



**HAL**  
open science

## Wasserstein Discriminant Analysis

Rémi Flamary, Marco Cuturi, Nicolas Courty, Alain Rakotomamonjy

► **To cite this version:**

Rémi Flamary, Marco Cuturi, Nicolas Courty, Alain Rakotomamonjy. Wasserstein Discriminant Analysis. Machine Learning, 2018, 107 (12), pp.1923-1945. 10.1007/s10994-018-5717-1 . hal-01377528

**HAL Id: hal-01377528**

**<https://hal.science/hal-01377528v1>**

Submitted on 7 Oct 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Wasserstein Discriminant Analysis

---

**Rémi Flamary**

Université Côte d'Azur  
remi.flamary@unice.fr

**Marco Cuturi**

Kyoto University  
mcuturi@i.kyoto-u.ac.jp

**Nicolas Courty**

University of Bretagne Sud  
courty@univ-ubs.fr

**Alain Rakotomamonjy**

Rouen University  
alain.rakoto@insa-rouen.fr

## Abstract

Wasserstein Discriminant Analysis (WDA) is a new supervised method that can improve classification of high-dimensional data by computing a suitable linear map onto a lower dimensional subspace. Following the blueprint of classical Linear Discriminant Analysis (LDA), WDA selects the projection matrix that maximizes the ratio of two quantities: the dispersion of projected points coming from different classes, divided by the dispersion of projected points coming from the same class. To quantify dispersion, WDA uses regularized Wasserstein distances, rather than cross-variance measures which have been usually considered, notably in LDA. Thanks to the underlying principles of optimal transport, WDA is able to capture both global (at distribution scale) and local (at samples scale) interactions between classes. Regularized Wasserstein distances can be computed using the Sinkhorn matrix scaling algorithm; We show that the optimization of WDA can be tackled using automatic differentiation of Sinkhorn iterations. Numerical experiments show promising results both in terms of prediction and visualization on toy examples and real life datasets such as MNIST and on deep features obtained from a subset of the Caltech dataset.

## 1 Introduction

Dimensionality reduction techniques can convert high-dimensional data with potentially redundant features into low-dimensional vectors [1, 2]. In doing so, dimensionality reduction can alleviate ensuing computations, and, ideally, extract relevant features out of noisy ones. Dimensionality reduction techniques come in all flavors: Some techniques only consider linear transformations of the data, whereas some extract features using nonlinear functions; Some of these techniques, such as PCA, are designed for unsupervised settings, whereas some can consider labeled data and fall in the supervised category. We consider in this paper *linear* and *supervised* techniques. Within that category, two families of methods stand out: Given a dataset of pairs of vectors and labels  $\{(\mathbf{x}_i, y_i)\}_i$ , the goal of *Fisher Discriminant Analysis* (FDA) and variants is to learn a linear map  $\mathbf{P} : \mathbb{R}^d \rightarrow \mathbb{R}^p$ ,  $p \ll d$ , such that the embeddings of these points  $\mathbf{P}\mathbf{x}_i$  can be easily discriminated using linear classifiers. *Mahalanobis metric learning* (MML) methods proceed in the same way, except that the quality of the embedding  $\mathbf{P}$  is judged by the ability of the  $k$ -nearest neighbor algorithm to obtain good classification accuracy. Although both FDA and MML learn a map  $\mathbf{P}$ , their algorithms are shaped by the classifiers they use in their final phase.

FDA[3, §4.3] attempts to maximize w.r.t.  $\mathbf{P}$  the sum of *all* distances  $\|\mathbf{P}\mathbf{x}_i - \mathbf{P}\mathbf{x}_{j'}\|$  between pairs of samples from different classes  $c, c'$  while minimizing the sum of *all* distances  $\|\mathbf{P}\mathbf{x}_i - \mathbf{P}\mathbf{x}_j\|$  between pairs of samples within the same class  $c$ . Because of this, it is well documented that the performance of FDA degrades when class distributions are multimodal. Several variants have been proposed to

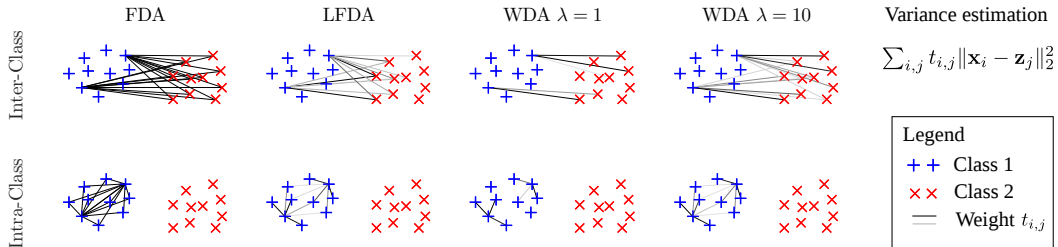


Figure 1: Weights used for inter/intra class variances for FDA, Local FDA and WDA for different regularizations  $\lambda$ . Only weights for two samples from class 1 are shown. The color of the link darkens as the weight grows. FDA computes a global variance with uniform weight on all pairwise distances, whereas LFDA focuses only on samples that lie close to each other. WDA relies on an optimal transport matrix  $\mathbf{T}$  that matches all points in one class to all other points in another class. WDA has both a global (due to matching constraints) and local (due to transportation cost minimization) outlook on the problem, with a tradeoff controlled by the regularization strength  $\lambda$ .

tackle this problem [3, §12.4]. For instance, a localized version of FDA was proposed in [4], which boils down to discarding in the computation all pairs of points that are not neighbors.

