Gromov-Wasserstein Averaging of Kernel and Distance Matrices
Gabriel Peyré, Marco Cuturi, Justin Solomon

To cite this version:


HAL Id: hal-01322992
https://hal.archives-ouvertes.fr/hal-01322992
Submitted on 3 Jun 2016

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

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

This paper presents a new technique for computing the barycenter of a set of distance or kernel matrices. These matrices, which define the interrelationships between points sampled from individual domains, are not required to have the same size or to be in row-by-row correspondence. We compare these matrices using the softassign criterion, which measures the minimum distortion induced by a probabilistic map from the rows of one similarity matrix to the rows of another; this criterion amounts to a regularized version of the Gromov-Wasserstein (GW) distance between metric-measure spaces. The barycenter is then defined as a Fréchet mean of the input matrices with respect to this criterion, minimizing a weighted sum of softassign values. We provide a fast iterative algorithm for the resulting nonconvex optimization problem, built upon state-of-the-art tools for regularized optimal transportation. We demonstrate its application to the computation of shape barycenters and to the prediction of energy levels from molecular configurations in quantum chemistry.

1. Introduction

Many classes of input data encountered in machine learning are best expressed using either pairwise distance matrices or kernels. For instance, the atoms making up proteins and other molecules have no canonical orientation and hence are only known up to rigid motion; so, the geometry of a molecule might be better described by the pairwise distances between atoms rather than by their absolute positions. Kernel and covariance matrices from different datasets exhibit similar structure, since they define data points by their roles relative to the rest of a dataset rather than in any absolute sense.

A major difficulty with this representation is that in many applications these matrices are not “registered” or “aligned,” meaning that there is no explicit correspondence between their rows and columns. This is the case for shapes that undergo pose variation, articulation or deformation. Even worse, in some cases, similarities are not defined over the same ground space. For example, different molecules may have varying numbers of atoms. This inconsistency yields matrices of varying dimensions.

In this paper, we propose a theoretical and computational framework for summarizing collections of unaligned distance or kernel matrices. Building upon the notion of Gromov-Wasserstein (GW) distances between metric-measure spaces (Mémoli, 2007), we provide machinery for registering matrices of varying sizes while building interpolants and barycenters inheriting structure from the inputs. We also derive a fast regularized approximation scheme providing a practical and efficient way to recover these barycenters in practice.

1.1. Previous Work

Countless techniques leverage matrices of distances or kernels to describe a dataset. While distances and kernels have opposite ordering—a large distance indicates small similarity—unless it is necessary to distinguish we will use the term “similarity matrix” to refer to any matrix containing pairwise relationships.

Aligning similarity matrices. Comparison of similarity matrices is challenging due to a lack of alignment. Mathematically, they only define the intrinsic structure
of a dataset, which remains unchanged e.g. under rotation, translation, and isometric motion. More concretely, there is unlikely to be a canonical ordering of the rows or columns of similarity matrices, and removing or adding a few rows/columns may not significantly affect the structure. Automatic computation of a map aligning two similarity matrices is itself a challenging problem; our work is fairly unique in that we simultaneously compute such an alignment while solving an optimization problem over such matrices.

Matching rows of similarity matrices usually becomes a quadratic assignment problem (Loiola et al., 2007), which is NP-hard. For the specific case of aligning similarity matrices derived from pairwise distances in a metric space (e.g., geodesic distances along a surface), a well-known instance of this matching problem is computation of the Gromov-Hausdorff distance (Gromov, 2001), which was applied to matching in (Mémoli, 2007).

To relax the matching problem to a continuous—but still non-convex—optimization problem, the seminal work of Mémoli considers matching between metric-measure spaces, incorporating a probability distribution on the ground space that can account for some form of uncertainty (Mémoli, 2011). The geodesic properties of this “space of metric spaces” are considered in (Sturm, 2012).

The recent work (Hendrikson, 2016) showcases the application of the corresponding Gromov-Wasserstein distance to clustering of biological and social networks. It is possible to further relax this formulation to a quadratic (Aflalo et al., 2015) or semidefinite (Kezurer et al., 2015) convex program, although this relaxation can fail if the input similarity matrices exhibit symmetries.

Rather than mixing the possible symmetric maps, we embrace the nonconvexity of the original problem and optimize quadratic matching objectives directly. After regularization, our matching subroutine resembles “softassign quadratic assignment” (Rangarajan et al., 1999; Gold & Rangarajan, 1996). We extend their iterations to general loss functions and to more advanced machine learning problems than nearest-neighbor search, in particular barycenter computation. We additionally leverage the fact that GW can compare arbitrary matrices, not just pairwise distances, making our machinery applicable to a broader range of tasks. The mapping component of our algorithm can be considered a generalized version of the method recently proposed in (Solomon et al., 2016).

