Fast Dictionary Learning with a Smoothed Wasserstein Loss

Antoine Rolet
antoine.rolet@iip.ist.i.kyoto-u.ac.jp
Graduate School of Informatics,
Kyoto University

Marco Cuturi
mcuturi@i.kyoto-u.ac.jp
Graduate School of Informatics,
Kyoto University

Gabriel Peyré
gabriel.peyre@ceremade.dauphine.fr
CNRS, CEREMADE,
Université Paris Dauphine

Abstract

We consider in this paper the dictionary learning problem when the observations are normalized histograms of features. This problem can be tackled using non-negative matrix factorization approaches, using typically Euclidean or Kullback-Leibler fitting errors. Because these fitting errors are separable and treat each feature on equal footing, they are blind to any similarity the features may share. We assume in this work that we have prior knowledge on these features. To leverage this side-information, we propose to use the Wasserstein (*a.k.a.* earth mover's or optimal transport) distance as the fitting error between each original point and its reconstruction, and we propose scalable algorithms to to so. Our methods build upon Fenchel duality and entropic regularization of Wasserstein distances, which improves not only speed but also computational stability. We apply these techniques on face images and text documents. We show in particular that we can learn dictionaries (topics) for bag-of-word representations of texts using words that may not have appeared in the original texts, or even words that come from a different language than that used in the texts.

1 Introduction

Consider a collection $X = (x_1, \dots, x_m)$ of m vectors of dimension n. Learning a dictionary for X can be stated informally as the goal of finding k dictionary elements $D = (d_1, \dots, d_k)$ of the same dimension n such that each X_i can be reconstructed using such a dictionary, namely such that there exists a matrix of mixture weights $\Lambda = (\lambda_1, \dots, \lambda_m)$ such that $X \simeq D\Lambda$.

When all elements of X are non-negative, and if it is desirable that all elements of D and Λ are nonnegative too, this problem becomes that of Non-negative Matrix Factorization (NMF) [Paatero and Tapper, 1994]. Lee and Seung [2001] proposed two algorithms for NMF, with the aim of solving problems of the form:

$$\min_{D \in \mathbb{R}_+^{n \times k}, \Lambda \in \mathbb{R}_+^{k \times m}} \sum_{i=1}^{m} \ell(x_i, D\lambda_i) + R(D, \Lambda),$$

where ℓ is either the Kullback-Leibler divergence or the squared Euclidean distance and R a regularizer. Dictionary learning and NMF have been used for various machine learning and signal processing tasks, including (but not limited to) semantic analysis [Hofmann, 1999, Lee and Seung, 1999], matrix completion [Zhang et al., 2006] and sound denoising [Schmidt et al., 2007].

Our goal in this paper is to generalize these approaches using a regularized Wasserstein (*a.k.a* optimal transport [Villani, 2009] or earth mover's [Rubner et al., 1998]) distance as the data fitting term ℓ. Such distances can leverage additional knowledge on the space of features using a metric between features called the ground metric. Since the seminal work of Rubner et al. [1998], several hundred papers have successfully used EMD in applications. Some recent works have for instance illustrated its relevance for text classification [Kusner et al., 2015], image segmentation [Rabin and Papadakis, 2015] and shape interpolation [Solomon et al., 2015].

We motivate the idea of using a Wasserstein fitting error with a toy example described in Figure 1. In this example we try to learn dictionaries for histogram representations of i.i.d. samples from mixtures of Gaussians. We consider $n = 100$ distributions $\rho_1, \dots, \rho_n$, each of which is a mixture of three univariate Gaussians of unit variance, with centers picked independently using $\mathcal{N}(-6, 2)$, $\mathcal{N}(0, 2)$ and $\mathcal{N}(6, 2)$ respectively. The relative weights of these Gaussians are

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51.

picked uniformly on $[0, 1]$ and subsequently normalized to sum to 1 for each distribution. We consider then a sample of m observations for each distribution ρ_i, and represent each sample as a histogram x_i of $n = 100$ bins regularly spaced on the segment $[-12, 12]$. Here the features are points on the quantization grid, and the ground metric is simply the Euclidean distance between these points. Wasserstein NMF recovers components which are centered around -6, 0 and 6 and resemble Gaussian pdfs. Because it is blind to the metric structure of $\mathbb{R}$, KL NMF fail to recover such intuitive components.

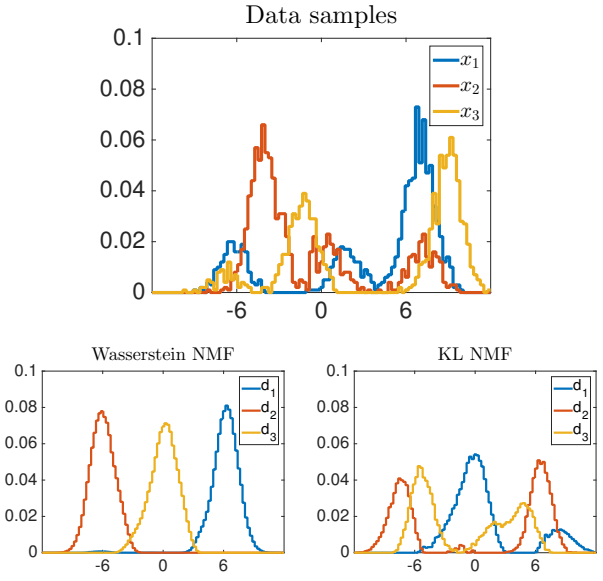

Figure 1: Dictionaries learned on mixtures of three randomly shifted Gaussians. Separable distances or divergences do not quantify this noise well because it is not additive in the space of histograms. Top: examples of data histograms. Bottom: dictionary learned with Wasserstein (left) and Kullback-Leibler (right) NMF.

