Unsupervised Alignment of Embeddings with Wasserstein Procrustes

Edouard Grave
Facebook AI Research

Armand Joulin
Facebook AI Research

Quentin Berthet
Statistical Laboratory
DPMMS, University of Cambridge

Abstract

We consider the task of aligning two sets of points in high dimension, which has many applications in natural language processing and computer vision. As an example, it was recently shown that it is possible to infer a bilingual lexicon, without supervised data, by aligning word embeddings trained on monolingual data. These recent advances are based on adversarial training to learn the mapping between the two embeddings. In this paper, we propose to use an alternative formulation, based on the joint estimation of an orthogonal matrix and a permutation matrix. While this problem is not convex, we propose to initialize our optimization algorithm by using a convex relaxation, traditionally considered for the graph isomorphism problem. We propose a stochastic algorithm to minimize our cost function on large scale problems. Finally, we evaluate our method on the problem of unsupervised word translation, by aligning word embeddings trained on monolingual data. On this task, our method obtains state of the art results, while requiring less computational resources than competing approaches.

1 Introduction

Aligning two clouds of embeddings, or high dimensional real vectors, is a fundamental problem in machine learning with applications in natural language processing such as unsupervised word and sentence translation (Rapp, 1995; Fung, 1995) or in computer vision such as point set registration (Cootes et al., 1995) and structure-from-motion (Tomasi and Kanade, 1992). Most of the successes of these methods were made in domains where either the dimension of the vectors were small or some geometrical constraints on the point clouds were known (Fischler and Bolles, 1987; Rangarajan et al., 1997; Leordeanu and Hebert, 2005; Liu et al., 2008).

When dealing with unstructured sets of high dimensional vectors, it is quite common to use a few anchor points to learn the matching (Mikolov et al., 2013; Xing et al., 2015). The supervision for this problem can be limited or noisy, for example using exact string matches in the context of word vectors alignment (Artetxe et al., 2017). Recently, several unsupervised approaches have obtained compelling results, by framing this problem as some form of distance minimization between distributions, using either the Wasserstein distance or adversarial training (Cao et al., 2016; Zhang et al., 2017a; Conneau et al., 2017; Alvarez-Melis and Jaakkola, 2018). The methods typically require a relatively sophisticated framework that leads to a hard, and sometimes unstable, optimization problem. Moreover, both weakly supervised and unsupervised methods greatly benefit from a refinement procedure (Artetxe et al., 2017; Conneau et al., 2017), often based on some variants of iterative closest points (ICP, Besl and McKay, 1992). It is thus not surprising that a more direct approach, only relying on ICP, was able to achieve similar performance (Hoshen and Wolf, 2018). However, this method is still very sensitive to the initialization, and requires a large number of random restarts, based on randomized principal component analysis to converge.

In this work, we propose a simple approach based on jointly learning the alignment and the linear transformation between the two point clouds. Our objective function is similar to the one of Rangarajan et al. (1997), Zhang et al. (2017b) and Alvarez-Melis et al. (2018) but our algorithm largely differs. In particular, our algorithm shares similarities with the work of Bojanowski and Joulin (2017) where a non-linear transformation and an alignment between two point clouds are jointly learned. While their goal is completely different, we take a similar optimization scheme to scale our ap-

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

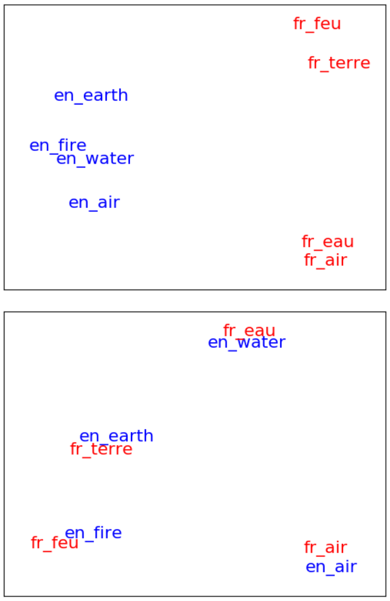

Figure 1: Illustration of the unsupervised alignment problem for word vectors. The goal is to jointly estimate the transformation to map the vectors, as well as the correspondence between the words. Left: PCA on the non-aligned word embeddings. Right: PCA on the word embeddings aligned with our method.

proach to very large sets of points. Our formulation is not convex and we propose a convex relaxation for the initialization based on a standard relaxation of the quadratic alignment problem for graph matching (Gold and Rangarajan, 1996). Our approach makes little assumption about the clouds or their distributions, is flexible and converges in a few minutes.

An interesting by-product of our formulation is that it draws some similarities between graph matching and aligning embeddings that could be used in either way. We validate this observation on several toy examples designed to give some insights on the strengths and weaknesses of our approach. We also show that our approach is competitive with the state-of-the-art among unsupervised approaches on word translation while running in a few minutes. More precisely, we make the following contributions:

- starting from a simple formulation, based on the joint estimation of an orthogonal matrix and a permutation matrix, we introduce a stochastic algorithm to minimize our cost function, which can scale to large datasets ;
- we show that this algorithm can be initialized using the solution of a convex problem, leading to better convergence ;
- we evaluate our method on toy experiments, as well as the task of bilingual lexicon induction.

2 Approach

In this section, we first describe Procrustes, which is a standard approach to align points when correspondences are known, and then the Wasserstein distance which is used to measure the distance between sets of points. We then present our method which is derived from these two techniques.