**Averaging similarity matrices.** The design of an “average” or barycenter of similarity matrices is informed by two primary factors: A measure of discrepancy between similarity matrices and a definition of a mean. We will GW distances to define discrepancies, so our remaining task is to define what it means to compute a barycenter.

Even when similarity matrices are the same size and registered, it may not make sense to average them arithmetically. One natural way to define a mean or barycenter of inputs \((C_s)\) is to minimize the sum \(\sum_s \lambda_s d^2(C, C_s)\) over all possible \(C\), where \(d\) is some notion of distance. This generalized notion of a mean is known as the *Karcher* or *Fréchet* mean, which generalizes the notion of an average to a wide class of metric spaces (Nielsen & Bhatia, 2012).

Fréchet barycenters in Wasserstein space were proposed in (Agueh & Carlier, 2011). Several algorithms for their computation have been proposed including (Cuturi & Doucet, 2014; Benamou et al., 2015); these assume a fixed ground distance metric. Our method, however, finds more commonality with algorithms that compute Fréchet means in the space of covariance or metric matrices. (Dryden et al., 2009) considers several simple options for covariance matrices, mostly with fixed alignment; their “Procrustes” mean provides one rudimentary strategy for alignment. (Sra, 2011) defines a metric on registered positive definite kernel matrices that can be used for computing Fréchet means.

Barycenter computation is a building block for many learning methods. For instance, \(k\)-means clustering alternates between distances and barycenter computation. The nearest-centroid classifier provides a similar approach to classification (Manning et al., 2008).

**Optimal transport (OT).** OT (Villani, 2003) is a way to compare probability distributions (histograms in the finite-dimensional case) defined over either the same ground space or multiple pre-registered ground spaces. The means of comparison is a convex linear program optimizing for a matching that moves the mass from one distribution to the other with minimal cost. OT has been applied to machine learning problems including comparison of descriptors (Cuturi, 2013), \(k\)-means (Cuturi & Doucet, 2014), semi-supervised learning (Solomon et al., 2014), domain adaptation (Courty et al., 2014), designing loss functions (Frogner et al., 2015), low-rank approximation (Seguy & Cuturi, 2015), and dictionary learning (Rolet et al., 2016).

As highlighted above, GW-style matching extends OT to the case when the ground spaces are not pre-registered, yielding a non-convex quadratic program to compute the transport. The resulting transportation matrix can be understood as a soft registration from one domain to the other. Our algorithm solves an entropically-regularized version of this quadratic program by extending recent advances in the computation of OT (Cuturi, 2013; Benamou et al., 2015).
1.2. Contributions

Our first contribution is the definition of a new discrepancy between similarity matrices. It extends the “Gromov-Wasserstein” distance between metric-measure spaces to arbitrary matrices, using a generic loss functions to compare pairwise similarities and entropic regularization. It can be defined over different ground measured spaces (i.e. each point is equipped with a measure), which are not required to be registered a priori. Entropic regularization enables the design of a fast iterative algorithm to compute a stationary point of the objective function defining the discrepancy.

Our second contribution is a new notion of the barycenter of a set of unregistered similarity matrices, defined as a Fréchet mean with respect to the GW discrepancy. We also exemplify our barycenter.

We showcase applications of our method to the computation of barycenters between shapes. We also exemplify how the GW discrepancy can be used to predict energy levels in quantum chemistry, where molecules are naturally represented using their Coulomb interaction matrices, a perfect fit for our unregistered dissimilarity matrix formalism.

The code to reproduce the results of this paper is available online.¹

1.3. Notation

The simplex of histograms with N bins is \( \Sigma_N = \{ p \in \mathbb{R}_{+}^{N} : \sum_i p_i = 1 \} \). The entropy of \( T \in \mathbb{R}_{+}^{N \times N} \) is defined as \( H(T) = - \sum_{i,j=1}^{N} T_{i,j} (\log(T_{i,j}) - 1) \). The set of couplings between histograms \( p \in \Sigma_{N_1} \) and \( q \in \Sigma_{N_2} \) is

\[
\mathcal{C}_{p,q} \defeq \{ T \in (\mathbb{R}_{+})^{N_1 \times N_2} : T \mathds{1}_{N_2} = p, T^\top \mathds{1}_{N_1} = q \}.
\]

Here, \( \mathds{1}_{N} = (1, \ldots, 1)^\top \in \mathbb{R}^{N} \). For any tensor \( L = (L_{i,j,k,l})_{i,j,k,l} \) and matrix \( (T_{i,j})_{i,j} \), we define the tensor-matrix multiplication as

\[
L \otimes T \defeq \left( \sum_{k,l} L_{i,j,k,l} T_{k,l} \right)_{i,j}.
\]

2. Gromov-Wasserstein Discrepancy

2.1. Entropic Optimal Transport

Optimal transport distances are useful to compare two histograms \( (p,q) \in \Sigma_{N_1} \times \Sigma_{N_2} \) defined on the same metric space, or at least on spaces that have previously registered. Given some cost matrix \( c \in \mathbb{R}^{N_1 \times N_2}_+ \), where \( c_{i,j} \) represents the transportation cost between position indexed by \( i \) and \( j \), we define the solution of entropically-regularized optimal transport between these two histograms as

\[
T(c, p, q) \defeq \arg \min_{T \in \mathcal{C}_{p,q}} \langle c, T \rangle - \varepsilon H(T),
\]

which is a strictly convex optimization problem.