Related Work Sandler and Lindenbaum [2009] were the first to consider NMF problems using a Wasserstein loss. They noticed that minimizing Wasserstein fitting errors requires solving an extremely costly linear program at each iteration of their block-coordinate iteration. Because of this, they settle instead for an approximation of the Wasserstein distance proposed by Shirdhonkar and Jacobs [2008]. However this approximation can only be used when the features are in $\mathbb{R}^d$, and its complexity is exponential in d, making it impractical when $d > 3$. Moreover the experimental approximation ratio for $d = 2$ in Shirdhonkar and Jacobs [2008] is rather loose (1.5) even with the best hyper-parameters. Zen et al. [2014] also proposed a semi-supervised method to learn D, Λ and a ground metric parameter. Their approach is to alternatively learn the ground metric as proposed previously in [Cuturi and Avis, 2014] and perform NMF by solving two very high dimensional linear programs. They apply their algorithm to histograms of small dimension ($n \leq 16$).

Our Contribution The algorithms we propose to solve dictionary learning and NMF problems with a Wasserstein loss scale to problems with far more observations and dimensions than previously considered in the literature [Sandler and Lindenbaum, 2009, Zen et al., 2014]. This is enabled by an entropic regularization of optimal transport [Cuturi, 2013] which results in faster and more stable computations. We introduce this regularization in Section 2, and follow in Section 3 with a detailed presentation of our algorithms for Wasserstein (nonnegative) matrix factorization of histogram matrices. In contrast to previously considered approaches, our approach can be applied with any ground metric. As with most dictionary learning problems, our objective is not convex but biconvex in the dictionary D and weights Λ and we use a block-coordinate descent approach. We show that each of these subproblems can be reduced to an optimization problem involving the Legendre-Fenchel conjugate of the objective, building upon recent work in Cuturi and Peyré [2016] that shows that the Legendre-Fenchel conjugate of the entropy regularized Wasserstein distance and its gradient can be obtained in closed form. We show in Section 4 that these fast algorithms are order of magnitudes faster than those proposed in Sandler and Lindenbaum [2009], whose experiments we replicate. Finally, we show that the features used to describe dictionary elements can be different from those present in the original histograms. We showcase this property to carry out cross-language semantic analysis: we learn topics in French using databases of English texts. A Matlab implementation of our methods and scripts to reproduce the experiment in the introduction are available at `http://arolet.github.io/wasserstein-dictionary-learning/`.

Notations If X is a matrix, X_i denotes its ith line, x_j its jth column and X_{ij} its element at the ith line and jth column. For $x, y \in \mathbb{R}^n$, $\langle x, y \rangle$ is the usual dot product between x and y. For $X, Y \in \mathbb{R}^{n \times m}$, $\langle X, Y \rangle \stackrel{\text{def.}}{=} \operatorname{tr}(X^T Y) = \sum_{i=1}^m \langle X_i, Y_i \rangle$ is the Frobenius dot-product between matrices X and Y. If A, B are two matrices of the same size, $A \odot B$ (*resp.* $\frac{A}{B}$) denotes the coordinate-wise product (*resp.* quotient) between A and B. Σ_n is the set of n-dimensional histograms: $\Sigma_n \stackrel{\text{def.}}{=} \left\{ q \in \mathbb{R}_+^n \,|\, \langle q, \mathbf{1} \rangle = 1 \right\}$. If A is a matrix, A^+ is its Moore-Penrose pseudoinverse. Exponentials and logarithms are applied element-wise to matrices and vectors.

If $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ is convex, its Legendre conjugate $f^* : x \in \mathbb{R}^n \mapsto \max_{y \in \Omega} \langle x, y \rangle - f(y)$ is convex. If $g : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ is concave, then $g_* = -(-g)^*$ is concave and is defined as $\forall x \in \mathbb{R}^n$, $g_*(x) = \min_{y \in \Omega} -\langle x, y \rangle - g(y)$.

2 Regularized Wasserstein Distances

We start this section by defining Wasserstein distances for histograms and then introduce their entropic regularization.

2.1 Definition of Wasserstein Distances

Let $p \in \Sigma_n$, $q \in \Sigma_s$. The polytope of transportation plans between p and q is defined as follows:

$$U(p,q) = \left\{ T \in \mathbb{R}_+^{n \times s} \text{ s.t. } \left| \begin{array}{c} T\mathbf{1} = p \\ T^T\mathbf{1} = q \end{array} \right. \right\}.$$

Let $M \in \mathbb{R}_+^{n \times s}$ be matrix of costs. The optimal transport cost between p and q with respect to M is

$$W(p,q) \stackrel{\text{def.}}{=} \min_{T \in U(p,q)} \langle M, T \rangle. \tag{1}$$

The optimization problem described above is a minimum cost network flow. Specialized algorithm can solve it with $O(((n+s)log(n+s))^2 + ns(n+s)\log(n+s))$[1] [Orlin, 1993]. In most applications, M is a pairwise distance matrix called the *ground metric*. Namely, there exists a metric space (Ω, d) and elements $y_1, \cdots, y_n$ and $z_1, \cdots, z_s$ in Ω such that $M_{ij} = \mathrm{d}(y_i, z_j)$. In that case W is a distance [Villani, 2009].

