Regularized Wasserstein Means for Aligning Distributional Data

Liang Mi, Wen Zhang, Yalin Wang
Arizona State University
{liangmi, wzhan139, ylwang}@asu.edu

Abstract

We propose to align distributional data from the perspective of Wasserstein means. We raise the problem of regularizing Wasserstein means and propose several terms tailored to tackle different problems. Our formulation is based on the variational transportation to distribute a sparse discrete measure into the target domain. The resulting sparse representation well captures the desired property of the domain while reducing the mapping cost. We demonstrate the scalability and robustness of our method with examples in domain adaptation, point set registration, and skeleton layout.

Introduction

Aligning distributional data is fundamental to many problems in machine learning. From the early work on histogram manipulation, e.g. (Stark 2000), to the recent work on generative modeling, e.g. (Beecks and others 2011), researchers have proposed various alignment techniques that benefit numerous fields including domain adaptation, e.g. (Sun and others 2016), and shape registration, e.g. (Ma and others 2016). A universal approach to aligning distributional data is through optimizing an objective function that measures the loss of the map between them. Regarding one distribution as the fixed target and the other the source, the alignment process in general follows an iterative manner where we alternatively update their correspondence and transform the source. When the source has much fewer samples or in a lower dimension, the process is essentially finding a sparse representation (Bengio and others 2013).

The optimal transportation (OT) loss, or the Wasserstein distance, has proved itself to be superiors in many aspects over several other distances (Gibbs and others 2002; Arjovsky and others 2017), benefiting various learning algorithms. By regarding the Wasserstein distance as a metric, researchers have been able to compute a sparse *mean* (Ho and others 2017) of a distribution, which is a special case of the *Wasserstein barycenter* problem (Agueh and others 2011) when there is only one target distribution. While OT algorithms find the correspondence between the distributions, updating the mean can simply follow the rule that each source sample is mapped to the weighted mean of its corresponding target sample(s) (Ye and others 2017).

In this paper, we raise the problem of regularizing the *Wasserstein means*. In addition to finding a mean that yields the minimum transportation cost, in many cases we also want to insert certain properties so that it satisfies other criteria. A common technique is adding regularization terms to the objective function. While most of the existing work, e.g. (Cuturi 2013; Courty and others 2017b), focus on regularizing the optimal transportation itself, we address the mean update rule and show the benefit from regularizing it. We introduce a new framework to compute OT-based sparse representation with regularization. We base our method on variational transportation (Mi and others 2018) which produces a map between the source and the target distributions in a many-to-one fashion. Different from directly mapping the source into the weighted average of its correspondence (Ye and others 2017; Courty and others 2017b; Mi and others 2018), we propose to regularize the mapping to cope with specific problems – domain adaptation, point set registration, and skeleton layout. The resulting mean, or centroid, can well represent the key property of the distribution while maintaining a small reconstruction error. Code is available at https://github.com/icemiliang/pyvot

Related Work

Optimal Transportation

The optimal transportation (OT) problem was raised introduced by Monge (Monge 1781) in the 18th century, which sought a mass-preserving transportation map between distributional data with the minimum cost. It resurfaced in 1940s when Kantorovich (Kantorovich 1942) introduced a relaxed version where mass *can* be split and provided the classic linear programming solution. A breakthrough for the mass-preserving, or non-mass splitting, OT happened in the early 1990s when Brenier (Brenier 1991) proved its existence under quadratic Euclidean cost. In more recent years, fast algorithms for computing, or approximating, OT have been proposed in both lines of research – non-mass-preserving, e.g. (Rabin and others 2011; Cuturi 2013; Solomon and others 2015) and mass-preserving, e.g. (Mérigot 2011; Lévy 2015; Kolouri and others 2016; Chen and others 2019).

We follow Monge's mass-preserving formulation. Specifically, we adopt (Mi and others 2018) with improvements to compute the OT because it gives us a clear path of each sample, not a spread-out map. Thus, we can directly regularize the support instead of the mapping.

Wasserstein Barycenters and Means

The Wasserstein distance is the minimum cost induced by OT. In most cases, the cost itself may not be as desired as the map, but it satisfies all metric axioms (Villani 2003) and thus often serves as the loss for matching distributions, e.g. (Ling and Okada 2007; Arjovsky and others 2017). Moreover, given multiple distributions, one can find their weighted average with respect to the Wasserstein metric. This problem was studied in (McCann 1997; Ambrosio and others 2008) for averaging two distributions and generalized to multiple distributions in (Agueh and others 2011), which coins the *Wasserstein barycenter* term.

A special case of the barycenter problem is when there is only one distribution and we want to find its sparse discrete barycenter. Because computationally it is equivalent to the *k-means* problem, (Ho and others 2017) defines it as the *Wasserstein means* problem. Before that, Cuturi and Doucet had discussed it in (Cuturi and others 2014) along with the connection of their algorithm to Lloyd's algorithm in that case. (Mi and others 2018) proposes an OT-based clustering method which is very close to the Wasserstein means problem. (Kolouri and others 2018) also made a contribution by discussing the *sliced Wasserstein Means* problem.

