0.3
Jaccard distance	0.45	0.53	0.11

6.3 Image Segmentation

We now consider learning metrics for normalized cuts and consider the Weizmann horses database [5], for which groundtruth segmentation is available. Using color and position features, we learn a metric with the method presented in Section 5.1 on 10 fully labelled images. We then test on the remaining 318 images.

We compare the results of this procedure to a cross-validation approach with an exhaustive search on a 2D grid adjusting one parameter for the position features and one other for color ones. The loss between groundtruth and segmentations obtained by the normalized cuts algorithm is measured either by Eq. (8) or the Jaccard distance. Results are summarized in Table 3, with some visual examples in Figure 2. The metric learning within the Gaussian kernel significantly improves performance. The running times of our pure Matlab implementation were in order of several hours to get convergence of the convex-concave procedure we used.

7 Conclusion

We have presented a large-margin framework to learn metrics for unsupervised partitioning problems, with application in particular to change-point detection in video streams and image segmentation, with a significant improvement in partitioning performance. For the applicative part, following recent trends in image segmentation (see, e.g., [18]), it would be interesting to extend our change-point framework so that it allows unsupervised co-segmentation of several videos: each segment could then be automatically labelled so that segments from different videos but with the same label correspond to the same action.

References

[1] H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716 – 723, dec 1974.

[2] S. Arlot, A. Celisse, and Z. Harchaoui. Kernel change-point detection, Feb. 2012. arXiv:1202.3878.

[3] F. Bach and M. Jordan. Learning spectral clustering. In *Adv. NIPS*, 2003.