2.1 Procrustes

Procrustes analysis learns a linear transformation between two sets of matched points $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\mathbf{Y} \in \mathbb{R}^{n\times d}$. If the correspondences between the two sets are known (i.e., which point of $\mathbf{X}$ corresponds to which point of $\mathbf{Y}$), then the linear transformation can be recovered by solving the least square problem:

$$\min_{\mathbf{W}\in\mathbb{R}^{d\times d}} \|\mathbf{XW} - \mathbf{Y}\|_2^2.$$

This technique has been successfully applied in many different fields, from analyzing sets of 2D shapes (Goodall, 1991) to learning a linear mapping between word vectors in two different languages with the help of a bilingual lexicon (Mikolov et al., 2013). Constraints on the mapping $\mathbf{W}$ can be further imposed to suit the geometry of the problem. For example, Xing et al. (2015) have shown empirically that orthogonal transformations are well suited to the mapping of word vectors. The corresponding orthogonal Procrustes corresponds to the following optimization problem:

$$\min_{\mathbf{Q}\in\mathcal{O}_d} \|\mathbf{XQ} - \mathbf{Y}\|_2^2, \tag{1}$$

where $\mathcal{O}_d$ is the set of orthogonal matrices. This orthogonality constraint is particularly interesting since it ensures that the distances between points are unchanged by the transformation. As shown by Schönemann (1966), the orthogonal Procrustes problem has a closed form solution equal to $\mathbf{Q}^* = \mathbf{UV}^\top$, where $\mathbf{USV}^\top$ is the singular value decomposition of $\mathbf{X}^\top\mathbf{Y}$.

2.2 Wasserstein distance

On the other hand, if we assume that the transformation between the two sets of points is known, finding the correspondences between these sets can be formulated as the optimization problem:

$$\min_{\mathbf{P}\in\mathcal{P}_n} \|\mathbf{X} - \mathbf{PY}\|_2^2,$$

where $\mathcal{P}_n$ is the set of permutation matrices, i.e. the set of binary matrices that enforces a 1-to-1 mapping:

$\mathcal{P}_n = \left\{\mathbf{P} \in \{0,1\}^{n\times n}, \quad \mathbf{P}\mathbf{1}_n = \mathbf{1}_n, \quad \mathbf{P}^\top\mathbf{1}_n = \mathbf{1}_n\right\}$. Enforcing a 1-to-1 mapping is not always realistic but it has the advantage to be related to a set of orthogonal matrices. Indeed, this makes the previous optimization problem equivalent to the following linear program:

$$\max_{\mathbf{P}\in\mathcal{P}_n} \operatorname{tr}\left(\mathbf{X}^\top\mathbf{P}\mathbf{Y}\right) = \max_{\mathbf{P}\in\mathcal{P}_n} \operatorname{tr}\left(\mathbf{P}\mathbf{Y}\mathbf{X}^\top\right).$$

The Kantorovitch formulation is known in the case of uniform distributions as the optimal assignment problem (Kuhn, 1955). It can be solved using the Hungarian algorithm, which has a complexity of $O(n^3)$. For large number n of points, the Hungarian algorithm is impractical and an approximation is required. For example, in the context of word translation, Zhang et al. (2017b) uses the fact that this problem is equivalent to minimizing the squared Wasserstein distance between the two sets of points $\mathbf{X}$ and $\mathbf{Y}$:

$$W_2^2(\mathbf{X},\mathbf{Y}) = \min_{\mathbf{P}\in\mathcal{P}_n} \sum_{i,j} P_{ij}\|\mathbf{x}_i - \mathbf{y}_j\|_2^2$$

to use an approximate Earth Mover Distance solver based on the Sinkhorn algorithm (Cuturi, 2013).

2.3 Procrustes in Wasserstein distance

In our case, we do not know the correspondence between the two sets, nor the linear transformation. Formally, our goal is thus to learn an orthogonal matrix $\mathbf{Q} \in \mathcal{O}_d$, such that the set of point $\mathbf{X}$ is close to the set of point $\mathbf{Y}$ and 1-to-1 correspondences can be inferred. We use the Wasserstein distance defined in Eq (2.2), as the measure of distance between our two sets of points and we combine it with the orthogonal Procrustes defined in Eq. (1), leading to the problem of Procrustes in Wasserstein distance:

$$\min_{\mathbf{Q}\in\mathcal{O}_d} W_2^2\left(\mathbf{X}\mathbf{Q},\mathbf{Y}\right) = \min_{\mathbf{Q}\in\mathcal{O}_d}\min_{\mathbf{P}\in\mathcal{P}_n} \|\mathbf{X}\mathbf{Q} - \mathbf{P}\mathbf{Y}\|_2^2. \qquad (2)$$

The problem is not jointly convex in $\mathbf{Q}$ and $\mathbf{P}$ but, as shown in the previous sections, there are exact solutions for each optimization problem if the other variable is fixed. Naively alternating the minimization in each variable does not scale and empirical results show that it quickly converges to bad local minima even on small problems (Zhang et al., 2017b).

2.4 Stochastic optimization

In this section, we describe a scalable optimization scheme for the problem defined in Eq. (2). The objective of the problem in Eq. (2) can be intepreted as the Wasserstein distance $W_2^2(p^{(n)}_{\mathbf{XQ}}, p^{(n)}_{\mathbf{Y}})$, between $p^{(n)}_{\mathbf{XQ}}$ and $p^{(n)}_{\mathbf{Y}}$, two empirical distributions of size n of the vectors of $\mathbf{XQ}$ and $\mathbf{Y}$. One possible approach would be to alternate full minimization of $\|\mathbf{XQ} - \mathbf{PY}\|_2^2$ in $\mathbf{P} \in \mathcal{P}_n$ and a gradient-based update in $\mathbf{Q}$. One difficulty with this method, and our formulation, is that the dimension of the permutation matrix $\mathbf{P}$ scales quadratically with the number of points n. Finding the optimal matching for a given orthogonal matrix has a complexity of $\mathcal{O}(n^3)$, or of $\mathcal{O}(n^2)$ up to logarithmic terms for an approximate matching via Sinkhorn (Cuturi, 2013; Altschuler et al., 2017). As a consequence, we use instead at each step t a new batch of two subsamples of size $b \le n$, denoted by $\mathbf{X}_t$ and $\mathbf{Y}_t$. We compute at each step the optimal matching $\mathbf{P_t} \in \mathcal{P}_b$ and the value $W_2^2(p^{(b)}_{\mathbf{X_t Q}}, p^{(b)}_{\mathbf{Y}_t})$. We then perform a gradient-guided step in $\mathbf{Q}$ for $\|\mathbf{X}_t\mathbf{Q} - \mathbf{P}_t\mathbf{Y}\|_2^2$. This surrogate objective, at each step, can be seen as a *subsampled* version of size b of our objective.