On the other hand, despite the fact that Mahalanobis metric learning (MML) was originally designed to operate with a  $k$ -nearest neighbor classifier, the first techniques that were proposed in that field [5] used a *global* criterion, based on all pairs of points. Later on, variations that focused instead exclusively on *close-range* interactions, such as LMNN [6], were shown to be far more efficient.

Supervised dimensionality approaches stemming either from FDA or MML consider thus *either* global *or* local interactions between points, but not both at the same time. We introduce in this work a novel approach that incorporates *both* global and local constraints. WDA can achieve this blend through the mathematics of optimal transport, which enforce that global and local aspects are taken into account when computing the weight  $t_{ij}$  that regulates the influence of the distance  $\|\mathbf{P}\mathbf{x}_i - \mathbf{P}\mathbf{x}_j\|$ . Indeed, such weights are decided by (i) making sure that *all* points in one set are matched to *all* points in the other set (global constraint); (ii) making sure that points are only matched on average to nearby points, thanks to the optimality of the transportation matrix (local cost). Our method has the added flexibility that it can interpolate between an exclusively global viewpoint (identical, in that case, to FDA), to a more local viewpoint with a global matching constraint (different, in that sense, to that of purely local tools such as LMNN or Local-FDA). In mathematical terms, we adopt the ratio formulation of FDA to maximize the ratio of the regularized Wasserstein distances between inter class populations and between the intra-class population with itself, when these points are considered *in their projected space*:

$$\max_{\mathbf{P} \in \Delta} \frac{\sum_{c,c' > c} W_\lambda(\mathbf{P}\mathbf{X}^c, \mathbf{P}\mathbf{X}^{c'})}{\sum_c W_\lambda(\mathbf{P}\mathbf{X}^c, \mathbf{P}\mathbf{X}^c)} \quad (1)$$

where  $\Delta = \{\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_p] \mid \mathbf{p}_i \in \mathbb{R}^d, \|\mathbf{p}_i\|_2 = 1 \text{ and } \mathbf{p}_i^\top \mathbf{p}_j = 0 \text{ for } i \neq j\}$  is the Stiefel manifold [7], the set of orthogonal  $d \times p$  matrices;  $\mathbf{P}\mathbf{X}^c$  is the matrix of projected samples from class  $c$  and  $W_\lambda$  is the regularized Wasserstein distance proposed in [8]. We recover FDA when the entropic regularization strength is very large (small  $\lambda$ ). When, on the contrary, that regularization is weak, our approach tries to split two class distributions by, using loose language, making sure that their optimal matching distance is as large as possible. In that process, each point in one class is paired with another point in another class, and only that distance is maximized. Figure 1 illustrates how inter and intra-class distances are computed, and the difference in how data points interact locally when comparing Wasserstein discriminant analysis (WDA), a global approach such as FDA, and a purely local one such as Local-FDA [4].

Another strong feature brought by regularized Wasserstein distances is that relations between samples (as given by the optimal transport matrix  $\mathbf{T}$ ) are estimated in the projected space. This is an important difference compared to all previous local approaches which estimate local relations in the original space and make the hypothesis that these relations are unchanged after projection. Estimating local relations in the projected space leads to more robust results in the presence of noisy dimensions, as illustrated in Figure 2, on a simulated example containing 8 out of 10 features with

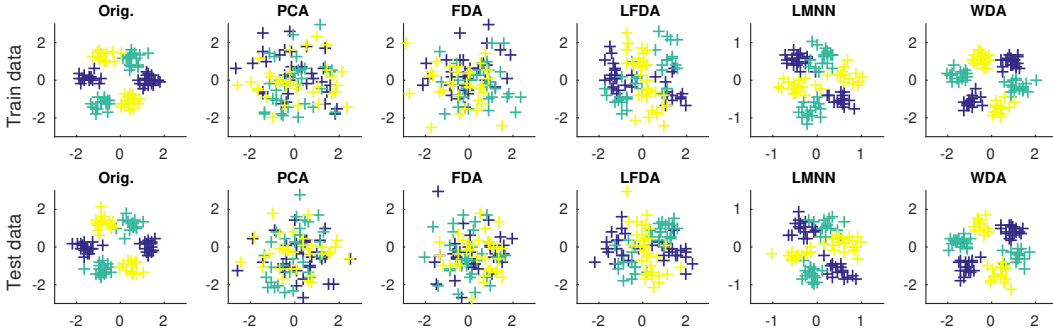


Figure 2: Illustration of subspace learning methods on a nonlinearly separable 3-class toy example of dimension  $d = 10$  with 2 discriminant features (shown on the left) and 8 Gaussian noise features. The projection onto  $p = 2$  for training (up) and test data (down) are reported for several subspace estimation methods.

only Gaussian noise. In this case, the 2D space estimated by our method is more discriminant than the local methods that rely on a noisy neighborhood estimation.

In what follows, we provide first background on regularized Wasserstein distances, present our Wasserstein-based discriminant analysis framework in §3 and discuss the optimization problem we tackle, which we solve using automatic differentiation. Numerical experiments that show the relevance of our approach are provided in §4.