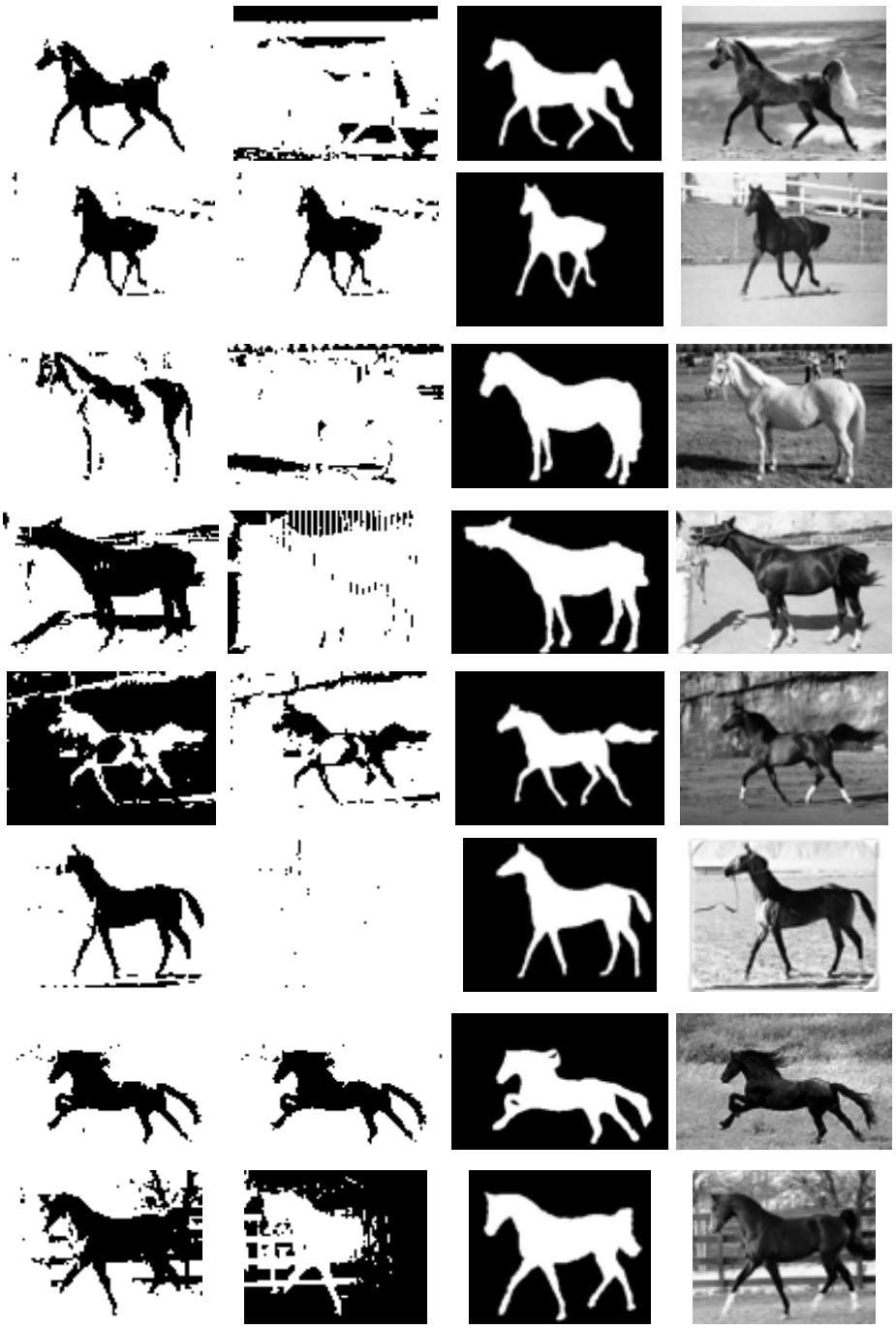

Figure 2: From left to right: image segmented with our learned metric, image segmented by a parameter adjusted by exhaustive search, groundtruth segmentation, original image in gray.

[4] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6(1):937, 2006.

[5] E. Borenstein and S. Ullman. Learning to segment. In *Proc. ECCV*, 2004.

[6] L. Boysen, A. Kempe, V. Liebscher, A. Munk, and O. Wittich. Consistencies and rates of convergence of jump-penalized least squares estimators. *Annals of Statistics*, 37:157–183.

[7] T. S. Caetano, L. Cheng, Q. V. Le, and A. J. Smola. Learning Graph Matching. In *IEEE 11th International Conference on Computer Vision (ICCV 2007)*, pages 1–8, 2007.

[8] J. Chen and A. K. Gupta. *Parametric Statistical Change Point Analysis*. Birkhäuser, 2011.

[9] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. PAMI*, 17(8):790–799, 1995.

[10] F. De la Torre and T. Kanade. Discriminative cluster analysis. In *Proc. ICML*, 2006.

[11] F. Desobry, M. Davy, and C. Doncarli. An online kernel change detection algorithm. *IEEE Trans. Sig. Proc.*, 53(8):2961–2974, 2005.

[12] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *Proc. ICCV*, 2009.

[13] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Adv. NIPS*, 2004.

[14] J. C. Gower and G. J. S. Ross. Minimum spanning trees and single linkage cluster analysis. *Applied statistics*, pages 54–64, 1969.

[15] T. Hocking, G. Schleiermacher, I. Janoueix-Lerosey, O. Delattre, F. Bach, and J.-P. Vert. Learning smoothing models of copy number profiles using breakpoint annotations. *HAL, archives ouvertes*, 2012.

[16] L. J. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[17] P. Jain, B. Kulis, J. V. Davis, and I. S. Dhillon. Metric and kernel learning using a linear transformation. *J. Mach. Learn. Res.*, 13:519–547, Mar. 2012.

[18] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *Proc. CVPR*, 2010.

[19] M. Lavielle. Using penalized contrasts for the change-point problem. *Signal Proces.*, 85(8):1501–1510, 2005.

[20] M. Marszałek, C. Schmid, H. Harzallah, and J. Van De Weijer. Learning object representations for visual object class recognition. Technical Report 00548669, HAL, 2007.

[21] B. Mcfee and G. Lanckriet. Metric learning to rank. In *In Proceedings of the 27th annual International Conference on Machine Learning (ICML*, 2010.

[22] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Adv. NIPS*, 2002.

[23] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):pp. 846–850, 1971.

[24] G. Rigaill. Pruned dynamic programming for optimal multiple change-point detection. Technical Report 1004.0887, arXiv, 2010.

[25] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. PAMI*, 22:888–905, 1997.

[26] M. Szummer, P. Kohli, and D. Hoiem. Learning CRFs using graph cuts. *Proc. ECCV*, 2008.

[27] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. *Adv. NIPS*, 2003.

[28] C. H. Teo, S. Vishwanathan, A. Smola, and V. Quoc. Bundle methods for regularized risk minimization. *Journal of Machine Learning research*, 2009.

[29] I. Tsochantaridis, T. Hoffman, T. Joachims, and Y. Altun. Support vector machine learning

for interdependent and structured output spaces. *Journal of Machine Learning Research*, 2005.

[30] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Adv. NIPS*, 2006.

[31] M. Welling. Robust higher order statistics. *Proc. Int. Workshop Artif. Intell. Statist.(AISTATS, 2005)*.

[32] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with applications to clustering with side-information. *Adv. NIPS*, 2002.

[33] A. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.

A Asymptotics of the loss between partitions

Note that in this section, we will denote by d_F^2 the "normalized" loss between partitions. This means that, with the notations of the article when considering two matrices M and N representing some partitions P and Q in the generic set of partitions $\mathcal{P}$, we have $Td_F^2 = \|M - N\|_F^2$. Throughout this section, we will refer to the size of a partition as the number of clusters.