As in Genevay et al. (2018), we use samples from distributions for stochastic minimization of an objective involving a Wasserstein distance. Similarly, our algorithm can be intepreted as the stochastic optimization of a population version objective $W_2^2(p_{xQ}, p_y)$, seeing $\mathbf{X}, \mathbf{Y}$ as i.i.d. samples of size n from latent unknown distributions p_x and p_y, and $\mathbf{X}_t, \mathbf{Y}_t$ as samples of size b. At a high level, this approach can be motivated by the convergence of $W_2(\mu_m, \nu_m)$ to $W_2(\mu, \nu)$, where μ_m and ν_m are the empirical distributions of i.i.d. samples of size m from latent distributions μ and ν, via the analysis of $W_2(\mu_m, \mu)$ and the triangle inequality. Results of these type, and analysis of the rates of convergence go back to Dudley (1969) for the W_1 distance, and have been extended to all W_p distances, including $p = 2$ (Weed and Bach, 2017; Weed and Berthet, 2019), the case of discrete distributions (Tameling et al., 2017; Sommerfeld et al., 2018), and the use of these methods in variational problems (Bernton et al., 2017). These results do not give formal guarantees for our approach, they provide motivation for the success of this method.

Optimizing a general function over the manifold of the orthogonal matrices, the *Stiefel manifold*, as been thoroughly studied (Absil et al., 2009). A simple solution is to take a step in a descending direction (given the full gradient or a stochastic approximation) while pulling back the update on the manifold. Most of the pull back operators are computationally expensive but our matrix $\mathbf{Q}$ is relatively small. In particular, we use the projection even though it requires a singular value decomposition as shown in the previous section. Overall, our optimization in $\mathbf{Q}$ is a stochastic gradient descent (SGD) with a projection on the Stiefel manifold.

The overall optimization scheme is summarized in Algorithm 1. At the iteration t, we sample a mini-batch $\mathbf{X}_t \in \mathbb{R}^{b\times d}$ and $\mathbf{Y}_t \in \mathbb{R}^{b\times d}$ of points of size b from the matrices $\mathbf{X}$ and $\mathbf{Y}$. We then compute the optimal permutation for this batch by solving the same problem as in Eq. (2.2). Given this permutation, we compute the gradient in $\mathbf{Q}$ to update the matrix. While this procedure is efficient and can scale to very large sets of points, it is not convex and the quality of its solution depends on its initialization. In particular, the initial point $\mathbf{Q}_0$ is very important for the quality of the final solution. In the next section, we discuss a strategy to initialize this optimization problem.

Algorithm 1 Stochastic optimization

```
for $t = 1$ to $T$ do
    Draw $\mathbf{X}_t$ from $\mathbf{X}$ and $\mathbf{Y}_t$ from $\mathbf{Y}$, of size $b$
    Compute the optimal matching between $\mathbf{X}_t$ and $\mathbf{Y}_t$ given the current orthogonal matrix $\mathbf{Q}_t$
        $\mathbf{P}_t = \operatorname{argmax}_{\mathbf{P} \in \mathcal{P}_b} \operatorname{tr}\left(\mathbf{Y}_t \mathbf{Q}_t^\top \mathbf{X}_t^\top \mathbf{P}\right).$
    Compute the gradient $\mathbf{G}_t$ with respect to $\mathbf{Q}$:
        $\mathbf{G}_t = -2\mathbf{X}_t^\top \mathbf{P}_t \mathbf{Y}_t.$
    Perform a gradient step and project on the set of orthogonal matrices:
        $\mathbf{Q}_{t+1} = \Pi_{\mathcal{O}_d}\left(\mathbf{Q}_t - \alpha \mathbf{G}_t\right).$
    For a matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$, the projection is given by $\Pi_{\mathcal{O}_d}(\mathbf{M}) = \mathbf{U}\mathbf{V}^\top$, with $\mathbf{U}\mathbf{S}\mathbf{V}^\top$ the singular value decomposition of $\mathbf{M}$.
end for
```

2.5 Convex relaxation

In this section, we propose a convex relaxation of the problem defined in Eq. (2). This relaxation comes from the observation that our problem is equivalent to

$$\max_{\mathbf{P} \in \mathcal{P}_n} \max_{\mathbf{Q} \in \mathcal{O}_d} \operatorname{tr}\left(\mathbf{Q}^\top \mathbf{X}^\top \mathbf{P} \mathbf{Y}\right).$$

Solving a linear program over $\mathcal{O}_d$ is equivalent to solving it as over its convex hull, i.e., the set of matrices with a spectral norm lower than 1. This value at this maximum is thus equal to the dual norm of the spectral norm, i.e., the trace norm (or nuclear norm) of $\mathbf{X}^\top \mathbf{P} \mathbf{Y}$. Thus, the problem is equivalent to:

$$\max_{\mathbf{P} \in \mathcal{P}_n} \|\mathbf{X}^\top \mathbf{P} \mathbf{Y}\|_*,$$

where $\mathbf{Z} \mapsto \|\mathbf{Z}\|_*$ is the trace norm. The trace norm requires to compute the singular values of the matrix $\mathbf{X}^\top \mathbf{P} \mathbf{Y}$, which is computationally expensive. Another possible formulation is to replace the trace norm by the Frobenius norm, leading to the following:

$$\max_{\mathbf{P} \in \mathcal{P}_n} \|\mathbf{X}^\top \mathbf{P} \mathbf{Y}\|_2^2 = \max_{\mathbf{P} \in \mathcal{P}_n} \operatorname{tr}\left(\mathbf{K}_\mathbf{Y} \mathbf{P}^\top \mathbf{K}_\mathbf{X} \mathbf{P}\right), \quad (3)$$