Our work focuses on *regularizing* the Wasserstein means. We obtain the mean by mapping the sparse points into the target domain according to the OT correspondence. We insert regularization into the mapping process so that the sparse points not only have a small OT loss but they also have certain properties induced by the regularization terms.

Our work should not be confused with other work on regularizing OT. For example, (Cuturi 2013) introduces entropy-regularized OT where the entropy term controls the sparsity of the map and it was later used in (Cuturi and others 2014) to compute Wasserstein barycenters. (Courty and others 2017b) also leveraged class labels to regularize OT for domain adaptation. (Ferradans and others 2014) proposed Sobolev norm-based regularized OT and further regularized barycenter and yet the regularization is still added to the OT, not the barycenter. These works only regularize OT and then directly update the support simply to the average of its correspondence. In this paper, we regularize the update.

Preliminaries

We begin with some basics on optimal transportation (OT). Suppose M is a compact metric space, $\mathcal{P}(M)$ is the space of all Borel probability measures on M and $\mu, \nu \in \mathcal{P}(M)$ are two such measures. A measure in the product space, $\pi(\cdot,\cdot) \in \mathcal{P}(M \times M)$, serves as a mapping between any two measures on M, i.e. $\pi \colon M \to M$. We define the cost function of the mapping as the geodesic distance $c(\cdot,\cdot) \colon M \times M \to \mathbb{R}^+$.

Optimal Transportation

For a mapping $\pi(\mu,\nu)$ to be legitimate, the push-forward measure of one measure has to be the other one, i.e. $\pi_\# \mu = \nu$. Thus, for any measurable subsets $B, B' \subset M$ we have $\pi(B \times M) = \mu(B)$ and $\pi(M \times B') = \nu(B')$. We denote the space of all legitimate product measures by $\Pi(\mu,\nu) = \{\pi \in \mathcal{P}(M \times M) \mid \pi(\cdot, M) = \mu,\ \pi(M, \cdot) = \nu\}$.

Optimal transportation seeks a solution $\pi \in \Pi(\mu,\nu)$ that produces the minimum total cost:

$$W_p(\mu,\nu) \overset{\text{def}}{=} \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{M \times M} (c(x,y))^p d\pi(x,y) \right)^{\frac{1}{p}}, \quad (1)$$

where p indicates the finite moment of the cost function. The minimum cost is the *p-Wasserstein distance*. In this paper, we only consider the 2-Wasserstein distance, W_2.

Monge's formulation restricts OT to preserve measures, that is, mass cannot be split during the mapping. Letting T denote such a mapping, $T \colon x \to y$, we have $d\pi(x,y) \equiv d\mu(x)\delta(y - T(x))$. Therefore, we formally define T as

$$T_{opt} = \arg\min_T \int_M c(x, T(x))^p d\mu(x). \quad (2)$$

In this paper, we follow (2). The details of the optimal transportation problem and the properties of the Wasserstein distance can be found in (Villani 2003; Gibbs and others 2002). With the abuse of notation, we use $\pi(\mu,\nu)$ to denote the Monge's OT map between μ and ν and since the map is applied to their supports x and y we also use $\pi \colon x \to y$ and $y = \pi(x)$ to denote the map.

Variational Optimal Transportation

Suppose μ is continuous and ν is a set of Dirac measures in $M = \mathbb{R}^n$, supported on $\Omega_\mu = \{x \in M \mid \mu(x) > 0\}$ and $\Omega_\nu = \{y_j \in M \mid \nu_j > 0\}$, $j = 1, ..., k$, and their total measure equal: $vol(\Omega) = \int_\Omega d\mu(x) = \sum_{j=1}^k \nu_j$. (Gu and others 2013) proposed a variational solution to this *semi-discrete OT* on $\mathbb{R}^n$. It starts from a vector $\boldsymbol{h} = (h_1, ..., h_k)^T$ and a piece-wise linear function: $\theta_{\boldsymbol{h}}(x) = \max\{\langle x, y_j \rangle + h_j\}$, $j = 1, ..., k$. Alexandrov proved in (Alexandrov 2005) that there exists a unique $\boldsymbol{h}$ that satisfies the following

$$vol(x \in \Omega \mid \nabla\theta_{\boldsymbol{h}}(x) = y_j) = \nu_j. \quad (3)$$

Furthermore, Brenier proved in (Brenier 1991) that $\nabla\theta_{\boldsymbol{h}} \colon x \to y$ *is* the Monge's OT-Map if the transportation cost is the quadratic Euclidean distance $\|x - \nabla\theta_{\boldsymbol{h}}(x)\|_2^2$.

Suppose $S_j(h) = \{x \in M \mid \nabla\theta_{\boldsymbol{h}}(x) = y_j\}$ is the projection of $\theta_{\boldsymbol{h}}$ on Ω. Variational OT (VOT) solves

$$\begin{aligned} E(\boldsymbol{h}) &\overset{\text{def}}{=} \int_\Omega \nabla\theta_{\boldsymbol{h}} d\mu - \sum_{j=1}^k \nu_j h_j \\ &\equiv \int_{\boldsymbol{0}}^{\boldsymbol{h}} \left(\sum_{j=1}^k \int_{\Omega \cap S_j(h)} d\mu \right) dh - \sum_{j=1}^k \nu_j h_j, \end{aligned} \quad (4)$$

and thus converts the OT problem into searching in a vector space $\mathcal{H} = \{h \in \mathbb{R}^k \mid \int_{\Omega \cap S_j(h)} d\mu > 0 \text{ for all } j\}$.