As shown in (Cuturi, 2013), the solution reads \( T(c, p, q) = \text{diag}(a) K \text{diag}(b) \) where \( K \defeq e^{-\varepsilon} \in \mathbb{R}^{N_1 \times N_2}_+ \) is the so-called Gibbs kernel associated to \( c \), and \((a, b) \in \mathbb{R}^{N_1}_+ \times \mathbb{R}^{N_2}_+ \) can be computed using Sinkhorn iterations

\[
a \leftarrow \frac{p}{K b} \quad \text{and} \quad b \leftarrow \frac{q}{K^\top a},
\]

where here \( \cdot \) denotes component-wise division.

2.2. Gromov-Wasserstein Discrepancy

Following the pioneering work of Mémoli, we consider input data expressed as metric-measure spaces (Mémoli, 2011). This corresponds to pairs of the form \((C, p) \in \mathbb{R}^{N \times N} \times \Sigma_{N} \), where \( N \) is an arbitrary integer (the number of elements in the underlying space). Here, \( C \) is a matrix representing either similarities or distances between these elements, and \( p \) is an histogram, which can account either for some uncertainty or relative importance between these elements. In case no prior information is known about a space, one can set \( p = \mathds{1}_{N} \) to the uniform distribution. In our setting, since we target a wide range of machine-learning problems, we do not restrict the matrices \( C \) to be distance matrices, i.e., they are not necessarily positive and does not necessarily satisfy the triangle inequality.

We define the Gromov-Wasserstein discrepancy between two measured similarity matrices \((C, p) \in \mathbb{R}^{N_1 \times N_1} \times \Sigma_{N_1} \) and \((\bar{C}, q) \in \mathbb{R}^{N_2 \times N_2} \times \Sigma_{N_2} \) as follows:

\[
GW(C, \bar{C}, p, q) \defeq \min_{T \in \mathcal{C}_{p,q}} \mathcal{E}_{C, \bar{C}}(T)
\]

where

\[
\mathcal{E}_{C, \bar{C}}(T) \defeq \sum_{i,j,k,l} L(C_{i,k,l}, \bar{C}_{j,k,l}) T_{i,j} T_{k,l},
\]

The matrix \( T \) is a coupling between the two spaces on which the similarity matrices \( C \) and \( \bar{C} \) are defined. Here \( L \) is some loss function to account for the misfit between the similarity matrices. Typical choices of \( L \) include the quadratic loss \( L(a, b) = L_2(a, b) \defeq \frac{1}{2} |a - b|^2 \) and the Kullback-Leibler divergence \( L(a, b) = \text{KL}(a | b) \defeq a \log(a/b) - a + b \) (which is not symmetric). This definition (4) of GW extends slightly the one considered by (Mémoli, 2011), since we consider an arbitrary loss \( L \) (rather than just the \( L^2 \) squared loss). In the case
which is equal to

\[ T \],

where \( \in \mathbb{C} \) is the tensor-matrix multiplication (1).

The following proposition shows how to compute \( \mathcal{L}(C, \bar{C}) \) efficiently for a general class of loss functions:

**Proposition 1.** If the loss can be written as

\[ L(a, b) = f_1(a) + f_2(b) - h_1(a)h_2(b) \quad (5) \]

for functions \( (f_1, f_2, h_1, h_2) \), then, for any \( T \in \mathbb{C}_{p,q} \),

\[ \mathcal{L}(C, \bar{C}) \odot T = c_{C, \bar{C}} - h_1(C)Th_2(\bar{C})^\top \quad (6) \]

where

\[ c_{C, \bar{C}} = f_1(C)p\mathbb{1}_{N_2} + N_1q^\top f_2(\bar{C})^\top \text{ is independent of } T. \]