A.1 Hypothesis

- We assume we consider P and Q two partitions of the same size, with a common number of clusters K.
- $\forall k, l \in \{1, \dots, K\}$, we denote $\epsilon_{k\to l} = |P_k \cap Q_l|$, the flow which goes out from P to Q when P goes to Q.
- We define the global outer flow as $\epsilon_{k\to} = \sum_{l\neq k} \epsilon_{k\to l}$ and the global inner flow as $\epsilon_{\to l} = \sum_{l\neq k} \epsilon_{k\to l}$

A.2 Main result

Theorem 1. *Let P and Q two partitions satisfying our hypothesis. If we note $M(P,Q) = \max_{k\neq l} \left\{ \frac{\epsilon_{k\to l}}{min(|P_k||P_l|)} \right\}$, then $\exists \delta : \mathcal{P}^2 \to \mathbb{R}$ such that $\sup_{P,Q,K\times M(P,Q)\leq \epsilon} |\delta(P,Q)| \to_{\epsilon\to 0} 0$ and $\forall P, Q \in \mathcal{P}$ of the same size K, T,*

$$Td_F^2(P,Q) = 2\sum_{k=1}^{K} \left(\frac{\epsilon_{k\to} + \epsilon_{\to k}}{|P_k|}\right) \times (1 + \delta(P,Q))$$

Proof. From the expressions of Section 3.1, we can write :

$$\begin{aligned} d_F^2(P,Q) &= 2K - 2\sum_{k,l} \frac{|P_k \cap Q_l|^2}{|Q_l||P_k|} \\ &= 2\sum_{k=1}^{K} (1 - \frac{|P_k \cap Q_k|^2}{|Q_k||P_k|}) + 2\sum_{k\neq l} \frac{\epsilon_{k\to l}^2}{|P_k|(|P_l| - \epsilon_{\to} + \epsilon_{\to k})} \end{aligned}$$

The second term can be pretty easily bounded using M

$$2\sum_{k\neq l}\frac{\epsilon_{k\to l}^2}{|P_k|(|P_l|-\epsilon_{k\to}+\epsilon_{\to l})}\leq 2M\sum_{k\neq l}\frac{\epsilon_{k\to l}}{|P_l|-\epsilon_{k\to}}.$$

We can go further, noticing that $\epsilon_{k\to}\leq KM|P_k|$, which leads eventually to, if $M\leq 1/2K$ (and this is the case if δ tends to 0 in the sense of the assumption of the theorem):

$$2\sum_{k\neq l}\frac{\epsilon_{k\to l}^2}{|P_k|(|P_l|-\epsilon_{\to}+\epsilon_{\to k})}\leq 2M\sum_{k\neq l}\frac{\epsilon_{k\to l}}{|P_l|-\epsilon_{k\to}}\leq 4M\sum_{k\neq l}\frac{\epsilon_{k\to l}}{P_l}.$$

Now, let's bound the first term, which is a little more long:

$$\begin{aligned}1-\frac{|P_k\cap Q_k|^2}{|Q_k||P_k|} &= 1-\frac{(|P_k-\epsilon_{k\to})^2}{|P_k(|P_k|-\epsilon_{k\to}+\epsilon_{\to k})}\\ &= \frac{\epsilon_{k\to}+\epsilon_{\to k}}{|P_k|}\times\big(\frac{1}{1+\frac{-\epsilon_{k\to}+\epsilon_{\to k}}{|P_k|}}\big)-\frac{\epsilon_{k\to}^2}{|P_k|(|P_k|-\epsilon_{k\to}+\epsilon_{\to k})}\end{aligned}$$

But, for the same reasons as when we bounded the second term

$$\frac{\epsilon_{k\to}^2}{|P_k|(|P_k|-\epsilon_{k\to}+\epsilon_{\to k})}\leq 2\sum_{k=1}^{K}\frac{\epsilon_{k\to}^2}{|P_k|^2}.$$

Using the fact that $\forall k,(K)M\geq\frac{\epsilon_{k\to}}{|P_k|}$, we finally get that, when $M\leq 1/2K$:

$$\frac{\epsilon_{k\to}^2}{|P_k|(|P_k|-\epsilon_{k\to}+\epsilon_{\to k})}\leq 4M\sum_{k=1}^{K}\frac{\epsilon_{k\to}}{|P_k|}.$$

Thus, putting everything together, when $KM\to 0$, we get the statement of the theorem. □

B Equivalence between the loss between partition and the Hausdorff distance for change point detection

As mentioned in the title of this , there is a deep link between the Hausdorff distance and the distance between partition we used throughout this paper in the case of change-point detection applications. We propose here to show that the two distances are equivalent.