## 2 Background on Wasserstein distances

Wasserstein distances, also known as earth mover distances, define a geometry over the space of probability measures using principles from optimal transport theory [9]. Recent computational advances [8, 10] have made it scale to dimensions relevant to machine learning applications.

**Notations and Definitions** Let  $\mu = \frac{1}{n} \sum_i \delta_{\mathbf{x}_i}$ ,  $\nu = \frac{1}{m} \sum_i \delta_{\mathbf{z}_i}$  be two empirical measures with locations in  $\mathbb{R}^d$  stored in the matrices  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$  and  $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_m]$ . The squared Euclidean distance between samples from  $\mu$  and  $\nu$  is defined as  $\mathbf{M}_{\mathbf{X}, \mathbf{Z}} := [\|\mathbf{x}_i - \mathbf{z}_j\|_2^2]_{ij} \in \mathbb{R}^{n \times m}$ . Let  $U_{nm}$  be the polytope of  $n \times m$  nonnegative matrices such that their row and column marginals are equal to  $\mathbf{1}_n/n$  and  $\mathbf{1}_m/m$  respectively. Writing  $\mathbf{1}_n$  for the  $n$ -dimensional vector of ones,

$$U_{nm} := \{\mathbf{T} \in \mathbb{R}_+^{n \times m} : \mathbf{T}\mathbf{1}_m = \mathbf{1}_n/n, \mathbf{T}^T\mathbf{1}_n = \mathbf{1}_m/m\}.$$

**Regularized Wasserstein distance** Let  $\langle A, B \rangle := \text{tr}(A^T B)$  be the Frobenius dot-product of matrices. For  $\lambda \geq 0$ , the regularized Wasserstein distance we adopt in this paper between  $\mu$  and  $\nu$  is:

$$W_\lambda(\mu, \nu) := W_\lambda(\mathbf{X}, \mathbf{Z}) := \langle \mathbf{T}_\lambda, \mathbf{M}_{\mathbf{X}, \mathbf{Z}} \rangle, \quad (2)$$

where  $\mathbf{T}_\lambda$  is the solution of an entropy-smoothed optimal transport problem,

$$\mathbf{T}_\lambda := \underset{\mathbf{T} \in U_{nm}}{\text{argmin}} \lambda \langle \mathbf{T}, \mathbf{M}_{\mathbf{X}, \mathbf{Z}} \rangle - \Omega(\mathbf{T}), \quad (3)$$

where  $\Omega(\mathbf{T})$  is the entropy of  $\mathbf{T}$  seen as a discrete joint probability distribution, namely  $\Omega(\mathbf{T}) := -\sum_{ij} t_{ij} \log(t_{ij})$ . Note that problem (3) can be solved very efficiently using the Sinkhorn fixed point iterations [8]. The solution of the optimization problem can be expressed as:

$$\mathbf{T} = \text{diag}(\mathbf{u})e^{-\lambda \mathbf{M}} \text{diag}(\mathbf{v}) = \mathbf{u}\mathbf{1}_m^T \circ e^{-\lambda \mathbf{M}} \circ \mathbf{1}_n \mathbf{v}^T, \quad (4)$$

where  $\circ$  stands for elementwise multiplication of matrices and the exponential is also applied elementwise. The Sinkhorn iterations consist in updating left/right scaling vectors  $\mathbf{u}^k$  and  $\mathbf{v}^k$  of the matrix  $\mathbf{K} = e^{-\lambda \mathbf{M}}$ . These updates take the following form for iteration  $k$ :

$$\mathbf{v}^k = \frac{\mathbf{1}_m/m}{\mathbf{K}^T \mathbf{u}^{k-1}}, \quad \mathbf{u}^k = \frac{\mathbf{1}_n/n}{\mathbf{K} \mathbf{v}^k} \quad (5)$$

with an initialization which will be fixed to  $\mathbf{u}^0 = \mathbf{1}_n$ . Because it only involves matrix products, the Sinkhorn algorithm can be streamed efficiently on parallel architectures such as GPGPUs.

**Optimal transport for machine learning** The geometry of the space of probability measures endowed with the Wasserstein metric has been recently considered in applied settings, notably to compute barycenters [11, 10], or generalizations of PCA [12]. Wasserstein distances have also been considered to carry out semi-supervised learning [13], domain adaptation [14], or to define new loss functions for multiclass classifiers [15]. A recent application related to ours is that of identifying univariate projections of high-dimensional probability distributions [16]. Contrary to our approach, this reference builds upon the closed form formulation of 1D-Wasserstein distances and was not designed to find subspaces of dimension  $p > 1$ .

### 3 Wasserstein Discriminant Analysis

In this section we discuss the optimization problem (1) and its relation to FDA. Then we propose an efficient approach for computing the gradient of the objective function.

**Optimization problem** In the remaining, to simplify notations, we define a separate empirical measure for each of the  $C$  classes: The samples of each class  $c$  are stored in matrices  $\mathbf{X}^c$ ; the number of samples from class  $c$  is  $n_c$ .