where $\mathbf{K}_\mathbf{X} = \mathbf{X}\mathbf{X}^\top$ and $\mathbf{K}_\mathbf{Y} = \mathbf{Y}\mathbf{Y}^\top$. This formulation has the advantage of leading to a quadratic assignment program, i.e.,

$$\min_{\mathbf{P} \in \mathcal{P}_n} \|\mathbf{K}_\mathbf{X} \mathbf{P} - \mathbf{P} \mathbf{K}_\mathbf{Y}\|_2^2, \quad (4)$$

since the permutation matrices are orthogonal, and the Frobenius norm is invariant by multiplication with an orthogonal matrix. This problem is a standard formulation for the graph isomorphism, or graph matching problem and it is known to be NP-hard in general (Garey and Johnson, 2002).

For this problem, many convex relaxations have been proposed. Of particular interest, Gold and Rangarajan (1996) replace the set of permutation matrices by its convex hull, the set of doubly stochastic matrices, namely the Birkhoff polytope, leading to the following convex relaxation:

$$\min_{\mathbf{P} \in \mathcal{B}_n} \|\mathbf{K}_\mathbf{X} \mathbf{P} - \mathbf{P} \mathbf{K}_\mathbf{Y}\|_2^2, \quad (5)$$

where $\mathcal{B}_n = \text{convex-hull}(\mathcal{P}_n)$ is the Birkhoff polytope. It would be interesting to see what would be the equivalent relaxation for the trace norm but this goes beyond the scope of this paper.

We use the Frank-Wolfe algorithm to minimize this problem (Frank and Wolfe, 1956). Once the global minimizer $\mathbf{P}^*$ has been attained, we compute a corresponding orthogonal matrix $\mathbf{Q}_0$ by solving:

$$\mathbf{Q}_0 = \operatorname{argmin}_{\mathbf{Q} \in \mathcal{O}_d} \|\mathbf{X}\mathbf{Q} - \mathbf{P}^* \mathbf{Y}\|_2^2.$$

This corresponds to taking the singular value $\mathbf{U}\mathbf{S}\mathbf{V}^\top$ of $\mathbf{P}^* \mathbf{Y} \mathbf{X}^\top$, and setting $\mathbf{Q}_0 = \mathbf{U}\mathbf{V}^\top$. Note that the matrix $\mathbf{P}^*$ is not necessarily a permutation matrix, but only doubly stochastic.

2.6 Improving the nearest-neighbor search

Once the source embeddings $\mathbf{X}$ are mapped to the target space, they are not perfectly aligned with target embeddings $\mathbf{Y}$ and a retrieval procedure is required. In high dimensional vector space, some points, called *hubs*, tend to be close to disproportionally many vectors and a direct nearest-neighbor (NN) search favors these hubs (Aucouturier and Pachet, 2008; Radovanović et al., 2010). This problem has been observed in many retrieval situations (Doddington et al., 1998; Pachet and Aucouturier, 2004; Dinu et al., 2014).

Several strategies have been proposed to diminish the effects of hubs, most are based on some definition of a local distance (Jegou et al., 2010; Conneau et al., 2017) or by reversing the direction of the retrieval (Dinu et al., 2014; Smith et al., 2017). In this paper, we consider the Inverted Softmax (ISF) proposed by Smith et al. (2017) and the Cross-Domain Similarity Local Scaling (CSLS) of Conneau et al. (2017). The ISF is defined for normalized vectors as:

$$\mathrm{ISF}(\mathbf{y},\mathbf{z}) = \frac{\exp(\beta\mathbf{y}^\top\mathbf{z})}{\sum_{\mathbf{y}'\in\mathbf{Y}}\exp(\beta\mathbf{y}'^\top\mathbf{z})},$$

where $\beta > 0$ is a temperature parameter.

CSLS is a similarity measure between the vectors $\mathbf{y}$ and $\mathbf{z}$ from 2 different sets $\mathbf{Y}$ and $\mathbf{Z}$, defined as

$$\mathrm{CSLS}(\mathbf{y},\mathbf{z}) = 2\cos(\mathbf{y},\mathbf{z}) - R_{\mathbf{Z}}(\mathbf{y}) - R_{\mathbf{Y}}(\mathbf{z}),$$

where $\cos(\mathbf{y},\mathbf{z}) = \frac{\mathbf{y}^\top\mathbf{z}}{\|\mathbf{y}\|\|\mathbf{z}\|}$ is the cosine similarity between $\mathbf{y}$ and $\mathbf{z}$, and

$$R_{\mathbf{Z}}(\mathbf{y}) = \frac{1}{K}\sum_{\mathbf{z}\in\mathcal{N}_Z(\mathbf{y})}\cos(\mathbf{z},\mathbf{y})$$

is the average of the cosine similarity between $\mathbf{y}$ and its K nearest neighbors among the vectors in $\mathbf{Z}$. It is an extension of the contextual dissimilarity measure of Jegou et al. (2010) to two sets of points. An advantage of the CSLS measure over ISF is that its free parameter (i.e., the number of nearest neighbors) is much simpler to set than the temperature of the ISF.