Proved in (Gu and others 2013), E (4) is convex in $\mathcal{H}$ when $\sum_{j=1}^{k} h_j = 0$. The gradient of (4) is (3). Thus, minimizing (4) when its gradient approaches $\mathbf{0}$ will give us the desired $\boldsymbol{h}$, and the map $\nabla\theta_{\boldsymbol{h}}$.

Wasserstein Barycenters

Given a collection of measures and weights $\{\mu_i, \lambda_i\}_{i=1}^{N}$, there exists such a measure ν that the weighted average of the Wasserstein distances between ν and all μ_i's reaches the minimum. As exposed in (Agueh and others 2011), Agueh and Carlier defined such a problem as finding a barycenter in the measure space with respect to the Wasserstein distance:

$$\nu = \underset{\nu \in \mathcal{P}_2(M)}{\arg\min} \sum_{i=1}^{N} \lambda_i W_2^2(\nu, \mu_i).$$

Wasserstein barycenters of discrete measures exist for mass splitting OT but may not for non-mass splitting or measure-preserving OT. Yet, proved in (Anderes and others 2016), when the weights are uniform and all measures have finite number of supports, there still exists a barycenter ν that preserves the measure and whose number of supports $|\Omega_\nu|$ has a tight upper bound $|\Omega_\nu| \leq \sum_{i=1}^{N} |\Omega_{\mu_i}| - N + 1$, and the OT from every μ_i to ν preserves the measure.

Wasserstein Means via Variational OT

A special case of the Wasserstein barycenters problem is when $N = 1$. In that case, we are computing a barycenter of a single probability measure. We call it the *Wasserstein mean* (WM). Beyond a special case, the barycenters and the means have the following connection.

Proposition 1. *Given a compact metric space M, a transportation cost $c(\cdot,\cdot)\colon M \times M \to \mathbb{R}^+$, and a collection of Borel probability measures $\mu_i \in \mathcal{P}(M)$, with weights λ_i, $i = 1, ..., N$, the Wasserstein mean ν_m of their average measure induces a lower bound of the average Wasserstein distance from the barycenter ν_b to them, provided that $|\Omega_{\nu_b}| \leq |\Omega_{\nu_m}| \leq k$ for some finite k.*

Proof. Since $W_2^2(\nu_b, \cdot)$ is convex for its metric property, according to Jensen's inequality, we have

$$W_2^2(\nu_b, \sum_{i=1}^{N} \lambda_i \mu_i) \leq \sum_{i=1}^{N} \lambda_i W_2^2(\nu_b, \mu_i).$$

Then, according to Wasserstein mean's definition,

$$W_2^2(\nu_m, \sum_{i=1}^{N} \lambda_i \mu_i) \leq W_2^2(\nu_b, \sum_{i=1}^{N} \lambda_i \mu_i),\ \forall \nu_b.$$

The result shows. The equal sign holds when $N = 1$. □

Algorithm 1: Wasserstein Means

```
Input  : $\mu(x) \in \mathcal{P}(M)$ and Dirac measures $\{\nu_j, y_j\}$
t = 0.
repeat
    $\nu^{(t+1)} \leftarrow$ Update weight according to (7).
    $\pi^{(t+1)} \leftarrow$ Compute OT with fixed $y^{(t)}, \nu^{(t)}$.
    $y^{(t+1)} \leftarrow$ Update support according to (6).
    $t \leftarrow t + 1$.
until convergence.
return $\pi, y, \nu$.
```

We should point out that if $\{\mu_i\}$ are discrete measures, then for the barycenter to exist we need to add the condition from (Anderes and others 2016) that $|\Omega_{\nu_b}| \leq \sum_{i=1}^{N} |\Omega_{\mu_i}| - N + 1$, which also bounds $|\Omega_{\nu_m}|$ through $|\Omega_{\nu_m}| \leq \sum_{i=1}^{N} |\Omega_{\mu_i}|$.

Now, approaching Wasserstein means is essentially through optimizing the following objective function:

$$\begin{aligned} \min f(\pi, y, \nu) &\overset{\text{def}}{=} \min_{\pi, y_j, \nu_j} \sum_{j=1}^{k} \sum_{y_j = \pi(x)} \mu(x) \|y_j - x\|_2^2, \\ \text{s.t.}\ \ \nu_j &= \sum_{y_j = \pi(x)} \mu(x). \end{aligned} \tag{5}$$

Compared to OT, solving WM w.r.t. (5) introduces 2 additional parameters – measure ν and its support y. When y and ν are fixed, (5) becomes a classic optimal transportation problem and we adopt variational optimal transportation (VOT) (Mi and others 2018) to solve it. Thus, (5) is minimizing the lower bound of the OT cost.