2.2 Entropic Regularization

Solving problems with Wasserstein distance fitting errors can require solving several costly optimal transport problems. As a minimum of affine functions, the Wasserstein distance itself is not a smooth function of its arguments [Cuturi and Doucet, 2014]. To avoid both of these issues, Cuturi [2013] proposed to smooth the optimal transport problem with an entropic term:

$$W_\gamma(p,q) \stackrel{\text{def.}}{=} \min_{T \in U(p,q)} \langle M, T \rangle - \gamma h(T), \tag{2}$$

where h is the (strictly concave) entropy function:

$$h(T) \stackrel{\text{def.}}{=} -\langle T, \log T \rangle. \tag{3}$$

The plan $T^\star$ solution of the problem in Equation (2) is unique and can be found by computing two vectors $u \in \mathbb{R}_+^n, v \in \mathbb{R}_+^s$ such that $\mathrm{diag}(u) K \,\mathrm{diag}(v) \in U(p,q)$, where $K = e^{-M/\gamma}$. The optimal solution is then $T^* = \mathrm{diag}(u) K \,\mathrm{diag}(v)$. The problem of finding these two vectors u, v is known as a matrix balancing problem, and is typically solved using Sinkhorn's [1967] algorithm. This algorithm has linear convergence, and requires $O(ns)$ operations at each iteration. The benefits of using an entropic regularization are not just computational. In many cases, the linear program defined in Equation (1) does not have a unique solution. Because of this, W is not differentiable with respect to either its first or second variable. W_γ, on the other hand, is differentiable as soon as $\gamma > 0$. Problems in which we want to optimize an objective depending on the Wasserstein distance are thus harder to solve with $\gamma = 0$, as argued in Cuturi and Peyré [2016] when trying to compute, for instance, Wasserstein barycenters.

2.3 Legendre Transform

We show in Section 3 that the optimization problems involved for dictionary learning with a Wasserstein error term can be solved using dual problems whose objectives involve the Legendre-Fenchel conjugate of the smoothed Wasserstein distance. To abbreviate formulas, we use the following notation for a given $p \in \Sigma_n$:

$$H_p \stackrel{\text{def.}}{=} q \mapsto W_\gamma(p,q).$$

Cuturi and Peyré [2016] showed that the Legendre transform of the entropy regularized Wasserstein distance, as well as its gradient, can be computed in closed form:

$$H_p^*(g) = \gamma \left(E(p) + \langle p, \log K\alpha \rangle \right),$$
$$\nabla H_p^*(g) = \alpha \odot \left(K^T \frac{p}{K\alpha} \right).$$

where $K \stackrel{\text{def.}}{=} e^{-M/\gamma}$ and $\alpha \stackrel{\text{def.}}{=} e^{g/\gamma}$. Moreover, they showed that for $p \in \Sigma_n$ and $g \in \mathbb{R}^s$, $\nabla H_p^*(g) \in \Sigma_s$ and that ∇H_p^* is $\frac{1}{\gamma}$-Lipschitz. When $\gamma = 0$, H_p^* is not differentiable anymore but elements of its subgradient can be computed efficiently, notably for Euclidean point clouds [Carlier et al., 2015].

3 Wasserstein Dictionary Learning

3.1 Problem Formulation

Let $X \in (\Sigma_n)^m$ be a matrix of m vectors in the n-dimensional simplex. Let k be a number of dictionary elements, fixed in advance. We consider the problem

$$\min_{\Lambda \in \mathbb{R}^{k \times m}, D \in \mathbb{R}^{s \times k}} \sum_{i=1}^m H_{x_i}(D\lambda_i) + R(\Lambda, D) \tag{4}$$
$$\text{s.t. } D\Lambda \in \Sigma_s^m \quad .$$

[1] or $n^3 \log n$ if $s = O(n)$

Problem (4) is convex separately (but not jointly) in D and Λ as long as R is convex. We propose in what follows to use a block-coordinate descent on D and Λ.

Sandler and Lindenbaum [2009] show that when $R = 0$, $\gamma = 0$ and either D or Λ is fixed, Equation (4) is a linear program of dimensions $m \times n \times s$ with $m\times(n\times s+n+s)$ constraints, each involving 1, n or $t\times k$ variables. Representing these constraints is challenging for common sized datasets, and solving such problems is usually intractable. They proposed to replace the Wasserstein distance by an approximation [Shirdhonkar and Jacobs, 2008], for which the gradients are easier to compute. However this approximation can only be used when M is a distance matrix in a Euclidean space of small dimension. We propose to consider instead a positive regularization strength $\gamma > 0$. This allows us to consider any cost matrix M, rather than only pairwise distance matrices, and makes the optimization problems smooth and better behaved, in the sense that when D or Λ are full rank, the optimizers of each block update is unique. We propose next in §3.3 an entropic regularization on the columns of D and Λ to enforce positivity of these coefficients.