3 Related work

Embedding alignment with no bilingual dictionary. When the permutation matrix is given by a bilingual dictionary, the problem stated in Eq. (2) has been studied in the context of word translation by Mikolov et al. (2013) with $\mathbf{Q}$ as an unconstrained linear transformation. Xing et al. (2015) have shown that constraining $\mathbf{Q}$ to be an orthonormal matrix yields better results if used with normalized embeddings, and Artetxe et al. (2016) have improved further the quality of supervised alignment by using centered normalized embeddings. Hoshen and Wolf (2018) is the closest attempt to learn unsupervised word translation with a formulation similar to Eq. (2). They use an iterative closest point approach initialized with a randomized PCA. Others have looked into unsupervised alignment of distributions: Cao et al. (2016) align the moments of the two distributions which assumes a Gaussian distribution over the embeddings, while others (Zhang et al., 2017a; Conneau et al., 2017) have preferred the popular Generative Adversarial Networks (GANs) framework of Goodfellow et al. (2014). Different techniques based on optimal transport have been proposed to refine the alignment obtained from generative adversarial networks (Zhang et al., 2017b; Xu et al., 2018). Closer to our work, Alvarez-Melis and Jaakkola (2018) have proposed the Gromov-Wasserstein loss as an alternative to earth mover distance. Our initialization shares similarities with Artetxe et al. (2018), where they also proposed an alignment based on the Gram matrices of the word vectors. As opposed to our work, they use an heuristic based on sorting the rows of these matrices, while we derive a principled approach from a convex relaxation. Finally, Hartmann et al. (2018) study the impact of the training algorithm of word embeddings, showing that a recent unsupervised alignment method fails when trying to align two English embeddings learned with different algorithms.

Graph matching. The graph matching formulation of Eq. (5) is a quadratic assignment problem, that is NP-Hard (Garey and Johnson, 2002). Polynomial algorithms exist for restricted classes of graphs (Hopcroft and Wong, 1974; Aho and Hopcroft, 1974). Heuristics have been proposed, see e.g. review papers of Conte et al. (2004) and Foggia et al. (2014). Closer to our approach is the convex relaxation to a linear program proposed in Almohamad and Duffuaa (1993) and even more, the convex relaxation proposed by Gold and Rangarajan (1996) to the set of doubly-stochastic matrices. Under certain conditions on the graph structures, the solution of this relaxation is the same as the exact isomorphism (Aflalo et al., 2014). There are no guarantees to converge to a corner of the convex hull, but solutions can be used as an initialization for an inexact or non-convex algorithm (Lyzinski et al., 2016) like the concave minimization problem of Zaslavskiy et al. (2009) or the problem formulated in Eq. (2). There is also a wide literature on the computational hardness of learning problems on graphs (Kučera, 1995; Feldman et al., 2013), and their implications for high-dimensional learning problems (Berthet and Rigollet, 2013; Berthet, 2014; Wang et al., 2016; Baldin and Berthet, 2018; Berthet and Ellenberg, 2019).

Wasserstein distance. As in Bojanowski and Joulin (2017), our formulation in Eq. (2) minimizes the Wasserstein distance (also known as Earth Mover's distance) to estimate the minimal transformation between two distributions of points (Rubner et al., 1998). We refer the reader to Peyré and Cuturi (2017) for an exhaustive survey on the subject. Kusner et al. (2015) use an EMD between word embeddings to compute the distance between documents, and later Rolet et al. (2016) use a smooth Wasserstein distance to align word embeddings distribution and discover cross-language topics. Flamary et al. (2016) optimize a different Wasserstein objective over the Stiefel manifold, for the task of discriminant analysis. The Wasserstein distance has also been used in statistics to give meaningful notions of means for misaligned signals (Panaretos et al., 2016;

	Seed	Data	Window	Algo
Distance	6.090	7.872	11.151	16.008
Relaxation	0.99	0.97	0.17	0.00
Rand. init	0.40	0.22	0.21	0.21
Convex init	1.00	0.99	1.00	0.98

Table 1: We report the accuracy of different methods for words with rank in the range 5,000-10,000. Relaxation indicates the convex formulation applied to the 2k first vectors from each set.

Zemel and Panaretos, 2017). Finally, our approach bears some similarities with the Gromov-Wasserstein that has been successfully used to align shapes in Mémoli (2007); Solomon et al. (2016).

4 Experiments

In this section, we evaluate our method in two settings. First, we perform toy experiments to gain some insight regarding our approach. We then compare it to various state of the art algorithms on the task of unsupervised word translation.

4.1 Toy experiments

Instead of generating purely random datasets, we train various word embeddings, using the same corpus but introducing noise in the training process. More specifically, we train skipgram and cbow models using the `fasttext` tool (Bojanowski et al., 2016) on 100M English tokens from the 2007 News Crawl corpus.[1] We consider the following settings to generate our toy datasets:

- **Seed:** we learn two models on the same data, with the same hyper-parameters. The only difference between the two models is the seed used to initialize the parameters of the models. Please note that during the training of skipgram models, frequent words are randomly subsampled, as well as negative examples. There is thus two sources of randomness between the two models: the initial parameters of the model and the sampling during training.

- **Data:** in that instance, we learn two embeddings using different split of the data. More precisely, we use the first 100M tokens for the first model and the next 100M tokens for the second model. Since both training splits come from the same domain, the models are closed.

- **Window:** in that instance, we learn two skipgram models on the same data, but with different window size. The first model uses a window of size 2, while the second model uses a window of size 10. We also use different seeds, similarly to the first instance.

- **Algorithm:** for the last setting, we consider two word embeddings from different models: skipgram and cbow.

Because all models are trained on English data, we have the ground truth matching between vectors from the two sets. We can thus estimate an orthogonal matrix using Procrustes, and measure the "distance" between the two sets of points. We report these for the different settings in Table 1. For these experiments, we compare the different approaches on the first 10,000 points of our models, and using a batchsize $b = 250$.

First, we compare our stochastic algorithm with random initialization with the same algorithm initialized with the solution of the convex relaxation. We observe that while the random initialization sometimes converges for the easy problems, its rate of success rapidly decreases. On the other hand, when initialized with the solution of the convex formulation, our stochastic algorithm always converges to a good solution. Interestingly, even when the solution of the relaxed problem is not a good solution, it still provides a good initialization for the stochastic gradient method.

4.2 Unsupervised word translation

In our second set of experiments, we evaluate our method on the task of unsupervised word translation. Given word embeddings trained on two monolingual corpora, the goal is to infer a bilingual dictionary by aligning the corresponding word vectors. We use the same exact setting as Conneau et al. (2017). In particular, we use the same evaluation datasets and code, as well as the same word vectors.