**Proof.** Under hypothesis (5), formula (1) shows that one has the decomposition \( \mathcal{L}(C, \bar{C}) \odot T = A + B + C \) where

\[
\begin{align*}
A_{i,j} &= \sum_{k} f_1(C_{i,k}) \sum_{\ell} T_{k,\ell} = (f_1(C)(T\mathbb{1}))_{i,j}, \\
B_{i,j} &= \sum_{k} f_2(\bar{C}_{j,k}) \sum_{\ell} T_{k,\ell} = (f_2(\bar{C})(T^\top \mathbb{1}))_{i,j}, \\
C_{i,j} &= \sum_{k} h_1(C_{i,k}) \sum_{\ell} h_2(\bar{C}_{j,\ell}) T_{k,\ell}
\end{align*}
\]

which is equal to \((h_1(C)(h_1(\bar{C})T^\top)^\top)_{i,j}\). \( \square \)

**Remark 1 (Computational complexity).** Formula (6) shows that for this class of losses, one can compute \( \mathcal{L}(C, \bar{C}) \odot T \) efficiently in \( O(N_1^2 N_2 + N_2^2 N_1) \) operations, using only matrix/matrix multiplications, instead of the \( O(N_1^2 N_2^2) \) complexity of the naive implementation of formula (1).

**Remark 2 (Special cases).** The square loss \( L = L_2 \) satisfies (5) for \( f_1(a) = a^2, f_2(b) = b^2, h_1(a) = a \) and \( h_2(\bar{b}) = 2\bar{b} \). The KL loss \( L = KL \) satisfies (5) for \( f_1(a) = a \log(a) - a, f_2(b) = b, h_1(a) = a \) and \( h_2(\bar{b}) = \log(b) \).

### 2.3. Entropic Gromov-Wasserstein Discrepancy

We consider the following entropic approximation of the initial GW formulation (4)

\[
GW_\varepsilon(C, \bar{C}, p, q) \overset{\text{def.}}{=} \min_{T \in \mathbb{C}_{p,q}} E_{C, \bar{C}}(T) - \varepsilon H(T). \quad (7)
\]

This is a non-convex optimization problem. We propose to use projected gradient descent, where both the gradient step and the projection are computed according to the KL metric.

Iterations of this algorithm are given by

\[
T \leftarrow \text{Proj}_{\mathbb{C}_{p,q}}^{KL} \left( T \odot e^{-\tau(\nabla E_{C, \bar{C}}(T) - \varepsilon \nabla H(T))} \right), \quad (8)
\]

where \( \tau > 0 \) is a small enough step size, and the KL projector of any matrix \( K \) is

\[
\text{Proj}_{\mathbb{C}_{p,q}}^{KL}(K) \overset{\text{def.}}{=} \arg\min_{T \in \mathbb{C}_{p,q}} \text{KL}(T|K). \quad (9)
\]

**Proposition 2.** In the special case \( \tau = 1/\varepsilon \), iteration (8) reads

\[
T \leftarrow T \odot (\mathcal{L}(C, \bar{C}) \odot T, p, q). \quad (9)
\]

**Proof.** As shown in (Benamou et al., 2015), the projection is nothing else than the solution to the regularized transport problem (2), hence

\[
\text{Proj}_{\mathbb{C}_{p,q}}^{KL}(K) = T(-\varepsilon \log(K), p, q).
\]

One also has

\[
\nabla E_{C, \bar{C}}(T) - \varepsilon \nabla H(T) = \mathcal{L}(C, \bar{C}) \odot T + \varepsilon \log(T).
\]

Re-arranging the terms in (8), one obtains, in the special case \( \tau \varepsilon = 1 \), the desired formula. \( \square \)

Iteration (9) defines a surprisingly simple algorithm, in which each update of \( T \) involves a Sinkhorn projection.

**Remark 3 (Convergence).** Using (Bot et al., 2015) (see Theorem 12), iterations (8) are guaranteed to converge provided that \( \tau \) is chosen small enough, \( \tau < \tau_{\text{max}} \). Unfortunately, in general, one does not have \( \tau_{\text{max}} > 1/\varepsilon \), so that the step size \( \tau = 1/\varepsilon \) advocated by Proposition 2 is not covered by the theory. However, we found that using \( \tau = 1/\varepsilon \) always leads to a converging sequence of \( T \), and that this works well in practice. Note that when \( L = L_2 \), this recovers the “softassign quadratic assignment” algorithms defined in (Rangarajan et al., 1999; Gold & Rangarajan, 1996). The convergence proof (Rangarajan et al., 1999), however, only ensures convergence of the functional values (not of the iterates) and also only applies when the function being minimized in (7) is convex, which is not the case for arbitrary matrices \((C, \bar{C})\).

### 3. Gromov-Wasserstein Barycenters

#### 3.1. Gromov-Wasserstein Barycenters

We define Gromov-Wasserstein barycenters of measured similarity matrices \((C_s)_{s=1}^S\), where \( C_s \in \mathbb{R}^{N_s \times N_s} \), using
Gromov-Wasserstein Averaging of Kernel and Distance Matrices

Minimization with respect to $C$. The optimization problem (10) over $(T_s)_s$ alone decouples as $S$ independent GW$_C$ optimizations

$$
\forall s, \min_{T_s \in C_{p,p_s}} E_{C,C_s}(T_s) - \varepsilon H(T_s).
$$

A stationary point of this optimization problem can be computed using the optimization algorithm detailed in Section 2.3.