3.2 Wasserstein Dictionary Learning

Mixture Weights Update We consider here the case where the dictionary D is fixed, and our goal is to compute mixture weights Λ

$$\operatorname*{argmin}_{\Lambda\in\mathbb{R}^{k\times m}} \sum_{i=1}^{m} H_{x_i}(D\lambda_i), \quad \text{s.t. } D\Lambda \in \Sigma_s^m. \tag{5}$$

This problem can be solved using a gradient descent, but computing the gradient is equivalent to evaluating each $H_{x_i}(D\lambda_i)$ for $i = 1, \cdots, m$, that is solving m intermediate matrix scaling problems. We propose to use duality instead and attack the problem by exploiting the fact that $H^*_{x_i}$ have closed form gradients.

Theorem 1. *Let $\Lambda^\star$ be a solution of Problem (5). $\Lambda^\star$ satisfies $D\lambda_i^\star = \nabla H^*_{x_i}(g_i^\star)$ for $i = 1, \ldots, m$, with*

$$g_i^\star \in \operatorname*{argmin}_{g\in\mathbb{R}^s} H^*_{x_i}(g) \text{ s.t. } D^T g = 0. \tag{6}$$

Moreover if D is full-rank this solution is unique.

Proof. Let us introduce the variable $Q = D\Lambda$. Problem (5) becomes

$$\min_{\Lambda\in\mathbb{R}^{k\times m}, Q\in\Sigma_s^m} \sum_{i=1}^{m} H_{x_i}(q_i) \text{ s.t. } D\Lambda = Q.$$

It is a convex optimization problem with affine constraints, so strong duality holds and its dual reads

$$\begin{aligned}
&\max_{G\in\mathbb{R}^{s\times m}} \min_{\Lambda\in\mathbb{R}^{k\times m}, Q\in\Sigma_s^m} \sum_{i=1}^{m} H_{x_i}(q_i) + \langle D\Lambda - Q, G\rangle \\
&= \max_{G\in\mathbb{R}^{s\times m}} \min_{\Lambda\in\mathbb{R}^{k\times m}} \langle D\Lambda, G\rangle + \min_{Q\in\Sigma_t^m} \sum_{i=1}^{m} H_{x_i}(q_i) - \langle q_i, g_i\rangle \\
&= \max_{G\in\mathbb{R}^{s\times m}} \min_{\Lambda\in\mathbb{R}^{k\times m}} \langle D\Lambda, G\rangle - \sum_{i=1}^{m} H^*_{x_i}(g_i) \qquad (7) \\
&= \max_{G\in\mathbb{R}^{s\times m}} \min_{\Lambda\in\mathbb{R}^{k\times m}} \sum_{i=1}^{m} \langle \Lambda_i, D^T g_i\rangle - H^*_{x_i}(g_i).
\end{aligned}$$

If $D^T G \neq 0$, then $\min_{\Lambda\in\mathbb{R}^{k\times m}} \sum_{i=1}^{m} \langle \lambda_i, D^T g_i\rangle = -\infty$. Since $H^*_{x_i}(g_i)$ is finite for all i, the maximum over G is realized only if $D^T G = 0$. The problem becomes

$$\max_{\substack{G\in\mathbb{R}^{s\times m} \\ \text{s.t. } D^T G=0}} -\sum_{i=1}^{m} H^*_{x_i}(g_i) = \sum_{i=1}^{m} \max_{\substack{g\in\mathbb{R}^s \\ \text{s.t. } D^T g=0}} -H^*_{x_i}(g).$$

This is the same optimization problem as in Equation (6) and it has only one solution $G^\star$. The first order conditions of (7) are $D\Lambda^\star = \left(\nabla H^*_{x_i}(g_i^\star)\right)_{i=1}^m$. If D is full rank, this linear equation has a unique solution. □

Remark 1. *Here, for $i = 1 \ldots m$, $D\lambda_i^\star$ is in the simplex because $\nabla H^*_{x_i}(g_i)$ is itself in the simplex (see section 2.3). However the column of $\Lambda^\star$ are not required to be in the simplex and could even possibly take negative values. If all the columns of D are in the simplex, then, however, columns of $\Lambda^\star$ need to sum to 1.*

We solve Equation (6) with a projected gradient descent and then recover $\Lambda^\star$ by solving the linear equation $D\Lambda^\star = \left(\nabla H^*_{x_i}(g_i^\star)\right)_{i=1}^m$.

Dictionary Update Assuming weights Λ are fixed, our goal is now to learn the dictionary matrix D.

Theorem 2. *Let $D^\star$ be a solution of*

$$\min_{D\in\mathbb{R}^{s\times k}} \sum_{i=1}^{m} H_{x_i}(D\lambda_i) \text{ s.t. } D\Lambda \in \Sigma_s^m.$$

$D^\star$ satisfies $D^\star\Lambda = \left(\nabla H^*_{x_i}(g_i^\star)\right)_{i=1}^m$, with

$$G^\star \in \operatorname*{argmin}_{G\in\mathbb{R}^{s\times m}} \sum_{i=1}^{m} H^*_{x_i}(g_i) \text{ s.t. } G\Lambda^T = 0. \tag{8}$$

Moreover if Λ is full-rank this solution is unique.

The proof is similar to that of Theorem 1. We solve Equation (8) with a projected gradient descent and then recover $D^\star$ by solving the linear equation $D^\star\Lambda = \left(\nabla H^*_{x_i}(g_i^\star)\right)_{i=1}^m$.