Baselines. We compare our method with Procrustes, as well as three unsupervised approaches: the adversarial training (adversarial) of Conneau et al. (2017), the Iterative Closest Point approach (ICP) of Hoshen and Wolf (2018) and Gromov-Wasserstein (GW) of Alvarez-Melis and Jaakkola (2018). All their numbers are taken from their papers.

Implementation details. For CSLS, we use 10 nearest neighbors and for ISF, we use a temperature of 25. Following Xing et al. (2015) and Artetxe et al. (2016), we normalize and center the embeddings for both Procrustes and our approach. The refinement

[1] http://statmt.org/wmt14/translation-task.html

	EN-ES	ES-EN	EN-FR	FR-EN	EN-DE	DE-EN	EN-RU	RU-EN
Procrustes	82.7	$\underline{84.2}$	$\underline{82.7}$	$\underline{83.4}$	74.8	$\underline{73.2}$	$\underline{51.3}$	$\underline{63.7}$
Adversarial*	81.7	83.3	82.3	82.1	74.0	72.2	44.0	59.1
Iterative Closest Point*	82.1	**84.1**	**82.3**	**82.9**	74.7	**73.0**	**47.5**	**61.8**
Gromov-Wasserstein	81.7	80.4	81.3	78.9	71.9	72.8	45.1	43.7
Ours	**$\underline{82.8}$**	84.0	**82.3**	**82.9**	**$\underline{75.6}$**	72.9	45.2	59.8

Table 2: Comparison with supervised and unsupervised state-of-the-art approaches. In underline the best performance, in bold, the best among unsupervised methods. * indicates unnormalized vectors.

Method	EN-ES	ES-EN	EN-FR	FR-EN	EN-DE	DE-EN	EN-RU	RU-EN
Proc. - NN	78.3	80.9	77.0	80.0	70.8	71.8	49.5	64.6
Proc. - CSLS	81.1	$\underline{84.8}$	$\underline{81.6}$	$\underline{84.0}$	$\underline{74.7}$	$\underline{74.4}$	$\underline{52.0}$	$\underline{67.8}$
Proc. - ISF	$\underline{81.4}$	83.4	81.2	82.9	71.8	72.4	50.4	$\underline{67.8}$
Adv. - NN	69.8	71.3	70.4	61.9	63.1	59.6	29.1	41.5
Adv. - CSLS	75.7	79.7	77.8	71.2	**70.1**	**66.4**	37.2	48.1
Ours - NN	77.0	75.6	76.7	74.1	66.1	62.1	33.4	49.6
Ours - CSLS	**80.0**	**81.1**	**80.7**	79.4	69.4	65.1	**37.8**	**53.4**
Ours - ISF	79.6	79.3	79.8	**79.6**	65.8	63.8	37.1	52.0

Table 3: Performance of different supervised and unsupervised initialization. We report results without refinement and with different retrieval criterion. In underline, the best performance, in bold, the best among unsupervised approaches. We normalize and center the word embeddings for Proc. and our approach, while Adv. uses unnormalized vectors.

step is from Conneau et al. (2017) and consists in alternating, for 5 epochs, between building a dictionary of mutual nearest neighbors (using CSLS) and running Procrustes on this dictionary. This procedure is thus ICP with the CSLS criterion. Our convex relaxation is too demanding to run on all the vectors, and thus, we use the 2.5K most frequent words for our initialization.

As noted in section 2, the choice of the batch size is important: a larger batch size will lead to a better approximation of our cost function, but is also more computationally intensive. We thus propose to increase the batch size during the optimization. More precisely, we perform 5 epochs, where we double the batch size at the beginning of each epoch while reducing the number of iterations to keep the computation time constant. The first epoch of our method uses a batch size of 500 and $5,000$ iterations. Finally, we use the Sinkhorn solver of Cuturi (2013) to compute approximate solutions of optimal transport problems, with a regularization parameter of 0.05.

Main results. In order to quantitatively assess the quality of each approach, we consider the problem of bilingual lexicon induction. This can be formulated as a retrieval problem, and following standard practice, we report the precision at one. As observed by Artetxe et al. (2017) and Conneau et al. (2017), refining the alignment significantly improves the performance of unsupervised approaches. We thus report the performance of different state-of-the-art methods in Table 2. For Procustes, we apply a refinement step that signficanlty improves the quality of the alignment. Several reasons can explain this: first the lexicons for supervision are not clean, and the ℓ_2-loss is sensible to outliers. Second the lexicons are small (around 10K) and the mapping is not regularized, besides the orthogonality constraint.

Overall our performance is on par with ICP and significantly better than adversarial training. The fact that ICP and our approach achieve comparable results is not surprising as we are considering a similar loss function. However, ICP requires a lot of random initializations to converge to a good solution, while initializing our approach with the result of the convex relaxation guarantees the same performance with a single run. In practice, this means that our method requires significantly less computational resources to obtain similar results as ICP.

We also report results on the dataset introduced by Dinu et al. (2014) in Table 5. Please note that we use the same setting and hyperparameters as for the experiments performed on the MUSE benchmarks, in order to illustrate the robustness of our method.

	100	200	400	800	1600
Time	1m47s	2m07s	2m54s	5m34s	22m13s
EN-ES	68.5	73.8	74.9	75.0	76.3
EN-FR	67.4	71.9	74.5	75.6	75.7
EN-DE	59.1	63.0	64.4	65.8	66.4
EN-RU	23.7	27.9	29.9	32.3	33.2

Table 4: Influence of the batch size: we report the precision at 1 after 4,000 iterations as a function of the batch size. We use the nearest neighbor (NN) approach to retrieve the translation of a given query.

Method	En-It
Mikolov et al. (2013)	33.8
Dinu et al. (2014)	38.5
Artetxe et al. (2016)	39.3
Smith et al. (2017)	43.1
Conneau et al. (2017)	45.1
Artetxe et al. (2018)	48.1
Alvarez-Melis and Jaakkola (2018)	**49.2**
Ours	45.2

Table 5: Accuracy of our approach, compared to previous work, on the dataset from Dinu et al. (2014). We use the same setting and hyperparameters for this benchmark as for the results reported in Table 2.