Minimization with respect to $C$. For given $(T_s)_s$, the minimization with respect to $C$ reads

$$
\min_C \sum_s \lambda_s (L(C,C_s) \otimes T, T).
$$

The following proposition shows, for a large class of losses, how to compute the global minimizer in closed form.

**Proposition 3.** If $L$ satisfies (5) and $f'_1/h'_1$ is invertible, then the solution to (12) reads

$$
C = \left( \frac{f'_1}{h'_1} \right)^{-1} \left( \sum_s \lambda_s T^\top_s h_2(C_s) T_s \right) \left( \frac{pp^\top}{pp^\top} \right),
$$

where we assume the normalization $\sum_s \lambda_s = 1$.

**Proof.** Using relation (6), the functional to be minimized reads

$$
\sum_s \lambda_s (f_1(C)p1^\top + 1p^\top_s f_2(C_s) - h_1(C)T_s h_2(C_s)^\top, T_s).
$$

The first order optimality condition for this optimization problem thus reads $f'_1(C) \odot (pp^\top) = h'_1(C) \odot \sum_s \lambda_s T_s h_2(C_s) T^\top_s$, which gives the desired formula.

The intuition underlying formula (13) is clear: Each $T^\top_s h_2(C_s) T_s$ is a “realigned” matrix where $T_s$ acts as a fuzzy permutation (optimal transportation coupling) of both rows and columns of the distance matrix $C_s$. These realigned metric are then averaged, where the precise notion of “averaging” depends on the loss $L$.

**Barycenters of PSD kernels using $L^2$ loss.** For the square loss $L = L_2$, the update (13) becomes

$$
C \leftarrow \frac{1}{pp^\top} \sum_s \lambda_s T^\top_s C_s T_s.
$$

This formula highlights an important property of the method:

**Proposition 4.** For $L = L_2$, if the $(C_s)_s$ are positive semidefinite (PSD) matrices, the iterates $C$ produced by the algorithm are also PSD.

**Proof.** Formula (14) shows that the update of $C$ corresponds to a linear averaging of the matrices $(\text{diag}(1/p) T^\top_s C_s T_s \text{diag}(1/p))_s$, which are all PSD since the matrices $(C_s)_s$ are.

This proposition implies, since the SDP cone is a closed set, that the output of our barycenter algorithm at convergence is an SDP kernel.

**Barycenters of infinitely divisible kernels using KL loss.** For the KL loss $L = KL$, the update (13) similar, but with a geometric mean in place of an arithmetic mean

$$
C \leftarrow \exp \left( \frac{1}{pp^\top} \sum_s \lambda_s T^\top_s \log(C_s) T_s \right).
$$

The following proposition shows that this loss is particularly attractive for averaging infinitely divisible kernels. Recall that $C \in \mathbb{R}^{N \times N}$ is infinitely divisible if and only if $U = \log(C)$ is a conditionally positive semidefinite kernel, i.e. for all $x \in \mathbb{R}^N$ such that $\langle x, 1_N \rangle = 0$, then $\langle Ux, x \rangle \geq 0$.

**Proposition 5.** For $L = KL$, if the $(C_s)_s$ are infinitely divisible kernels, the iterates $C$ produced by the algorithm are also infinitely divisible.
Algorithm 1 Computation of GW$_{e}$ barycenters.

Input: $(C_{s}, p_{s})_{s}, p$

Initialize $C$.

repeat

// minimize over $(T_{s})_{s}$

for $s = 1$ to $S$

Initialize $T_{s}$.

repeat

// compute $c_{s} = L(C, C_{s}) \otimes T_{s}$ using (6).

$c_{s} \leftarrow f_{1}(C) + f_{2}(C_{s})^{T} - h_{1}(C)T_{s}h_{2}(C_{s})^{T}$

// Sinkhorn iterations (3) to compute $T(c_{s}, p, q)$

Initialize $a \leftarrow 1$, set $K \leftarrow e^{-c_{s}/\varepsilon}$.

repeat

$a \leftarrow \frac{a}{K_{b}}, b \leftarrow \frac{b}{K^{T}a}$.

until convergence

Update $T_{s} \leftarrow \text{diag}(a)K\text{diag}(b)$.

until convergence

// minimize over $C$ using (13).