3.3 Wasserstein NMF

In order to enforce non-negativity constraints on the variables, we consider the problem

$$\min_{\substack{\Lambda\in\Sigma_k^m\\ D\in\Sigma_s^k}} \sum_{i=1}^m H_{x_i}(D\lambda_i) - \rho_1 E(\Lambda) - \rho_2 E(D) \quad \text{s.t. } D\Lambda \in \Sigma_s^m \quad , \tag{9}$$

where E is defined for matrices which columns are in the simplex as $E(A) = \langle A, \log A\rangle$. This regularization allows us to derive similar results as those in Section 3.2 which we can use to find non-negative iterates for D and Λ efficiently.

Enforcing Positive Weights Problem (9) with a fixed dictionary is convex but as in Section 3.2 the gradient of the objective is computationally expensive. We have a similar duality result:

Theorem 3. *The solution of*

$$\min_{\Lambda\in\Sigma_k^m} \sum_{i=1}^m H_{x_i}(D\lambda_i) - \rho_1 E(\lambda_i) \text{ s.t. } D\Lambda \in \Sigma_s^m,$$

is $\Lambda^\star = \left(\frac{e^{-D^T g_i^\star/\rho_1}}{\langle e^{-D^T g_i^\star/\rho_1}, \mathbf{1}\rangle}\right)_{i=1}^m$, *with*

$$g_i^\star \in \operatorname*{argmin}_{g\in\mathbb{R}^s} H_{x_i}^*(g) - \rho_1 E_*(-D^T g/\rho_1). \tag{10}$$

The proof, similar to that of Theorem 1, is given in appendix. The objective and gradient of the optimization problem in Equation (10) can be computed in closed form with the following formulas:

$$E_*(x) = -\log\langle e^x, \mathbf{1}\rangle, \quad \nabla E_*(x) = -\frac{e^x}{\langle e^x, \mathbf{1}\rangle}.$$

We solve Equation (10) with an accelerated gradient scheme [Nesterov, 1983]. The gradient of the objective in Equation (10) is

$$\left(\nabla H_{x_i}^*(g_i) - D\frac{e^{-D^T g_i/\rho_1}}{\langle e^{-D^T g_i/\rho_1}, \mathbf{1}\rangle}\right)_{i=1}^m.$$

Enforcing Positive Dictionaries Similarly, we have the following theorem:

Theorem 4. *The solution of*

$$\min_{D\in\Sigma_s^k} \sum_{i=1}^m H_{x_i}(D\lambda_i) - \rho_2 \sum_{i=1}^k E(d_i) \text{ s.t. } D\Lambda \in \Sigma_s^m,$$

is $D^\star = \left(\frac{e^{-G^\star \Lambda_i^T/\rho_2}}{\langle e^{-G^\star \Lambda_i^T/\rho_2}, \mathbf{1}\rangle}\right)_{i=1}^k$, *with*

$$G^\star \in \operatorname*{argmin}_{G\in\mathbb{R}^{s\times m}} \sum_{i=1}^m H_{x_i}^*(g_i) - \sum_{i=1}^k \rho E_*(-G\Lambda_i^T/\rho_2). \tag{11}$$

The proof is similar to that of Theorem 3. We solve Equation (11) with an accelerated gradient scheme. The gradient of the objective of in Equation (11) is

$$\left(\nabla H_{x_i}^*(g_i)\right)_{i=1}^m - \sum_{i=1}^k \frac{e^{-G\Lambda_i^T/\rho_2}\Lambda_i}{\langle e^{-G\Lambda_i^T/\rho_2}, \mathbf{1}\rangle}.$$

3.4 Convergence

As pointed by Sandler and Lindenbaum [2009], the alternate optimization process generates a sequence of lower bounded non-increasing values for the objective of Problem (4), so the sequence of objectives converges. When, moreover, we use an entropic regularization ($\rho_1, \rho_2 > 0$, §3.3), successive updates for D and Λ remain in the simplex, which is compact, and thus satisfy the conditions of [Tropp, 2003, Theorem 3.1], taking into account that the hypothesis made in that theorem that the divergence is definite is not actually used in the proof. Thus every accumulation point of the sequences of iterates of D and Λ is a generalized fixed point. Moreover, if the iterates remain of full rank, then Theorem 3.2 in the same reference applies, and the sequences either converge or have a continuum of accumulation points. Although this full rank hypothesis is not guaranteed to hold, we observe that it holds in practice when the entropic regularization term does not dominate the objective.

3.5 Implementation

Projection Step for Unconstrained Dictionary Learning We solve Equations (6) and (8) with projected gradient descent methods. The orthogonal projector of the optimization problem is $\text{proj}_{\text{Ker}(\text{D}^\text{T})} := G \mapsto G - DD^+G$ in Equation (6) and $\text{proj}_{\text{Ker}(\Lambda)} := G \mapsto G - G\Lambda^+\Lambda$ in Equation (8). Precomputing DD^+ (resp. $\Lambda^+\Lambda$) uses $O(s^2)$ (resp. $O(m^2)$) memory space, and then the projection is performed in complexity $O(s^2 \times m)$ (resp. $O(s \times m^2)$). When either s or m is large, storing such a matrix is too expensive and leads to slowdowns due to memory management. In such a case, we can precompute D^+ (resp. Λ^+), which takes $O(s\times k)$ (resp. $O(m\times k)$) memory space, and compute $\text{proj}_{\text{Ker}(\text{D}^\text{T})}(G)$ as $G - D(D^+G)$ (resp. $\text{proj}_{\text{Ker}(\Lambda)}(G)$ as $G - (G\Lambda^+)\Lambda$) in $O(s^2 \times m^2 \times k^2)$ operations.