Using the definition (2) of the regularized Wasserstein distance, we can write the Wasserstein Discriminant Analysis optimization problem as

$$\begin{aligned} \max_{\mathbf{P} \in \Delta} \quad & \left\{ J(\mathbf{P}, \mathbf{T}(\mathbf{P})) = \frac{\sum_{c,c' > c} \langle \mathbf{P}^T \mathbf{P}, \mathbf{C}^{c,c'} \rangle}{\sum_c \langle \mathbf{P}^T \mathbf{P}, \mathbf{C}^{c,c} \rangle} \right\} \quad (6) \\ \text{s.t. } \mathbf{C}^{c,c'} = & \sum_{i,j} T_{i,j}^{c,c'} (\mathbf{x}_i^c - \mathbf{x}_j^{c'}) (\mathbf{x}_i^c - \mathbf{x}_j^{c'})^T, \quad \forall c, c' \\ \text{and } \mathbf{T}^{c,c'} = & \operatorname{argmin}_{\mathbf{T} \in U_{n_c n_{c'}}} \lambda \langle \mathbf{T}, \mathbf{M}_{\mathbf{P}\mathbf{X}^c, \mathbf{P}\mathbf{X}^{c'}} \rangle - \Omega(\mathbf{T}), \end{aligned}$$

which can be reformulated as

$$\max_{\mathbf{P} \in \Delta} \quad J(\mathbf{P}, \mathbf{T}(\mathbf{P})) \quad (7)$$

$$\text{s.t. } \mathbf{T}(\mathbf{P}) = \operatorname{argmin}_{\mathbf{T} \in U_{n_c n_{c'}}} E(\mathbf{T}, \mathbf{P}) \quad (8)$$

where  $\mathbf{T} = \{\mathbf{T}_{c,c'}\}_{c,c'}$  contains all the transport matrices between classes. The objective function  $J$  and the inner problem function  $E$  defined as

$$J(\mathbf{P}, \mathbf{T}(\mathbf{P})) = \frac{\langle \mathbf{P}^T \mathbf{P}, \mathbf{C}_b \rangle}{\langle \mathbf{P}^T \mathbf{P}, \mathbf{C}_w \rangle}, \quad E(\mathbf{T}) = \sum_{c,c' > c'} \lambda \langle \mathbf{T}^{c,c'}, \mathbf{M}_{\mathbf{P}\mathbf{X}^c, \mathbf{P}\mathbf{X}^{c'}} \rangle - \Omega(\mathbf{T}^{c,c'}) \quad (9)$$

where  $\mathbf{C}_b = \sum_{c,c' > c} \mathbf{C}^{c,c'}$  and  $\mathbf{C}_w = \sum_c \mathbf{C}^{c,c}$  are the between and within cross-covariance matrices that depend on  $\mathbf{T}(\mathbf{P})$ . Optimization problem (7)-(8) is a classic bilevel optimization problem and it has a particular form that allows its resolution using gradient descent [17]. Indeed since  $T(\mathbf{P})$  is smooth and optimization problem (8) is strictly convex one can compute the gradient directly *w.r.t.*  $\mathbf{P}$  using the chain rule as follows

$$\nabla_{\mathbf{P}} J(\mathbf{P}, \mathbf{T}(\mathbf{P})) = \frac{\partial J(\mathbf{P}, \mathbf{T})}{\partial \mathbf{P}} + \sum_{c,c' > c} \frac{\partial J(\mathbf{P}, \mathbf{T})}{\partial \mathbf{T}^{c,c'}} \frac{\partial \mathbf{T}^{c,c'}}{\partial \mathbf{P}} \quad (10)$$

The first term in gradient (10) suppose that  $\mathbf{T}$  is constant and can be computed (Eq. 94-95 [18]) as

$$\frac{\partial J(\mathbf{P}, \mathbf{T})}{\partial \mathbf{P}} = \mathbf{P} \left( \frac{2}{\sigma_w^2} \mathbf{C}_b - \frac{2\sigma_b^2}{\sigma_w^4} \mathbf{C}_w \right) \quad (11)$$

with  $\sigma_w^2 = \langle \mathbf{P}^T \mathbf{P}, \mathbf{C}_w \rangle$  and  $\sigma_b^2 = \langle \mathbf{P}^T \mathbf{P}, \mathbf{C}_b \rangle$ . In order to compute the second term in (10), we will separate the cases when  $c = c'$  and  $c \neq c'$  since it corresponds to their position in the fraction. Their partial derivative is obtained directly from the scalar product and is a weighted vectorization of the transport cost matrix

$$\frac{\partial J(\mathbf{P}, \mathbf{T})}{\partial \mathbf{T}^{c,c' \neq c}} = \operatorname{vec} \left( \frac{1}{\sigma_b^2} \mathbf{M}_{\mathbf{P}\mathbf{X}^c, \mathbf{P}\mathbf{X}^{c'}} \right), \quad \frac{\partial J(\mathbf{P}, \mathbf{T})}{\partial \mathbf{T}^{c,c}} = -\operatorname{vec} \left( \frac{\sigma_w^2}{\sigma_b^4} \mathbf{M}_{\mathbf{P}\mathbf{X}^c, \mathbf{P}\mathbf{X}^c} \right) \quad (12)$$

We will see in the remaining that the main difficulty stands in computing the Jacobian  $\frac{\partial \mathbf{T}^{c,c'}}{\partial \mathbf{P}}$  since the optimal transport matrix is not available as a closed form. In the sequel, we solve this problem by means of an automatic differentiation approach wrapped around the Sinkhorn fixed point iteration algorithm.