$C \leftarrow \left( \sum_{s} \lambda_{s} T_{s}^{\top} h_{2}(C_{s})T_{s}^{\top} \right)^{-1}$

until convergence

\textbf{Proof.} According to formula (15), one has to show that
\[ U \overset{\text{def.}}{=} \sum_{s} \lambda_{s} \text{diag}(1/p)T_{s}^{\top} U_{s}T_{s} \text{diag}(1/p) \] is conditionally PSD proved that the $U_{s}$ matrices are. This is indeed the case, because, for any $x$ such that $\langle x, 1_{N} \rangle = 0$, one has $\langle Ux, x \rangle = \sum_{s} \lambda_{s} \langle U_{s}x_{s}, x_{s} \rangle$, where $x_{s} \overset{\text{def.}}{=} T_{s}x_{s}^{p}$, and since $\langle x_{s}, 1_{N_{s}} \rangle = \langle \xi_{s}, T_{s}^{\top} 1_{N_{s}} \rangle = \langle \xi_{s}, p \rangle = 0$, one has $\langle U_{s}x_{s}, x_{s} \rangle \geq 0$ so that $\langle Ux, x \rangle \geq 0$, which proves the result. \hfill \qed

This proposition implies, since the cone of infinitely divisible kernels is a closed set, that the output of our barycenter algorithm at convergence is infinitely divisible.

A important example of infinitely divisible kernels is given by the form $C = e^{-D^{2}/2\sigma}$, where $D_{i,j} = \|x_{i} - x_{j}\|$ is the Euclidean distance matrix between points $(x_{i})_{i}$, and $\sigma > 0$ is a bandwidth parameter that controls the “locality” of the resulting kernel. One verifies that $\text{KL}(e^{-D^{2}/2\sigma^{2}}|e^{-\sigma^{2}/2}) = \frac{1}{2\sigma^{2}}(D^{2} - \sigma^{2}) + o(1/\sigma^{4})$, so that in the regime of large bandwidth $\sigma \rightarrow +\infty$, using the KL loss in conjunction with this class of kernels becomes equivalent to using the $L^{2}$ loss between squared Euclidean matrices. A similar result in fact holds for any smooth Bregman divergence loss.

\textbf{Pseudocode.} Algorithm 1 details the steps of the optimization technique. Note that it makes use of three nested iterations: (i) blockwise coordinate descent on $(T_{s})_{s}$ and $C$, (ii) projected gradient descent on each $T_{s}$, and (iii) Sinkhorn iterations to compute the projection.

4. Experiments

4.1. Point Clouds

\textbf{Embedded barycenters.} Figure 1 provides an example illustrating the behavior of our GW barycenter approximation. In this experiment, we extract 500 point clouds of handwritten digits from the dataset (LeCun et al., 1998), rotated arbitrarily in the plane. We represent each digit as a symmetric Euclidean distance matrix and optimize for a 500 \times 500 barycenter using Algorithm 1 (uniform weights, $\varepsilon = 1 \times 10^{-3})$; notice that most of the input point clouds consist of fewer than 500 points. We then visualize the barycenter matrix as a point cloud in the plane using multidimensional scaling (MDS). Each digit is colored by transferring RGB values from the barycenter point cloud using the computed map $T$.

This experiment illustrates a few properties of our algorithm. Most prominently, note that the input digits are \textit{rotated arbitrarily}, which—unlike tools based on classical optimal transportation—does not affect the computation of the barycenter. Also, to avoid bias in this experiment we initialize $C$ as a random matrix, and yet our nonconvex optimization algorithm still reaches a meaningful barycenter.

Figure 2 illustrates a similar experiment on the 2D shape data from (Thakoor et al., 2007). This second experiment illustrates some additional properties of our barycenters. Interestingly, even though we do not impose curve topol-
For clustering finds meaningful clusters of handwritten digits. Even without supervision, this rudimentary technique is applicable to the number of data points assigned to each cluster. The visualization technique is shown above the table. Color is proportional to the number of data points assigned to each cluster. Even without supervision, this rudimentary technique for clustering finds meaningful clusters of handwritten digits from the dataset. All the “1” digits are clustered correctly. Digits 2 and 4 are mixed considerably in the computed clustering, reflecting the wide variety of handwritten styles for these two digits.

**Regularization.** Figure below illustrates the dependence of our barycenter on the regularizer $\varepsilon$.

\[
\varepsilon = 1.5 \times 10^{-3} \quad \varepsilon = 4 \times 10^{-3} \quad \varepsilon = 8 \times 10^{-3} \quad \varepsilon = 1.6 \times 10^{-2} \quad \varepsilon = 3.2 \times 10^{-2}
\]

It shows the barycenter from a test in Figure 2 for increasing values of $\varepsilon$. When $\varepsilon$ is small, the embedded barycenter most closely resembles the input data. As $\varepsilon$ increases, the barycenters become smoother, with the advantage of higher resilience to noise and fewer iterations needed for optimization.