Parallelization of the Dictionary Update Parallelization on multiple processes is easy for the weights updates because each weight vector λ_i can be computed independently. The dictionary updates however cannot be reduced to completely independent subproblems. Indeed the constraint in Equation (8) makes a dependence on the columns of D. Similarly the gradient of the objective in Equation (11) cannot be separated into independent sub-problems.

We show how to use parallel processes to speed-up the unconstrained dictionary updates. The most computationally expensive part it to solve the optimization problem of Equation (8). The objective and gradient of this problem can be computed independently for each column. Then we can gather the gradient on a single process and project it. Since the constraint is linear we can directly project the gradient before computing the step-size of the descent, so that if this computation involves computing the objective (like a backtracking line-search does for example) the projection does not need to be repeated.

We also propose a scheme to partially parallelize the positive dictionary updates. The objective and gradient of the optimization problem in Equation (11) are found by computing $e^{-G\Lambda^T}$, which cannot be computed separately on columns of G. An efficient way to still compute $e^{-G\Lambda^T}$ in parallel is to split G column-wise into $(G^{(1)}, \dots, G^{(p)})$ where p is the number of processes available, and compute $e^{-G^{(i)}\Lambda^T}$ on process i. The managing process computes $e^{-G\Lambda^T}$ as $\prod_{i=1}^{p} e^{-G^{(i)}\Lambda^T}$ (here the product is point-wise) and gives the result to all the other processes so that they can finish computing the gradient. By doing so most of the work is done in parallel and each process only shares a matrix of size $s \times k$ twice per gradient/objective calculation. Since usually $k \ll m$ this allows to use all the available processes while keeping communication overhead low.

4 Experiments

4.1 Face Recognition

We reproduce here the face recognition experiment of Sandler and Lindenbaum [2009] on the ORL dataset [Samaria and Harter, 1994] with the same preprocessing, classification and evaluation method in order to compare computation time. Each image is downsampled so that its longer side is 32. We represent images as column vectors that we normalize so that they sum to 1 and store them in matrix X. The cost matrix M is the Euclidean distance between pixels. For evaluation, the dataset is split evenly in two, trained on one set and tested on the other several times, and we take the take best classification performance obtained. Table 1 shows the classification accuracy obtained with unconstrained Wasserstein Dictionary Learning (Section 3.2). The results are comparable to those of Sandler and Lindenbaum [2009].

Learning the dictionary and coefficients with a Matlab implementation of our algorithm on an single core of a 2.4Ghz Intel Quad core i7 CPU with $k = 40$ takes on average $20s$ for $\gamma = 1/30$ and $90s$ for $\gamma = 1/50$, while Sandler and Lindenbaum [2009] report up to 20 minutes just for the D step with a comparable CPU. The whole NMF can take up to 10 minutes when we use the entropy positivity barrier with $\rho_1 = \rho_2 = 1/10$.

Table 1: Classification accuracy for the face recognition task on the ORL dataset.

k	10	20	30	40	50
$\gamma = 1/30$	93%	**95.5%**	**97%**	96.5%	96%
$\gamma = 1/50$	91%	95%	95%	**97%**	94.5%
Sandler09	**94.5%**	90.5%	95%	96.5%	**97%**

4.2 Semantic Analysis

The goal of semantic analysis is to extract a few representative histograms of words (*a.k.a.* topics) from large corpora of texts. To tackle this task, Probabilistic Latent Semantic Indexing (PLSI, Hofmann [1999]) learns a non-negative factorization of the form $X = D\Sigma\Lambda$, which models the document generation process: D is the matrix of word probabilities knowing the topic, Σ is the diagonal matrix of topic probabilities and Λ is the matrix of document probability knowing the topic. Ding et al. [2008] shows that PLSI optimizes the same objective as the algorithm in Lee and Seung [1999] for a Kullback-Leibler error term.

We use the same approach as Lee and Seung [1999] to learn topics from a database of texts with NMF. The input data is a *bag-of-words* representation of the documents. Let $Y = \{y_1, \dots, y_n\}$ be the vocabulary of the database, a text document is represented as vector of word frequencies: X_{ij} is the frequency of the word y_i in the j^{th} text. We get topics D by learning a factorization $D\Lambda$ with NMF. The cost of the factorization is usually its Euclidean distance or Kullback-Leibler divergence to X. In order to use a Wasserstein cost instead, we need a meaningful cost for transporting words from one to another.

Recent works [Pennington et al., 2014, Zou et al., 2013], building upon earlier references [Bengio et al., 2003], propose to compute Euclidean embeddings for words such that the Euclidean or cosine distances between the respective image of two words corresponds to some form of semantic discrepancy between these words. As recently shown by Kusner et al. [2015], these embeddings can be used to compare texts using the toolbox of optimal transport: Bag-of-words histograms can be compared with Wasserstein distances using the Euclidean metric between the words as the ground metric M. We leverage these results to learn topics from a text database using Wasserstein NMF.

4.2.1 Datasets

We learned topics on two datasets labeled. Labels are ignored for performing NMF, and are only used for evaluation. For each dataset, let m be the number of documents, n the vocabulary size and c the number of labels. (i) **BBCsport** [Greene and Cunningham, 2006] is a dataset of news articles about sports, labeled according to which sport the article is about, in which we removed stop-words ($n = 12,669$, $m = 737$, $c = 5$). We split the dataset as a 80/20 training / testing set for classification. (ii) **Reuters** is a dataset of news articles labeled according to their area of interest. We used the version described in Cardoso-Cachopo [2007], with the same train-test split for classification, and removed stop-words and words that appeared only once across the corpus ($n = 13,038$, $m = 7,674$, $c = 8$).