Once the gradient is computed, we can use classical manifold optimization tools such as projected gradient [19] or a trust region algorithm as implemented in Manopt [20]. The latter toolbox includes tools to optimize over the Stiefel manifold, notably automatic conversions from Euclidean to Riemannian gradients.

**Relation to Fisher Discriminant Analysis** We consider the limit behavior of WDA as  $\lambda$  approaches 0. In this case, we can see from Eq. (3) that the matrix  $\mathbf{T}$  does not depend on the data. The solution  $\mathbf{T}$  for each Wasserstein distance is the matrix that maximizes entropy, namely the uniform probability distribution  $\mathbf{T} = \frac{1}{nm} \mathbf{1}_{n,m}$ . The cross-covariance matrices become thus

$$\mathbf{C}^{c,c'} = \frac{1}{n_c n_{c'}} \sum_{i,j} (\mathbf{x}_i^c - \mathbf{x}_j^{c'}) (\mathbf{x}_i^c - \mathbf{x}_j^{c'})^T$$

and the matrices  $\mathbf{C}_w$  and  $\mathbf{C}_b$  correspond then to intra- and inter-class covariances as used in FDA. Since these matrices do not depend on  $\mathbf{P}$ , the optimization problem (1) boils down to the usual Rayleigh quotient which can be solved using a generalized eigendecomposition of  $\mathbf{C}_w^{-1} \mathbf{C}_b$ . When  $\lambda > 0$ , the smoothed optimal transport promotes instead cross-covariance matrices that tend to favor local relations as illustrated in Figure 1.

Also note that the regularization parameter ( $\lambda < \infty$ ) is required in the current formulation (1). Indeed, the classical Wasserstein distance of a discrete distribution with itself is 0. The entropic regularization ensures that the mass of each sample is split among neighbors hence promoting a local variance estimation and avoiding a division by 0 in (1). If one wants to use the non-regularized Wasserstein distance, then the intra-class distance should be computed between two non-overlapping splits of the examples of each class which would also lead to non-zero distance.

**Automatic Differentiation** A possible way to compute the Jacobian  $\partial \mathbf{T}^{c,c'} / \partial \mathbf{P}$  is to use the implicit function theorem as proposed in [21] for hyperparameter estimation. Although we have tried it, as detailed in supplementary material, this computational approach fails to scale because it requires inverting a very large matrix, and also assumes that the exact optimal transport  $\mathbf{T}$  can be obtained, despite the fact that one only gets it after a finite number of Sinkhorn iterations. Following the gist of [22] (which do not differentiate Sinkhorn iterations but a more complex fixed point iteration designed to compute Wasserstein barycenters), we propose in this section to differentiate the transportation matrices obtained after running exactly  $L$  Sinkhorn iterations, with a predefined  $L$ . Writing  $\mathbf{T}^k(\mathbf{P})$ , for the solution obtained after  $k$  iterations as a function of  $\mathbf{P}$  for a given  $c, c'$  pair,

$$\mathbf{T}^k(\mathbf{P}) = \text{diag}(\mathbf{u}^k) e^{-\lambda \mathbf{M}} \text{diag}(\mathbf{v}^k)$$

where  $\mathbf{M}$  is the distance matrix induced by  $\mathbf{P}$ .  $\mathbf{T}^L(\mathbf{P})$  can then be directly differentiated:

$$\frac{\partial \mathbf{T}^k}{\partial \mathbf{P}} = \frac{\partial [\mathbf{u}^k \mathbf{1}_m^T]}{\partial \mathbf{P}} \circ e^{-\lambda \mathbf{M}} \circ \mathbf{1}_n \mathbf{v}^k{}^T + \mathbf{u}^k \mathbf{1}_m^T \circ \frac{\partial e^{-\lambda \mathbf{M}}}{\partial \mathbf{P}} \circ \mathbf{1}_n \mathbf{v}^k{}^T + \mathbf{u}^k \mathbf{1}_m^T \circ e^{-\lambda \mathbf{M}} \circ \frac{\partial [\mathbf{1}_n \mathbf{v}^k{}^T]}{\partial \mathbf{P}} \quad (13)$$

Note that the recursion occurs as  $\mathbf{u}^k$  depends on  $\mathbf{v}^k$  whose is also related to  $\mathbf{u}^{k-1}$ . The Jacobians that we need can then be obtained from Equation (5). For instance, the gradient of one component of  $\mathbf{u}^k \mathbf{1}_m^T$  at the  $j$ -th line is

$$\frac{\partial \mathbf{u}_j^k}{\partial \mathbf{P}} = -\frac{1/n}{[\mathbf{K} \mathbf{v}^k]_j^2} \left( \sum_i \frac{\partial \mathbf{K}_{j,i}}{\partial \mathbf{P}} \mathbf{v}_i^k + \sum_i \mathbf{K}_{j,i} \frac{\partial \mathbf{v}_i^k}{\partial \mathbf{P}} \right),$$

while for  $\mathbf{v}^k$ , we have

$$\frac{\partial \mathbf{v}_j^k}{\partial \mathbf{P}} = -\frac{1/m}{[\mathbf{K}^T \mathbf{u}^{k-1}]_j^2} \left( \sum_i \frac{\partial \mathbf{K}_{i,j}}{\partial \mathbf{P}} \mathbf{u}_i^{k-1} + \sum_i \mathbf{K}_{i,j} \frac{\partial \mathbf{u}_i^{k-1}}{\partial \mathbf{P}} \right),$$