**Clustering.** The figure on the right illustrates the incorporation of our barycenter technique into a larger machine learning pipeline. It shows results of a $k$-means unsupervised clustering of 250 handwritten characters. In this example, we construct a dataset with 50 point clouds from each of digits 0 to 4 from the example in Figure 1. We then cluster the point clouds by representing them as pairwise distance matrices and applying the $k$-means algorithm ($k = 5$), with $k$-means++ initialization (Arthur & Vassilvitskii, 2007). We underscore that each handwritten character is represented using an unaligned point cloud with varying numbers of points.

Each row in the figure corresponds to a different character, and each column corresponds to a different cluster center (MDS embedding shown above the table). Color is proportional to the number of data points assigned to each cluster. Even without supervision, this rudimentary technique for clustering finds meaningful clusters of handwritten digits from the dataset. All the “1” digits are clustered correctly. Digits 2 and 4 are mixed considerably in the computed clustering, reflecting the wide variety of handwritten styles for these two digits.

**Shape interpolation.** Figure 3 shows an application of the computation of GW barycenters between $S = 2$ input matrices two perform shape interpolation. The input matrices $(C_1, C_2) \in (\mathbb{R}^{N \times N})^2$ are Euclidean distance matrices of two input planar clouds $(x_{s,i})_i$ for $s \in \{1, 2\}$ uniformly sampled inside a shape, so that we use $p_1 = p_2 \frac{1}{N} I_N$. This means that $C_{s,i,j} \triangleq ||x_{s,i} - x_{s,j}||$ for $s \in \{1, 2\}$ and $i, j = 1, \ldots, N$. The interpolation is achieved by first computing a barycenter matrix $C_t$ minimizing (10) with our optimization algorithm, for $\lambda = (t, 1 - t), t \in [0, 1]$, using the Euclidean loss $L = L_2$. A barycentric point cloud $\bar{x}_i = (\bar{x}_{t,i})_{i=1}^N$ is then reconstructed using SMACOF multidimensional scaling (MDS), which amounts to computing a local minimizer of $\sum_{i,j}[C_{t,i,j} - ||\bar{x}_{t,i} - \bar{x}_{t,j}||^2$ with respect to $(\bar{x}_{t,i})$. This local minimizer $\bar{x}_i$ is computed by a gradient descent scheme, initialized with $x_1$.

**L$^2$ vs. KL losses.** Figure 4, upper part, shows a comparison of the same interpolation as those computed in Figure 3, computed with two different losses. Row #1 shows barycenters of the distance matrices $C_{s,i,j} = ||x_{s,i} - x_{s,j}||$, for $s \in \{1, 2\}$, using the squared Euclidean loss $L = L_2$ as in the previous paragraph. The display is obtained by applying SMACOF MDS to the barycenter matrix $C_t$. Row #2 shows barycenter of the infinitely divisible kernels $C_{s,i,j} = \exp(-||x_{s,i} - x_{s,j}||^2/\sigma^2)$, for $s \in \{1, 2\}$, using the Kullback-Leibler loss $L = KL$, and a bandwidth $\sigma = 1$ (for point clouds in the unit square). The display is then obtained by applying SMACOF MDS to $\sqrt{-\sigma \log(C_t)}$. The KL interpolation produces less regular interpolation because it imposes only local constraints at the scale of the bandwidth.

**Figure 3.** Progressive interpolation $(\bar{x}_t)_{t=0}^1$ between two shapes $(x_0, x_1)$ using GW barycenters.

**Figure 4.** Upper part: comparison between interpolation using $L^2$ loss and KL loss. Lower part: comparison between interpolation using pairwise Euclidean and inner geodesic distances.
Euclidean vs. Geodesic distance matrices. Euclidean pairwise distance, though simple to compute and to manipulate, fail to capture the intrinsic geometry of shapes, and thus leads to physically unrealistic shape interpolation, that ignore in particular articulation. Figure 4, lower part, shows how this issue can be in large part alleviated by replacing the extrinsic Euclidean distance $d_{s,i,j} = \|x_{s,i} - x_{s,j}\|$ for $s \in \{1, 2\}$ by the inner geodesic distance (i.e. the length of the shortest path joining $x_{s,i}$ to $x_{s,j}$ while staying inside the shape). Note that, because the corresponding distance is not Euclidean anymore, the SMACOF MDS only produces an approximate embedding, the so-called bending invariant (Elad & Kimmel, 2003) used to perform isomeric-invariant shape recognition.

4.2. Quantum chemistry

To demonstrate the interest of the Gromov-Wasserstein discrepancy, we consider its application to a regression problem on molecules. Several recent works (Rupp et al., 2012; Hansen et al., 2013) have proposed to predict atomization energies for molecules using descriptors of labeled molecules, rather than estimating them through expensive numerical simulations. (Rupp et al., 2012) proposed in particular to represent each molecule in the qm7 dataset of organic 7165 molecules through its so-called “Coulomb matrix,” which we describe next.