Then, it boils down to solving for y and ν. Certainly (5) is differentiable at all $y \in \mathbb{R}^{n \times k}$ and is convex. It's optimum w.r.t. y can be achieved at

$$\tilde{y_j} = \frac{\int_{\Omega_\mu \cap S_j} x d\mu(x)}{\int_{\Omega_\mu \cap S_j} d\mu(x)}. \tag{6}$$

It is essentially to update the mean to the centroid of corresponding measures, adopted in for example (Cuturi and others 2014; Ye and others 2017; Courty and others 2017b). The slight difference in our method is that VOT is non-mass splitting and thus the centroid in our case has a clear position without the need for weighting.

As discussed in (Cuturi and others 2014), (5) is not differentiable w.r.t. ν. However, we can still get its optimum through the following observation.

Observation 1. *The critical point of the function $\nu \to f(\pi, \nu)$ is where ν induces π being the gradient map of the unweighted Voronoi diagram formed by ν's support y. In that case, every empirical sample $\mu(x)$ at x is mapped to its nearest y_j, which coincides with Lloyd's algorithm.*

Proof. Suppose ν induces the OT map π from every x to its nearest y_j. Then, the map $\pi'\colon x \to y_{j'}$ that satisfies any other $\nu' = \int_{\Omega \cap S_{j'}} d\mu(x)$ will yield an equal or larger cost $\int_\Omega \|y_j - x_i\|_2^2 d\mu(x_i) \leq \int_\Omega \|y_{j'} - x_i\|_2^2 d\mu(x_i)$. □

Thus, we can write the update rule for ν as

$$\begin{aligned} \tilde{\nu}(y_j) &= \int_{\Omega \cap S_j} d\mu(x), \\ \text{s.t. } S_j &= \{x \in M \mid \|x - y_j\|_2 \leq \|x - y_i\|_2, i \neq j\}. \end{aligned} \tag{7}$$

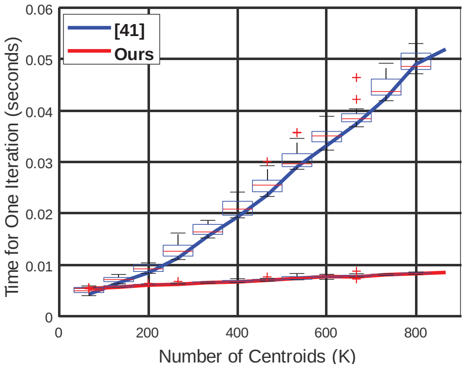


Figure 1: Comparison of time over number of centroids.

Updating the three parameters π, y, and ν can follow the *block coordinate descent* method. Since at each iteration we have closed-form solutions in the y and ν directions, there is no need to do a line search there. We wrap up our algorithm for computing the Wasserstein means in Alg. 1

As discussed in (Cuturi and others 2014), when $N = 1$ and $p = 2$, computing the Wasserstein barycenter (in this case the Wasserstein mean) is equivalent to Lloyd's k-means algorithm. The difference also occurs when we have a constraint on the weight $\nu_j(y)$. Ng (Ng 2000) considered a uniform weight for all S_j. Our algorithm can adapt to any constraint on $\nu_j \geq 0$. In this case, our algorithm is equivalent to (Cuturi and others 2014) where the update of the support is equivalent to re-centering it by our (6).

Complexity In practice, we use the total mass of the discrete measures inside each S. Then, we vectorize the computation with PyTorch because parameters in VOT, $\boldsymbol{h}$ (4), can be optimized individually and thus parallelly. Given N empirical samples and K centroids, our implementation of OT runs $\mathcal{O}(KN)$ on CPU and theoretically $\mathcal{O}(N)$ on GPU. Figure 1 shows timing over $K \leftarrow [20 : 1000]$. $N = 10,000$. The boxes along the plots come from 10 runs of 300 iterations for each K. The dimension of the data is 3. y axis is in seconds per iteration. The plot shows the increased K add few burden to RWM. The complexity added by regularization is as follows. The complexity in 5.1 is $\mathcal{O}(K)$; 5.3 is $\mathcal{O}(K^3)$ mainly from solving SVD, but in practice we choose a small or a constant number $K' << K$ for SVD; 5.4 is $\mathcal{O}(K)$ for computing curvature. Thus, the total computational complexity of RWM is $\mathcal{O}(N) + \mathcal{O}(K^3)$, depending on the regularization term. We also compute the pair-wise distances between empirical samples and centroids beforehand as in (Cuturi 2013), making the memory consumption on the level of $\mathcal{O}(KN)$.

Regularized Wasserstein Means

In many problems of machine learning, the solution that comes purely from the perspective of the mapping cost may not serve the best to represent the connection between origins and their images, let alone overfitting. Regularization is a common technique to introduce desired properties in the solution. In the previous section, we talked about the Wasserstein means problem and its optimizers: OT $\pi(\nu, \mu)$, support y, and the measure $\nu(y)$. In this section, we detail our strategies to regularize y along with several regularization terms that we propose to penalize the Wasserstein means cost. For simplicity, we fix the given $\nu(y)$ in the following arguments and only consider π and y in the *regularized Wasserstein means* (RWM) problem.