Impact of the initialization. Several state-of-the-art approaches relies on a two step procedure: after a good initialization, they iteratively refine the alignment with a ICP approach based on a CSLS criterion. Hoshen and Wolf (2018) uses hundreds of initializations to find a good starting point, while the adversarial training often gives a decent initialization with a single run. We compare the quality of this initialization with the solution of our relaxed method in Table 3. For both NN and CSLS, our approach performs better than the GAN based method of Conneau et al. (2017) on 6 out of 8 pairs of languages while being simpler. However, we are still significantly worse than the supervised approach, by often more than 5%. It is interesting to notice that for both Procrustes and our approach, ISF and CSLS obtain similar performance. Empirically, we found that ISF is more sensitive to its free parameter. In particular, the temperature in ISF should vary with the number of points, and the fact that a single value works across all these datasets is potentially an indirect consequence of the evaluation, that is to restrict the size of the sets to 200K words. We thus propose to use CSLS in most of our experiments.

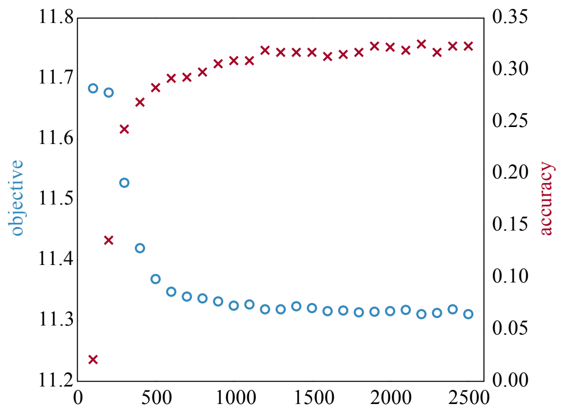

Figure 2: Accuracy and objective function value on the Muse English-Russian benchmark during stochastic optimization. We used a batchsize of 2000 and artificially low learning rate for illustration purposes. We observe that our objective function is well correlated with the test accuracy.

Impact of batch size on performance. The batch size b plays an important role on our loss as it trades off speed for distance to the original formulation. Table 4 shows its influence on the performance as well as running time. As expected, our model converges faster for small b since we run the Sinkhorn algorithm on $b \times b$ matrices, leading to a complexity of $\mathcal{O}(b^2)$ up to logarithmic terms (Altschuler et al., 2017). The larger b is, the better the approximation of the squared Wasserstein distance is, and the closer we are to the real loss. Despite saying nothing about the quality of the local minima, it is interesting to see that in practice this converts into better performance as well.

5 Conclusion

This paper presents an approach to align embeddings in high dimensional space. While the overall problem is non-convex and computationally expensive, we present an efficient stochastic algorithm to solve the problem. We also develop a convex relaxation that can be used to initialize our approach. We validate our method on a few toy examples and a real application, namely unsupervised word translation, where we achieve performances that are on par with the state-of-the-art. A few questions remain, in particular the link between graph matching and point cloud alignments can be further investigated. It should be possible to identify the type of problems where our approach is guaranteed to work and where it will provably fail. Finally, we think that it is possible to improve the relaxation procedure.

References

Absil, P.-A., Mahony, R., and Sepulchre, R. (2009). *Optimization algorithms on matrix manifolds*. Princeton University Press.

Aflalo, Y., Bronstein, A., and Kimmel, R. (2014). Graph matching: relax or not? *arXiv preprint arXiv:1401.7623*.

Aho, A. V. and Hopcroft, J. E. (1974). *The design and analysis of computer algorithms*. Pearson Education India.

Almohamad, H. and Duffuaa, S. O. (1993). A linear programming approach for the weighted graph matching problem. *IEEE Transactions on pattern analysis and machine intelligence*, 15(5):522–525.

Altschuler, J., Weed, J., and Rigollet, P. (2017). Near-linear time approximation algorithms for optimal transport via sinkhorn iteration.

Alvarez-Melis, D. and Jaakkola, T. (2018). Gromov-wasserstein alignment of word embedding spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Alvarez-Melis, D., Jegelka, S., and Jaakkola, T. S. (2018). Towards optimal transport with global invariances. *arXiv preprint arXiv:1806.09277*.

Artetxe, M., Labaka, G., and Agirre, E. (2016). Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294.

Artetxe, M., Labaka, G., and Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 451–462.

Artetxe, M., Labaka, G., and Agirre, E. (2018). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. *arXiv preprint arXiv:1805.06297*.

Aucouturier, J.-J. and Pachet, F. (2008). A scale-free distribution of false positives for a large class of audio similarity measures. *Pattern recognition*, 41(1):272–284.

Baldin, N. and Berthet, Q. (2018). Optimal link prediction with matrix logistic regression. *Preprint*.

Bernton, E., Jacob, P. E., Gerber, M., and Robert, C. P. (2017). Inference in generative models using the wasserstein distance.

Berthet, Q. (2014). Optimal testing for planted satisfiability problems. *Electron. J. Stat.*

Berthet, Q. and Ellenberg, J. (2019). Detection of planted solutions for flat satisfiability problems. *AIStats 2019*.

Berthet, Q. and Rigollet, P. (2013). Complexity theoretic lower bounds for sparse principal component detection. *J. Mach. Learn. Res. (COLT)*, 30.

Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Bojanowski, P. and Joulin, A. (2017). Unsupervised learning by predicting noise. In *International Conference on Machine Learning*, pages 517–526.

Cao, H., Zhao, T., Zhang, S., and Meng, Y. (2016). A distribution-based model to learn bilingual word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1818–1827.

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. (2017). Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Conte, D., Foggia, P., Sansone, C., and Vento, M. (2004). Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, 18(03):265–298.

Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1995). Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300.