4.2.2 Monolingual Semantic Analysis

We used a pretrained Glove word embedding [Pennington et al., 2014] to map words to a Euclidean space of dimension 300. Let $\vec{y_1}, \dots, \vec{y_s}$ be the embeddings of the words in the dataset's vocabulary, and $\vec{z_1}, \dots, \vec{z_s}$ the embeddings of the words in the target vocabulary, that is the words that are allowed to appear in the topics. We define the cost matrix of the Wasserstein distance as the cosine distance in the embedding: $M_{ij} = 1 - \frac{\langle \vec{y_i}, \vec{z_j} \rangle}{\|\vec{y_i}\| \|\vec{z_j}\|}$. We then find D and Λ with Wasserstein NMF (W-NMF, Section 3.3).

Figure 2 shows a word cloud representation (`wordle.net`) for 4 relevant topics for the dataset BBCsport. Depending on the parameters, the full Wasserstein NMF computation takes from 20 minutes to an hour for BBCsport and around 10 hours for Reuters using a Matlab implementation running on a single GPU of an Nvidia Tesla K80 card.

Target Words Selection Since we can choose as target words any word that is defined for the embedding, we need a way to select which to use. We chose to use a list of $3,000$ frequent words in English[2]. Other approaches can be considered such as using the dataset's vocabulary, tokenized or not, or taking the most frequent words for each class in the dataset.

4.2.3 Cross-language Semantic Analysis

Lauly et al. [2014] propose a bilingual word representation that maps words in two different languages to the same Euclidean space. By setting the vocabulary of the topics as a subset of the words in the target language, we can learn topics in that language. Figure 3

Figure 2: Word clouds representing 4 of the 15 topics learned on BBCsport in English. Top-left topic: competitions. Top-right: time. Bottom-left: soccer actions. Bottom-right: drugs.

illustrates what we would expect with $k = 1$, which is the Wasserstein iso-barycenter problem. We use a pretrained embeddings of dimension 40 from Lauly et al. [2014] in order to learn topics in French. Note that this method could also learn topics in one language from a bilingual dataset, or in both languages.

As in Section 4.2.2, we use the cosine distance in the embedding as the ground metric. Table 4 shows word cloud representations for 4 relevant topics for the dataset Reuters. Computation times are similar to those with a target vocabulary in English.

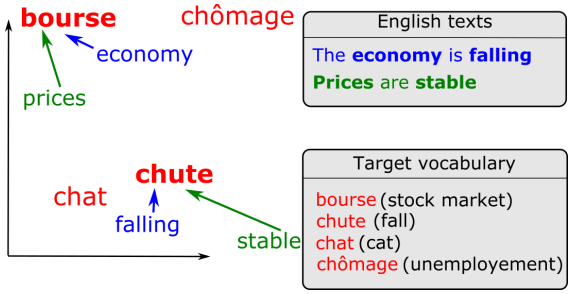

Figure 3: The Wasserstein iso-barycenter of two English sentences with a target vocabulary in French. Arrows represent the optimal transport plan from a text to the barycenter. The barycenter is supported on the bold red words which are pointed by arrows. The barycenter is not equidistant to the extreme points because the set of possible features is discrete.

Target words selection We chose as the target dictionary a list of of $6,000$ frequent words in French[3].

[2] Available at `https://simple.wiktionary.org/wiki/Wiktionary:BNC_spoken_freq`

[3] Available at `http://wortschatz.uni-leipzig.de/Papers/top10000fr.txt`

Figure 4: Word clouds representing 4 of the 24 topics learned on Reuters in French. Top-left topic: international trade. Top-right: oil and other resources. Bottom-left: banking. Bottom-right: management and funding.

Table 2: Text classification error. W-NMF$_f$ is the classifier using W-NMF with a French target vocabulary.

Method	KL-NMF	E-NMF	W-NMF	W-NMF$_f$
Reuters	6.9%	8.2%	**6.0**%	9.8%
BBCsport	9.4%	12.8%	**5.4**%	20.8%

4.2.4 Classification Performance

We compared the classification error obtained on the two datasets with our method to those obtained by using the mixture weights produced by Euclidean NMF (E-NMF) and Kullback-Liebler NMF (KL-NMF). We use a k-NN classifier with a Hellinger distance between the mixture weights. k is selected by 10-fold cross-validation on the training set, using the same partitions for all methods. We set the number of topics to $3c$. Parameters γ, ρ_1 and ρ_2 were set to be as small as we could (small values can make the gradients infinite because of machine precision) without a particular selection procedure. See supplementary materials for a representation of all the topics of every method.

Wasserstein NMF with a target vocabulary in English performs better on this auxiliary task than Euclidean or KL NMF. Although this does not prove that the topics are of better quality, it shows that Wasserstein NMF can drastically reduce the vocabulary size without loosing discriminative power. As we can see in Figures 2, 4, the topics themselves are semantically coherent and related to the datasets' content.

The classification error for W-NMF with a French target vocabulary on BBCsports is rather bad, although the topics are coherent and related to the content of the articles. The confusion matrix (Table 3) shows that more than half of the articles about tennis are misclassified. In fact, the other methods produce a topic about tennis, but W-NMF with a French dictionary does not. Table 4 shows the French words closest to some English query words according to the ground metric. While the closest words to football and bank are semantically related to their query word, the closest words to tennis are not. This illustrates how our method relies on the ground metric, given by word embeddings in this case.