We start with a general loss function:

$$\begin{aligned} \mathcal{L}(\pi, y) &= \mathcal{L}_{ot}(\pi, y) + \lambda \mathcal{L}_{\text{reg}}(y), \\ \mathcal{L}_{ot}(\pi, y) &= \int_{\Omega} \|y - x\|_2^2 d\mu(x), \text{ where } y = \pi(x). \end{aligned} \tag{8}$$

We call the first term the *OT loss* or data loss. Our goal here is to explore $\mathcal{L}_{\text{reg}}(y)$ and the use of it. Optimizing (8) can also follow the block coordinate descent method. First, we fix the mean and compute the OT. Unlike in Alg. 1 where we directly update the mean to the average of their correspondences, next, we regularize the mean to satisfy certain properties through local minimization on (8).

Minimizing the OT loss $\mathcal{L}_{ot}(\pi, y)$ w.r.t. y can be simplified to minimizing the quadratic loss for each support, i.e. $\mathcal{L}_{\tilde{y}} = \sum_j \|y_j - \tilde{y}_j\|_2^2$, since they are equivalent:

$$\begin{aligned} \int_{S_j} \|y_j - x\|_2^2 d\mu(x) &= (y_j^2 - 2y_j \int_{S_j} x d\mu(x) + C_1) \\ &= \|y_j - \int_{S_j} x d\mu(x)\|_2^2 + C_2 = \|y_j - \tilde{y}_j\|_2^2 + C_2. \end{aligned} \tag{9}$$

C_1, C_2 are some constants. $\tilde{y}_j$ is from (6) and S_j is the set in which x is mapped to y_j. It is defined by VOT as $S_j = \{x \in M | \langle y_j, x\rangle - h_j \geq \langle y_i, x\rangle - h_i\}, \forall i \neq j$, see Sec. . Thus, we re-write (8) as

$$\mathcal{L}(\pi, y) = \sum_j \|y_j - \tilde{y}_j\|_2^2 + \lambda \mathcal{L}_{\text{reg}}(y) \tag{10}$$

Note, that $\mathcal{L}_{\text{reg}}$ undermines the metric properties of the Wasserstein distance and yet the distance is not our concern but the data term of the loss we designed for a broad range of applications. We provide the general algorithm to compute regularized Wasserstein means in Alg. 2.

Citing the convergence proof from (Grippo and others 2000), as long as we add a convex regularization term, because $\pi\colon x \to y$ is compact and convex, our 2-block coordinate descent-based algorithm indeed will converge. In the

```
Algorithm 2: Regularized Wasserstein Means
Input  : $\mu(x) \in \mathcal{P}(M)$, $\{\nu_j, y_j\}$
$t = 0$.
repeat
    $\pi^{(t+1)} \leftarrow$ Compute OT $\pi(\mu, \nu)$ with fixed $y^{(t)}$.
    $\tilde{y} \leftarrow$ Compute new centroid according to (6).
    repeat
        $y^{(t+1)} \leftarrow$ Update centroid by optimizing (10).
    until $y^{(t+1)}$ converges.
    $t \leftarrow t + 1$.
until $\pi$ and $y$ converge.
return $\pi, y$.
```

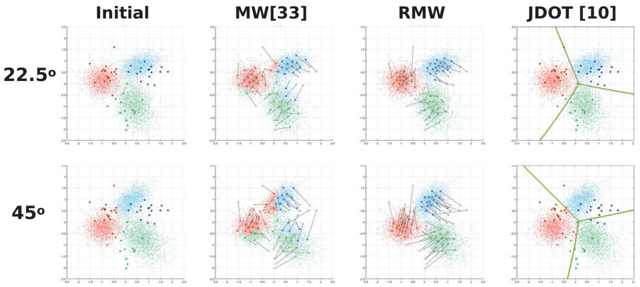
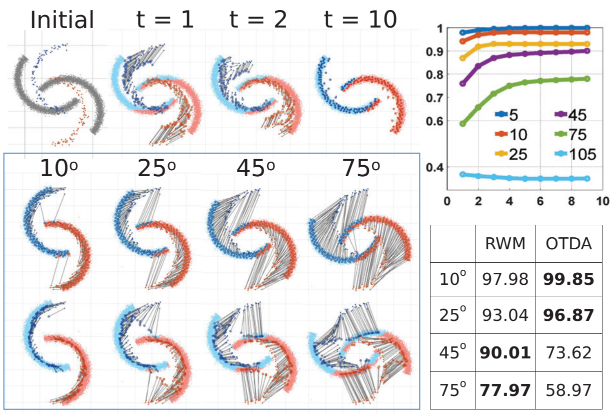


Figure 2: Regularizing the WM by the intra-class triplets can adapt it to domains that suffer unknown rotations.



	RWM	OTDA
10°	97.98	**99.85**
25°	93.04	**96.87**
45°	**90.01**	73.62
75°	**77.97**	58.97

Figure 3: RWM adapting shifted two moons: 1st row performance over iteration under 45^o; 2nd and 3rd rows performances of RWM and OTDA under different degrees.

rest of this section, we discuss in detail several regularization terms based on class labels, geometric transformation, and length and curvature, all of which are convex.