B.1 Hypothesis and notations

- We consider the segmentations P and Q has having been embedded in $[0,1]$ so that we can consider a distance d on $[0,1]$ to define the Hausdorff distance between the frontiers of the elements of P and Q.
- We denote $l_m(P)$ the minimal length of a segment in a partition $P \in \mathcal{P}$ and l_{ma} the maximal one.
- We denote by d_h the Hausdorff distance between partitions as described in Section 3

B.2 Main result

Theorem 2. *Let P,Q denote two partitions. If* $|P| = |Q|$ *and* $d_h(P,Q) = \epsilon < \frac{1}{2} l_m(P)$, *then we have the following:*

$$\frac{\epsilon}{l_{ma}(P)} \leq d_F^2(P,Q) \leq 12K \frac{\epsilon}{l_m(P)}.$$

Moreover, without assuming $|P| = |Q|$, *we get*

$$d_F^2(P,Q) \geq \frac{\epsilon}{\max(l_{ma}(P), l_m(Q)))} \geq \frac{\epsilon}{T}$$

Proof. First, let's do the majorization part Using the expressions of Section 3.1, we have to minorate $\sum_{k,l=1}^{K} \frac{|P_k \cap Q_l|^2}{|P_k||Q_l|}$. Note that the hypothesis of the Hausdorff distane being inferior to the half of the minimal length is just here to say that the l-th segment of partition Q can only overlap with $l-1$-th, lth and $l+1$-th elements of P. Thus :

$$\begin{aligned}
\sum_{k,l=1}^{K} \frac{|P_k \cap Q_l|^2}{|P_k||Q_l|} &= \sum_{k,=1}^{K} \frac{|P_k \cap Q_k|^2}{|P_k||Q_k|} + \sum_{k,=1}^{K-1} \frac{|P_k \cap Q_{k+1}|^2}{|P_k||Q_{k+1}|} + \sum_{k=0}^{K-1} \frac{|P_k \cap Q_{k-1}|^2}{|P_k||Q_{k-1}|} \\
&\geq \sum_{k=1}^{K} \frac{(|P_k| - 2\epsilon)^2}{|P_k| + 2\epsilon} \\
&= \sum_{k=1}^{K} \frac{((1 - 2\frac{\epsilon}{|P_k|})^2}{1 + 2\frac{\epsilon}{|P_k|}} \\
&\geq K - 6\epsilon \sum_{k=1}^{K} \frac{1}{|P_k|} \\
&\geq K - 6\frac{\epsilon K}{l_m(P)}
\end{aligned}$$

which gives us the majorization. Note that we used the fact that $\forall x \in [0,1]$, the inequality $\frac{(1-x)^2}{1+x} \geq 1 - 3x$ holds.

For the minoration, note that it is true all the time, but we will just give the proof in the case where the Hausdorff distance is such that $d_h(P,Q) \leq l_m(P)/2$ and where $|P| = |Q|$.

First, let's begin by some general statements :
i)By definition $\epsilon = \max\{\max_{\bar{P}_i \in \partial P} \min_{\bar{Q}_j \in \partial Q} d(\bar{P}_i, \bar{Q}_j) \max_{\bar{Q}_i \in \partial Q} \min_{\bar{P}_j \in \partial P} d(\bar{Q}_i, \bar{P}_j)\}$.
ii) If the first term in the max is attained, that means there exists some (i^*, j^*) such that $|\bar{P}_{i^*} - \bar{Q}_{j^*}| = \epsilon$. It also means that, if we look at the sequences, there is no elements of ∂Q is between $\bar{P}_{i^*}$ and $\bar{Q}_{j^*}$. Thus, by definition of the loss $d_F^2(P,Q) \geq 2\sum_{\alpha \in P_j \cap Q_{j^*-1}, \beta \in P_j \setminus Q_{j^*-1}} \frac{1}{P_j^*}^2$, and a short computation leads to $d_F^2(P,Q) \geq 2\frac{\epsilon}{|P_j^*|}(1 - \frac{\epsilon}{|P_{j^*}|})_+$.
iii) If the second term in the max is attained, the same minoration holds by permuting indices.
Let's go back to our special case, we have $|P_i^*| > 2\epsilon$ and $|Q_i^*| > 2\epsilon$. This leads to

$$d_F^2(P,Q) \geq \max(\frac{\epsilon}{l_{ma}(P)}, \frac{\epsilon}{l_{ma}(Q)})$$

□