Coulomb matrices. For a molecule $s$ with $N_s$ atoms, each described by a relative location $r_i \in \mathbb{R}^3$ in space and a nuclear charge $Z_i$, Rupp et al. proposed to form the $N_s \times N_s$ matrix $C_s$ with off-diagonal terms $Z_i Z_j / \|r_i - r_j\|$ for $1 \leq i \neq j \leq N_s$, and diagonal terms $\frac{1}{2} Z_i^2$. Although this plays no major role in our analysis, we found that these 7165 Coulomb matrices are all infinitely divisible positive definite kernel matrices. Rupp et al. argue that the Coulomb matrix is a convenient descriptor for molecules in the sense that it is invariant with respect to translations and rotations by construction. It has, however, an important weakness: $C_s$ has an arbitrary ordering of the $N_s$ atoms in $s$. As we explain next, this was addressed by Rupp et al. by creating randomly permuted copies of these matrices.

Previous work. Because their approach requires manipulating matrices of the same size, Rupp et al. pad all $N_s \times N_s$ Coulomb matrices $C_s$ with zeros to obtain $23 \times 23$ matrices (23 is the maximal number of atoms in a molecule in the qm7 database). To cope with the ordering problem, they also propose, as a representation for each molecule, to form several randomly permuted copies of $C_s$ for each molecule $s$ (between 8 and 1000 in (Hansen et al., 2013)). (Hansen et al., 2013, Table 3) provide out-of-sample mean absolute error (MAE) predictions for several techniques using 5 fold cross-validation. We report some of them in Table 4.2.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$-nearest neighbors</td>
<td>71.54</td>
<td>95.97</td>
</tr>
<tr>
<td>Linear regression</td>
<td>20.72</td>
<td>27.22</td>
</tr>
<tr>
<td>Gaussian kernel ridge regression</td>
<td>8.57</td>
<td>12.26</td>
</tr>
<tr>
<td>Laplacian kernel ridge regression (8)</td>
<td>3.07</td>
<td>4.84</td>
</tr>
<tr>
<td>Multilayer Neural Network (1000)</td>
<td>3.51</td>
<td>5.96</td>
</tr>
<tr>
<td><strong>GW 3-nearest neighbors</strong></td>
<td>10.83</td>
<td>29.27</td>
</tr>
</tbody>
</table>

Table 1. Mean-Absolute and Root Mean Squared errors for the atomization energy prediction in the qm7 database of 7165 molecules. All results quoted from (Hansen et al., 2013, Table 3) except ours. Number in between parenthesis stand for random copies used in the algorithm. Although the GW distance is not directly competitive performance-wise, this result shows that an acceptable performance on this task can be recovered exclusively using simple metric tools.

Our approach. We propose to use the original $N_s \times N_s$ Coulomb matrices $C_s$ directly as inputs for our GW distances. Namely, we compute a discrepancy between two molecules $s, s'$ that is exactly $GW(C_s, C_{s'}, 1_{N_s} / N_s, 1_{N_{s'}} / N_{s'})$, using $L = L_2$ and a regularization strength $\varepsilon$ tuned heuristically on a subset of 500 points of the database. Using a $3$-nearest neighbor regression approach we obtain a MAE of 10.83 which, although not directly competitive with the best approach recorded in (Hansen et al., 2013), it remarkable in the sense that it is obtained with an extremely simple classifier. Our result contradicts thus the observation made by Hansen et al., p.3414, that This [poor performance of k-nn] indicates that there are meaningful linear relations between physical quantities in the system and that it is insufficient to simply look up the most similar molecules (as k-nearest neighbors does). The k-nearest neighbors approach fails to create a smooth mapping to the energies.

5. Conclusion

The Gromov-Wasserstein discrepancy measure is relevant to countless problems in machine learning for which the primary role of a given data point is defined relative to other members of the dataset. After introducing an entropic regularizer to the problem, we in particular find that GW is not only theoretically attractive but also a practical tool with an efficient and easily-implemented non-convex optimization scheme. Our approach highlights the fact that using unregistered similarity matrices as features in machine learning problems is fruitful in many contexts. GW provides a natural framework for manipulation of these matrices, as we demonstrate on classification, clustering, and regression tasks for shapes and molecules. We suspect this elegant and efficient solution to extend to many other problem instances endowed with natural geometric structure.
Acknowledgements  The work of G. Peyré has been supported by the European Research Council (ERC project SIGMA-Vision). M. Cuturi gratefully acknowledges the support of JSPS young research A grant 26700002. J. Solomon acknowledges the support of the NSF Mathematical Sciences Postdoctoral Research Fellowship (award number 1502435).

References