Table 3: Confusion matrices for BBCsports for k-NN with W-NMF. Columns represent the ground truth and lines predicted labels. Labels: athletism (a), cricket (c), football (f), rugby (r) and tennis (t).

English target vocabulary

	a	c	f	r	t
a	21	0	0	0	0
c	0	25	0	0	0
f	0	0	50	**4**	**1**
r	0	0	**3**	26	0
t	0	0	0	0	19

French target vocabulary

	a	c	f	r	t
a	18	0	**1**	0	**2**
c	0	22	**3**	0	**2**
f	**3**	0	44	**5**	**6**
r	0	**3**	**3**	25	**1**
t	0	0	**2**	0	9

Table 4: 10 French words closest to some English words according to the ground metric

football	football supporters championnat sportives sportifs joueurs sportif jeux matches sport
bank	banque banques bei bancaire federal bank emprunts reserve crédit bancaires
tennis	bienfaiteurs murray ex-membre ballet b92 sally sylvia markovic hakim socialo-communiste

5 Conclusion

We show how to efficiently perform dictionary leaning and NMF using optimal transport as the data fitting term, with an optional entropy positivity barrier. Our method can be applied to large datasets in high dimensions and does not require any assumption on the cost matrix. We also show that with this data fitting term, the reconstruction $D\Lambda$ can use different features than the data X. Other than our application to cross-language semantic analysis, this can be used for example to reduce the number of target features by quantization for the dictionary while keeping the original features for the dataset.

While we only consider entropy as a barrier for positivity in this work, our approach can be generalized to other regularizers, as long as the gradient of $R_\star$ or its proximal operator can be computed efficiently. We believe that extensions to other classes of regularizers is an interesting area for future work.

References

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003.

A. Cardoso-Cachopo. Improving Methods for Single-label Text Categorization. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, 2007.

G. Carlier, A. Oberman, and E. Oudet. Numerical methods for matching for teams and wasserstein barycenters. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1621–1642, 2015.

M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013.

M. Cuturi and D. Avis. Ground metric learning. *The Journal of Machine Learning Research*, 15(1):533–564, 2014.

M. Cuturi and A. Doucet. Fast computation of wasserstein barycenters. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014.

M. Cuturi and G. Peyré. A smoothed dual approach for variational Wasserstein problems. *SIAM Journal on Imaging Sciences*, 2016. to appear.

C. Ding, T. Li, and W. Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927, 2008.

D. Greene and P. Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd International Conference on Machine learning (ICML-06)*, pages 377–384. ACM Press, 2006.

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

M.J. Kusner, Y. Sun, N.I. Kolkin, and K.U. Weinberger. From word embeddings to document distances. In *Proceedings of The 32nd International Conference on Machine Learning (ICML-16)*, 2015.

S. Lauly, H. Larochelle, M. Khapra, B. Ravindran, V.C. Raykar, and A. Saha. An autoencoder approach to learning bilingual word representations. In *Advances in Neural Information Processing Systems*, pages 1853–1861, 2014.

D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755): 788–791, 1999.

D.D. Lee and H.S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562, 2001.

Y. Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady. Vol. 27. No. 2.*, 1983.

James B Orlin. A faster strongly polynomial minimum cost flow algorithm. *Operations research*, 41(2):338–350, 1993.

P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2): 111–126, 1994.

J. Pennington, R. Socher, and C.D. Manning. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543, 2014.

J. Rabin and N. Papadakis. Convex color image segmentation with optimal transport distances. In *Scale Space and Variational Methods in Computer Vision*, pages 256–269. Springer, 2015.

Y. Rubner, C. Tomasi, and L.J. Guibas. A metric for distributions with applications to image databases. In *Computer Vision, 1998. Sixth International Conference on*, pages 59–66. IEEE, 1998.

F.S. Samaria and A.C. Harter. Parameterisation of a stochastic model for human face identification. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pages 138–142. IEEE, 1994.

R. Sandler and M. Lindenbaum. Nonnegative matrix factorization with earth mover's distance metric. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1873–1880. IEEE, 2009.

M.N. Schmidt, J. Larsen, and F.T. Hsiao. Wind noise reduction using non-negative sparse coding. In *Machine Learning for Signal Processing, 2007 IEEE Workshop on*, pages 431–436. IEEE, 2007.

S. Shirdhonkar and D.W. Jacobs. Approximate earth movers distance in linear time. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967.

J. Solomon, F. de Goes, Pixar Animation Studios, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L.J. Guibas. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (Proc. SIGGRAPH 2015)*, 2015.

J.A. Tropp. An alternating minimization algorithm for non-negative matrix approximation, 2003.

C. Villani. *Optimal transport: old and new*, volume 338. Springer Verlag, 2009.

G. Zen, E. Ricci, and N. Sebe. Simultaneous ground metric learning and matrix factorization with earth mover's distance. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 3690–3695. IEEE, 2014.

S. Zhang, W. Wang, J. Ford, and F. Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *SDM*, volume 6, pages 548–552. SIAM, 2006.

W.Y. Zou, R. Socher, D.M. Cer, and C.D. Manning. Bilingual word embeddings for phrase-based machine translation. pages 1393–1398, 2013.