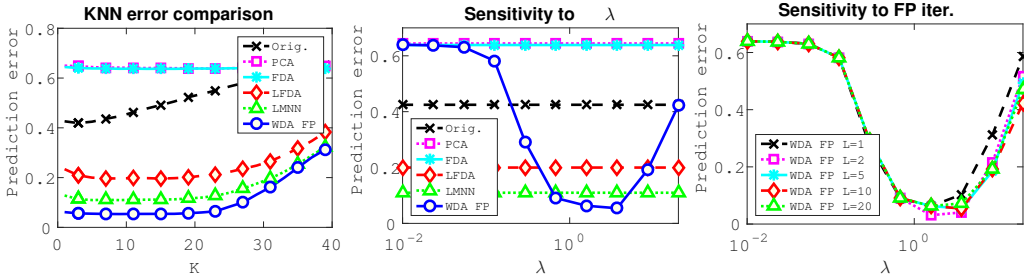


Figure 3: Prediction error on the simulated dataset with projection dimension  $p = 2$ . (left) Error for a varying  $K$  in the KNN classifier (middle) Evolution of WDA performance for different regularization parameter values  $\lambda$  (right) comparison of WDA performances as a function of the number of fixed point iterations.

and finally

$$\frac{\partial \mathbf{K}_{i,j}}{\partial \mathbf{P}} = -2K_{i,j} \mathbf{P}(\mathbf{x}_i - \mathbf{x}'_j)(\mathbf{x}_i - \mathbf{x}'_j)^T.$$

The Jacobian  $\frac{\partial \mathbf{T}^k}{\partial \mathbf{P}}$  can be thus obtained by keeping track of all the Jacobians at each iteration and then by successive applying those equations. This approach is far cheaper than the implicit function theorem approach. Indeed, in this case, the computation of  $\frac{\partial \mathbf{T}}{\partial \mathbf{P}}$  is dominated by the complexity of computing  $\frac{\partial \mathbf{K}}{\partial \mathbf{P}}$  whose costs for one iteration is  $O(pn^2d^2)$  for  $n = m$ . The complexity is then linear in  $L$  and quadratic in  $n$ .

## 4 Numerical experiments

In this section we illustrate WDA on three datasets. First, we evaluate our approach on a simple simulated dataset with a 2-dimensional discriminant subspace that is known a priori. Then we apply WDA on MNIST and Caltech datasets. Note that in the spirit of reproducible research all the Matlab code will be made available to the community on Github and mloss.org upon publication.

**Practical implementation** In order to make the method less sensitive to the dimension and scaling of the data, we propose to use a pre-computed adaptive regularization parameter  $\lambda_{c,c'}$  for each Wasserstein distances in (1). In practice we initialize  $\mathbf{P}$  with the PCA projection, and we use it to find the following  $\lambda_{c,c'} = \lambda \left( \frac{1}{n_c n_{c'}} \sum_{i,j} \|\mathbf{P}x_i^c - \mathbf{P}x_j^{c'}\|^2 \right)^{-1}$  between class  $c$  and  $c'$ . These values are computed *a priori* and fixed in the remaining iterations, but they will promote a similar regularization between inter and intra-class distances.

**Simulated dataset** This dataset has been designed to evaluate the ability of a subspace method to estimate a linear subspace when the classes are non-linearly separable. It is a 3-class problem in dimension  $d = 10$  where only the first two dimensions are discriminant and the remaining 8 contain Gaussian noise with the same variance as the discriminant dimensions. In the 2 discriminant features each class contains two modes that have centers positioned so that all the classes have the same mean at  $[0, 0]$ . An example of samples in the discriminant dimensions is shown in the left part of Figure 2.

First, we depict in Figure 2, the samples projected in several subspaces with different approaches. We can see that for this dataset WDA seems to estimate the subspace with the best generalization property. To verify this trend, we compute the prediction errors of all the linear subspace methods with a K-Nearest-Neighbors classifier (KNN) for  $n = 100$  training examples and  $n_t = 5000$  test examples in Figure 3. In this simulation, all prediction errors are averaged over 20 data generations and the neighbors parameters of LMNN and LFDA have been selected empirically to maximize performances ( $K = 5$  for LMNN and  $K = 1$  for LFDA). We can see in the left part of the figure that WDA and to a lesser extent LMNN can estimate a relevant subspace that is robust to the choice of  $K$ . LFDA clearly outperforms PCA, Orig. and LDA since it can handle non-linearity but fails to attain the performances of LMNN and WDA. In the center plot, we illustrate the sensitivity of

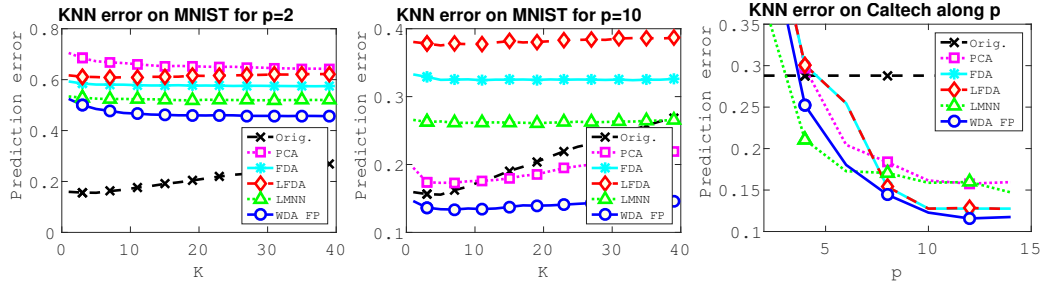


Figure 4: (left) Averaged prediction error on MNIST with projection dimension  $p = 2$  and (middle)  $p = 10$ . (right) Averaged prediction error on the Caltech-256 dataset as a function of  $p$ .