Dinu, G., Lazaridou, A., and Baroni, M. (2014). Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*.

Doddington, G., Liggett, W., Martin, A., Przybocki, M., and Reynolds, D. (1998). Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the nist 1998 speaker recognition evaluation. Technical report.

Dudley, R. (1969). The speed of mean glivenko-cantelli convergence. *The Annals of Mathematical Statistics*, 40(1):40–50.

Feldman, V., Grigorescu, E., Reyzin, L., Vempala, S., and Xiao, Y. (2013). Statistical algorithms and a lower bound for planted clique. In *Proceedings of the*

Fourty-Fifth Annual ACM Symposium on Theory of Computing, STOC 2013.

Fischler, M. A. and Bolles, R. C. (1987). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in computer vision*, pages 726–740. Elsevier.

Flamary, R., Cuturi, M., Courty, N., and Rakotomamonjy, A. (2016). Wasserstein discriminant analysis.

Foggia, P., Percannella, G., and Vento, M. (2014). Graph matching and learning in pattern recognition in the last 10 years. *International Journal of Pattern Recognition and Artificial Intelligence*, 28(01):1450001.

Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics (NRL)*, 3(1-2):95–110.

Fung, P. (1995). Compiling bilingual lexicon entries from a non-parallel english-chinese corpus. In *Third Workshop on Very Large Corpora.*

Garey, M. R. and Johnson, D. S. (2002). *Computers and intractability*, volume 29. wh freeman New York.

Genevay, A., Peyré, G., and Cuturi, M. (2018). Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 1608–1617.

Gold, S. and Rangarajan, A. (1996). A graduated assignment algorithm for graph matching. *IEEE Transactions on pattern analysis and machine intelligence*, 18(4):377–388.

Goodall, C. (1991). Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 285–339.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Hartmann, M., Kementchedjhieva, Y., and Søgaard, A. (2018). Why is unsupervised alignment of english embeddings from different algorithms so hard? In *EMNLP.*

Hopcroft, J. E. and Wong, J.-K. (1974). Linear time algorithm for isomorphism of planar graphs (preliminary report). In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 172–184. ACM.

Hoshen, Y. and Wolf, L. (2018). An iterative closest point method for unsupervised word translation. *arXiv preprint arXiv:1801.06126.*

Jegou, H., Schmid, C., Harzallah, H., and Verbeek, J. (2010). Accurate image search using the contextual dissimilarity measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):2–11.

Kučera, L. (1995). Expected complexity of graph partitioning problems. *Discrete Appl. Math.*, 57(2-3):193–212.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2):83–97.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966.

Leordeanu, M. and Hebert, M. (2005). A spectral technique for correspondence problems using pairwise constraints. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1482–1489. IEEE.

Liu, C., Yuen, J., Torralba, A., Sivic, J., and Freeman, W. T. (2008). Sift flow: Dense correspondence across different scenes. In *European conference on computer vision*, pages 28–42. Springer.

Lyzinski, V., Fishkind, D. E., Fiori, M., Vogelstein, J. T., Priebe, C. E., and Sapiro, G. (2016). Graph matching: Relax at your own risk. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):60–73.

Mémoli, F. (2007). On the use of gromov-hausdorff distances for shape comparison.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168.*

Pachet, F. and Aucouturier, J.-J. (2004). Improving timbre similarity: How high is the sky. *Journal of negative results in speech and audio sciences*, 1(1):1–13.

Panaretos, V. M., Zemel, Y., et al. (2016). Amplitude and phase variation of point processes. *The Annals of Statistics*, 44(2):771–812.

Peyré, G. and Cuturi, M. (2017). *Computational Optimal Transport.*

Radovanović, M., Nanopoulos, A., and Ivanović, M. (2010). Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(Sep):2487–2531.

Rangarajan, A., Chui, H., and Bookstein, F. L. (1997). The softassign procrustes matching algorithm. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 29–42. Springer.

Rapp, R. (1995). Identifying word translations in non-parallel texts. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 320–322. Association for Computational Linguistics.

Rolet, A., Cuturi, M., and Peyré, G. (2016). Fast dictionary learning with a smoothed wasserstein loss. In *Artificial Intelligence and Statistics*, pages 630–638.

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

Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Smith, S. L., Turban, D. H., Hamblin, S., and Hammerla, N. Y. (2017). Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv preprint arXiv:1702.03859*.

Solomon, J., Peyré, G., Kim, V. G., and Sra, S. (2016). Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics (TOG)*, 35(4):72.

Sommerfeld, M., Schrieber, J., and Munk, A. (2018). Optimal transport: Fast probabilistic approximation with exact solvers.

Tameling, C., Sommerfeld, M., and Munk, A. (2017). Empirical optimal transport on countable metric spaces: Distributional limits and statistical applications.

Tomasi, C. and Kanade, T. (1992). Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154.

Wang, T., Berthet, Q., and Plan, Y. (2016). Average-case hardness of rip certification. *NIPS*.

Weed, J. and Bach, F. (2017). Sharp asymptotic and finite-sample rates of convergence of empirical measures in wasserstein distance. *arXiv preprint arXiv:1707.00087*.

Weed, J. and Berthet, Q. (2019). Estimation of smooth densities in wasserstein metric. *Preprint*.

Xing, C., Wang, D., Liu, C., and Lin, Y. (2015). Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011.

Xu, R., Yang, Y., Otani, N., and Wu, Y. (2018). Unsupervised cross-lingual transfer of word embedding spaces. In *EMNLP*.

Zaslavskiy, M., Bach, F., and Vert, J.-P. (2009). A path following algorithm for the graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2227–2242.

Zemel, Y. and Panaretos, V. M. (2017). Fréchet means and procrustes analysis in wasserstein space.

Zhang, M., Liu, Y., Luan, H., and Sun, M. (2017a). Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1959–1970.

Zhang, M., Liu, Y., Luan, H., and Sun, M. (2017b). Earth mover's distance minimization for unsupervised bilingual lexicon induction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1934–1945.