Triplets Empowered by Class Labels

We begin with a fair assumption that samples of the same class reside closer to each other and samples that belong to different classes are relatively far away from each other. This behavior can be expressed by signed distances between samples. Given that, we propose to regularize the mean update process by adding a *triplet* loss, promoting intra-class connection and discouraging inter-class connections.

The triplet loss was proposed in (Schroff and others 2015), inspired by (Weinberger and others 2009). It targets the metric learning problem which is finding an embedding space where samples of the same desired property reside close to each other and vise versa. In triplets, samples are characterized into three types – *anchor*, *positive*, and *negative*, denoted as y^a, y^p, and y^n. The motivation is that the anchor is closer by a margin of α to a positive than it is to a negative:

$$\mathcal{L}_{\text{reg}}(y) = \sum_{i}^{K} [\|y_j^a - y_j^p\|_2^2 - \|y_j^a - y_j^n\|_2^2 + \alpha]_+ .$$

The overall RWM loss w.r.t. y (10) becomes

$$\mathcal{L}(y) = \sum_j \|y_j - \tilde{y}_j\|_2^2 + \lambda \mathcal{L}_{\text{triplet}}(y). \tag{11}$$

Fig. 2 shows an example of aligning Gaussian mixtures by (11). Suppose a mixture has three components with different parameters, each belonging to a different class shown in three colors. We rotate the mixture by a certain degree to emulate an unknown shift and apply our method to recover the shift.

We sample the source domain 50 times and the target domain 5,000 times at 22.5^o and 45^o. Fig. 2 1st column shows the setups. The 2nd column shows the result from computing the WM without regularization as in (Mi and others 2018). The 3rd column shows our result. Our method can well drive source samples into the correct target domain. The lighter colors on the target samples in the 2nd column indicate the predicted class by using the OT correspondence. Since our OT preserves the measure during the mapping, we can deterministically label each unknown sample by querying its own centroid's class. Note, that this is equivalent to the 1NN classification algorithm based on the *power Euclidean distance* (Mi and others 2018). Only when the weight of every centroid equals each other will the power distance coincide with the Euclidean distance. In the last column, we show the result from (Courty and others 2017a). It learns an RBF SVM classifier on the target samples.

Geometric Transformations

While OT recovers a transformation between two domains that induces the lowest cost, it does not consider the structure within the domains. Pre-assuming a type of the transformation and then estimating its parameters is one of the popular approaches to solving domain alignment-related problems, for example in (Gopalan and others 2011; Courty and others 2017b). In this way, the structure of the domain can be preserved to some extent. Let us follow this trend and assume that two domains can be matched by a geometric transformation with modifications, that is, any transformation between domains is a combination of a parametric geometric transformation and an arbitrary transformation. This leads to our following strategy that we, on the one hand, regularize the mean to be roughly a geometric transformation in order to preserve the structure of the source domain during the mapping but on the other hand also allow OT to adjust the mapping so that it can recover irregular transformations.

We follow Alg. 2. First, compute OT to obtain the target mean positions $\tilde{y} = \pi(x)$ and then use the paired means $\{y, \tilde{y}\}$ to determine the parameters of a geometric transformation $\mathcal{T}$ subject to $\tilde{y} = \mathcal{T}y$ through a least squares estimate. Suppose $y_j^{\mathcal{T}} = \mathcal{T}y$ is the estimate purely based on the affine transformation, then, we have the RWM loss

$$\mathcal{L}(\pi, y) = \sum_j \|y_j - \tilde{y}_j\|_2^2 + \lambda \sum_j \|y_j - y_j^{\mathcal{T}}\|_2^2. \tag{12}$$

Candidates of the geometric transformations include but not limited to perspective, affine, and rigid transformations.

We demonstrate (12) with *two moons* in Fig. 3. The known domain contains 200 samples in blue and red. The unknown domain is the known domain after a rotation, sampled $10,000$ times in grey. We assume the prior is a rigid transformation. The top row shows the result on the 45^o case after several iterations. In the end, RWM almost recovers the transformation with a small error. Top right shows accuracy over iterations under different degrees. The 2^{nd} row shows the result under different degrees of rotation. We weight in OTDA-GL's result (Courty and others 2017b) in the 3^{rd} row showing RWM's superiority over OTDA under large transformations and its inferiority under small transformations. We also notice that RWM maps the samples *into* the domain which OTDA fails to.

Topology Represented by Length and Curvature

The nature of many-to-one mapping in the WM problem enables itself to be suitable for skeleton layout. Consider a 3D thin, elongated point cloud. Our goal is to find a 3D curve consisting of sparse points to represent the shape of the cloud. The problem with directly using WM for skeleton layout is that the support is unstructured. Therefore, we propose to pre-define the topology of the curve and add the length and curvature to regularize its geometry, both intrinsically (length) and extrinsically (curvature).