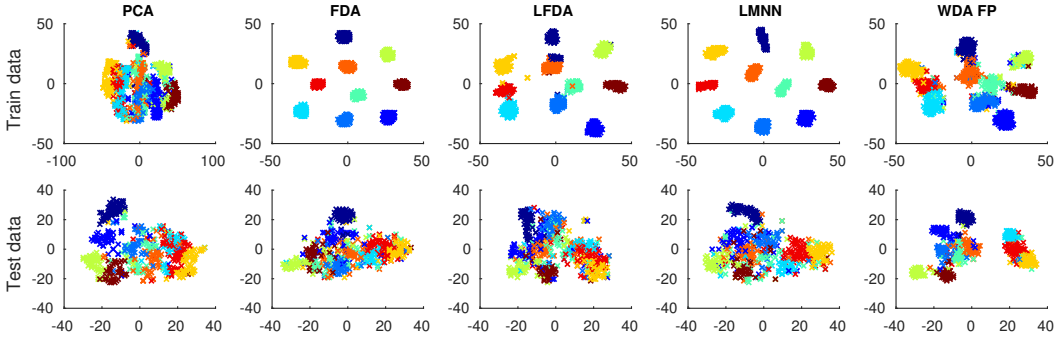


Figure 5: 2D tSNE of the MNIST samples projected on  $p = 10$  for different approaches. (up) training set (down) test set.

WDA *w.r.t.* the regularization parameter  $\lambda$ . WDA returns the best performance on almost a full order of magnitude which suggests that a coarse validation can be performed in practice. The right part of Figure 3 shows the performance of the WDA with fixed point gradient (WDA FP) for different values of the number of inner iterations  $L$ . We can see that even if this parameter leads to different performances on large values of  $\lambda$  it still leads to the best overall performances even for small  $L$ . The variability between the curves comes from the non-convexity of the objective function, since each descent direction might lead to a different stationary point.

**MNIST dataset.** Our aim on this dataset is to measure how robust our approach is when only few training samples are available in high-dimension. To this end, we draw  $n = 1000$  samples for training and report the KNN prediction error as a function of  $k$  for the different subspace methods when projecting onto  $p = 2$  and  $p = 10$  dimensions (resp. left and middle plots of Figure 4). In order to leverage the impact of a random choice of the initial distribution, the reported scores are averages of 20 realizations of the same experiment. We also limit the analysis to  $L = 10$  as the number of fixed point iterations. As expected, when  $p = 2$  there is an important loss of information which leads to large classification errors compared to a KNN in the original space (Orig.). Nevertheless, WDA seems to be the method that finds the best 2D subspace. When  $p = 10$ , WDA finds a better subspace than the original space which suggests that all the discriminant information available in the training dataset has been extracted efficiently. Conversely, LFDA and LMNN struggle in finding a relevant subspace in this configuration. We believe that this is due to the small number of training samples and the fact that those approaches look at local relationship whereas the Wasserstein distance is a global measure that can be regularized to better avoid overfitting and to better capture local informations. In addition to better prediction performance, we want to emphasize that in this configuration WDA leads to a dramatic compression of the data from 784 to 10 features while keeping the discriminative information.

To gain a better understanding of the corresponding embedding, we project the data from the 10-dimensional space to a 2-dimensional one using tSNE, that is a classic nonlinear projection method used for visualizing high dimensional samples for instance in deep architectures [23]. In order to



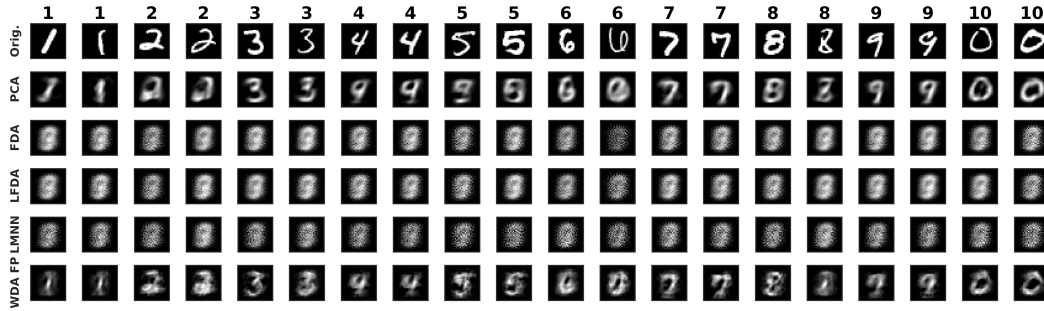


Figure 6: Image reconstruction after projection in the subspaces for test samples. The title on top of each column corresponds to the class of the sample.

be able to compare the embeddings, we first apply tSNE on samples in the PCA space and use it as an initialization for all the other methods. The resulting 2D projections for both training and testing samples are shown in Figure 5. Here we can clearly see the overfitting of FDA, LFDA and LMNN that separate accurately the training samples but fail to separate the test samples.