We give an order of the support so that they can form a piece-wise linear curve. For each three adjacent supports, y_{j-1}, y_j, y_{j+1}, we fit a quadratic spline curve $\gamma(t)$ of 100 points. Its length is approximated by summarizing the length segment $\int_0^{length} ds = \int_0^1 \|\gamma'(t)\| dt$, and its curvature at the middle point y_i can be approximated by the total curvature $\int_0^{length} \mathcal{K}^2(t) ds$, $\mathcal{K}(t) = \frac{\|\gamma'(t) \times \gamma''(t)\|}{\|\gamma'(t)\|^3}$ as in (Ulen and others 2015). Thus, the regularization on the length and curvature can express itself as follows:

$$\lambda \mathcal{L}_{\text{reg}} = \lambda_1 \sum_{1 \leq i < k} g(\gamma'(y_i)) + \lambda_2 \sum_{1 < i < k} l(\gamma''(y_i)). \quad (13)$$

where $g(\cdot)$ and $l(\cdot)$ are some functions computed out of the length and curvature based on y, which are both convex making (13) convex. We could go further and include torsion into the term but since we do not pursue a perfectly smooth curve but rather the reasonable embedding of the supports in the interior of the point cloud, we have passed torsion.

In case the shape have branches, we can easily extend (13) considering the skeleton as a whole when computing the OT and regularizing each branch separately. Suppose, now, the skeleton $\Gamma = \{\gamma_j\}$ is a set of 1-D curves. Finally, we propose the following loss for skeleton layout:

$$\begin{aligned} \mathcal{L}(\pi, y) &= \sum_j \|y_j - \tilde{y}_j\|_2^2 \\ &+ \sum_{\gamma \in \Gamma} \Big(\lambda_1 \sum_{1 \leq i < k} g(\gamma'(y_i)) + \lambda_2 \sum_{1 < i < k} l(\gamma''(y_i))\Big). \end{aligned} \quad (14)$$

Applications

We demonstrate the use of RWM in domain adaptation (class label), point set registration (geometric transformation), and skeleton layout (topology).

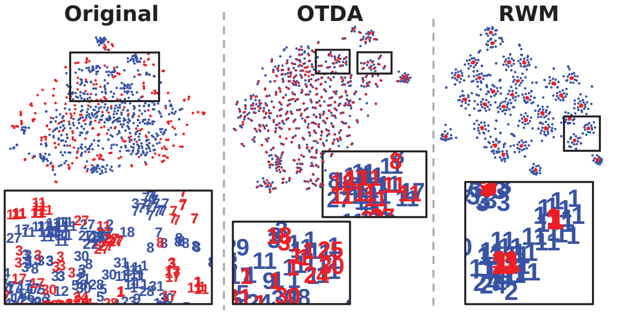


Figure 4: t-SNE plots of Office samples after OTDA and RWM.

Domain Adaptation

We evaluate our method on the office-31 dataset (Saenko and others 2010). Office-31 includes two subsets – Amazon and Webcam. We adapt from Webcam to Amazon (W → A). The Amazon set contains 2,848 images from 31 categories. Each category has a different number of samples from 36 to 100. The Webcam set archives 826 images from the same 31 categories, each having between 11 to 43 samples.

We use the Decaf-fc6 and Decaf-fc7 features provided along with the dataset. Each sample now is encoded into a vector of 4,096 dimensions. The setup is similar to OTDA (Courty and others 2017b). We randomly select 20 samples per class from Amazon and 10 samples per class from Webcam, because the 'ruler' category of Webcam only has 11 samples and we want each class to have an equal number of samples. Then, we normalize the weight of the sample so that the total weight from Amazon and from Webcam are both one. Each sample is assumed to have an equal weight: Amazon sample $1/620$ and Webcam sample $1/310$.

We compare RWM with OTDA and also include 1NN and the original WM as baselines. The experiments are repeated 10 times and Tab. 1 summarizes the averaged results. RWM outperforms other methods by a large margin. We also show the resulting t-SNE embeddings in Fig. 4. From left to right are the original embeddings, embeddings after OTDA, and embeddings after RWM. Blue dots represent Amazon samples and red dots Webcam samples. Numbers indicate classes. RWM successfully cluster samples from the same class into distinguishable clusters while OTDA on the other hand very well integrates the source domain into the target domain (but with larger errors). Zoom in the pictures to see the samples of 1, 'bike', and 11, 'keyboard'. The regularization weight of OTDA Laplacian is 0.3. It is from a search in $\{1, 0.3, 0.1, 0.03, 0.01\}$. The weight of RWM is 1 from a search in $\{3, 1, 0.3, 0.1, 0.03, 0.01\}$

Table 1: Classification Accuracy (%) on Office-31 W → A

Feature	1NN	WM	OTDA	RWM
Decaf-fc6	30.2±1.3	32.7±2.3	33.9±2.1	**36.4**±2.7
Decaf-fc7	31.3±1.9	34.6±2.2	35.8±1.5	**43.2**±2.6

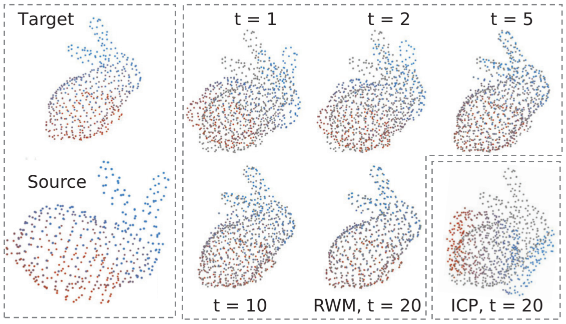


Figure 5: Alignment of translationally and rotationally shifted bunnies after RWM and ICP. t indicates the number of iterations.