We finally compare in Figure 6 the different reconstructions induced by the adjoint projection operator. One can see that contrary to the other discriminative methods, our method preserves a coherent spatial information related to the class, which illustrates again the generalization capability of WDA. This means that the estimated subspace is both generative as for PCA and discriminative thanks to the use of a proper divergence between discrete distributions.

**Caltech dataset** In this experiment, we use a subset described in [24] from the Caltech-256 image collection [25]. The dataset uses features that are the output of the DeCAF deep learning architecture [24]. More precisely, they are extracted as the sparse activation of the neurons from the 6th fully connected layer of a convolutional network trained on imageNet and then fine-tuned for the considered visual recognition task. As such, they form vectors of 4096 dimensions. In this setting, 500 images are considered for training, and the remaining portion of the dataset for testing (623 images). There are 9 different classes in this dataset. We examine in this experiment how the proposed dimensionality reduction performs when changing the subspace dimensionality (right plot in Figure 4). For this problem, the regularization parameter  $\lambda$  of WDA was empirically set to  $10^{-2}$  and we use early stopping at 10 iterations for WDA. The  $K$  from KNN was set to 3 which is a common standard setting for this classifier. The reported results are averaged over 10 realizations of the same experiment. When  $p \geq 5$ , WDA already finds a subspace which gathers all the discriminative information from the original space. In this experiment LMNN reaches a better subspace for small  $p$  values but WDA is the best performing method for  $p \geq 6$ . Those results highlight the potential interest for using WDA as linear dimensionality reduction layers in neural-nets architecture.

## 5 Conclusion

This work presents the Wasserstein Discriminant Analysis, a new and original linear discriminant subspace estimation method. Based on the framework of Wasserstein distances, which measure a global similarity between empirical distributions, WDA operates by separating distributions of different classes in the subspace, while maintaining a coherent structure at a class level. To this extent, the use of regularization in the Wasserstein formulation allows to effectively bridge a gap between a global coherency and the local structure of the class manifold. This comes at a cost of a difficult optimization of a bilevel program, for which we showed that a naive computation of the gradient is achievable but computationally too costly in real settings. We proposed instead an efficient method that relies on differentiating the map obtained after a finite number of iterations of the Sinkhorn algorithm. Numerical experiments show that the method performs well on a variety of features, including those obtained with a deep neural architecture. Future work will consider stochastic versions of the same approach in order to enhance further the ability of the method to handle large volume of high-dimensional data.

## References

- [1] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: a comparative review. *Journal of Machine Learning Research*, 10:66–71, 2009.
- [2] Christopher JC Burges. *Dimension reduction: A guided tour*. Now Publishers, 2010.
- [3] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Springer series in statistics Springer, Berlin, 2001.
- [4] Masashi Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *The Journal of Machine Learning Research*, 8:1027–1061, 2007.
- [5] Eric P Xing, Andrew Y Ng, Michael I Jordan, and Stuart Russell. Distance metric learning with application to clustering with side-information. *Advances in neural information processing systems*, 15:505–512, 2003.
- [6] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.
- [7] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.
- [8] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *NIPS*, pages 2292–2300, 2013.
- [9] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.
- [10] Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.
- [11] M. Cuturi and A. Doucet. Fast computation of wasserstein barycenters. In *ICML*, 2014.
- [12] V. Seguy and M. Cuturi. Principal geodesic analysis for probability measures under the optimal transport metric. In *NIPS*, pages 3294–3302. 2015.
- [13] Justin Solomon, Raif Rustamov, Guibas Leonidas, and Adrian Butscher. Wasserstein propagation for semi-supervised learning. In *ICML*, pages 306–314, 2014.
- [14] N. Courty, R. Flamary, and D. Tuia. Domain adaptation with regularized optimal transport. In *ECML PKDD*, September 2014.
- [15] C. Frogner, C. Zhang, H. Mobahi, M. Araya, and T. Poggio. Learning with a wasserstein loss. In *NIPS*, pages 2044–2052. 2015.
- [16] J. Mueller and T. Jaakkola. Principal differences analysis: Interpretable characterization of differences between distributions. In *NIPS*, pages 1693–1701. 2015.
- [17] Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, 153(1):235–256, 2007.
- [18] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7:15, 2008.
- [19] M Schmidt. Minconf-projection methods for optimization with simple constraints in matlab, 2008.
- [20] Nicolas Boumal, Bamdev Mishra, P-A Absil, and Rodolphe Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459, 2014.
- [21] Yoshua Bengio. Gradient-based optimization of hyperparameters. *Neural computation*, 12(8):1889–1900, 2000.
- [22] Nicolas Bonneel, Gabriel Peyré, and Marco Cuturi. Wasserstein barycentric coordinates: Histogram regression using optimal transport. *ACM Transactions on Graphics*, 35(4), 2016.
- [23] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.
- [24] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: a deep convolutional activation feature for generic visual recognition. In *Proceedings of The 31st International Conference on Machine Learning*, pages 647–655, 2014.
- [25] G. Griffin, A. Holub, and P. Perona. Caltech-256 Object Category Dataset. Technical Report CNS-TR-2007-001, California Institute of Technology, 2007.