Point Set Registration

Registering point sets is key to many downstream applications such as surface reconstruction and stereo matching. Point set registration algorithms aim to assign correspondences between two sets of points and to recover the transformation between them (Myronenko and Song 2010). Figure 5 left shows a Stanford Bunny in a grey point set and its shifted version in a colored point set after a random noisy translation and a rotation. We apply (12) to recovering the transformation. With this example, we also test our algorithm under the extreme condition when we have the same number of empirical samples and centroids. Our algorithm RWM still produces a one-to-one map between the two point sets. The transformation then perfectly aligns them while the traditional iterative closest point (ICP) algorithm fails to recover the transformation. The reason is that ICP assigns the correspondence based on nearest neighbors while RWM uses OT which considers the point set as a whole when computing the correspondence. Note, that by pre-defining the regularization as a rigid transformation and adjusting its weight, we can perform both rigid and non-rigid registration. In the above example, the regularization weight is $\lambda = 10$. Our alignment technique might be further incorporated into e.g. (Yang and others 2016) for globally optimal alignment.

Skeleton Layout

. Suppose we have a point cloud $\mu \in \mathcal{P}(\mathbb{R}^3)$ and a graph $G = (V, E)$ representing the topology of the shape. Then, the problem is finding particular embeddings of the nodes $y(\nu) : \nu \to \mathbb{R}^3$ that can relate the graph to the geometry of the point cloud.

Now, consider the human shape point cloud in Fig. 6 top left. We initial a rough embedding of a graph by fixing its ends $V_0 \subset V$ to certain known positions $y_{\nu \in V_0}$ which are head, hands, and feet in this example, and set the rest of nodes evenly distribute along their branches. Our goal is to embed the nodes $\nu \in V \backslash V_0$ in this $\mathbb{R}^3$ space by applying (14). Because the weight of each centroid determines its boundaries with other centroids, it has to be adjusted to the local density of the cloud so that all the centroids could roughly evenly lay on the skeleton. Thus, we relax the restriction on weight and reinstate (7). We update the weight by momentum gradient descent, $\nu(y_j)^{(t+1)} \leftarrow \lambda \nu(y_j)^{(t)} + (1-\lambda) \int_{\Omega \cap S_j} d\mu(x)$ to prevent it from quickly trapped into a local minimum like k-means.

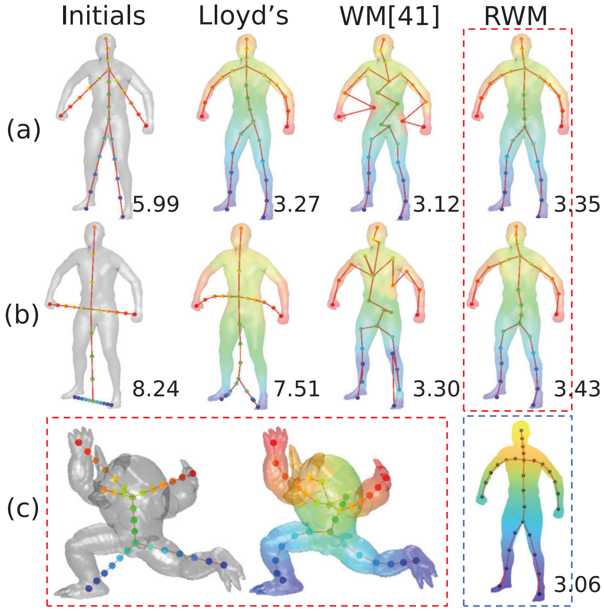


Figure 6: Skeleton layout. RWM embeds a pre-defined graph which relates to the shape of the cloud. Numbers indicating MSE showing RWM balances between MSE and topology.

Top right of Fig. 6 shows our result. The skeleton successfully captures the shape of the point cloud. Colors of the skeleton nodes based on their position in the graph are transferred to the surface according to their OT correspondences. We compare the result from Lloyd's k-means algorithm and with RW in the 2^{nd} and 3^{rd} columns. Equal weight of regularization is added to Lloyd's algorithm to make it a fair comparison. We also test our method in an extreme initial condition. As shown in (b), our algorithm eventually recovers a coherent, correct shape, but without the regularization we could end up with "ill-posed" embeddings. The figure also writes the mean square errors (MSE). Our method achieves small MSEs while maintaining the topology. In the bottom left, we show result from Stanford Armadillo. In the bottom right, we show the result from (Solomon and others 2015) as the ground truth. It regards the problem as a *Wasserstein propagation* problem and adopted Wasserstein barycenter techniques to relate the samples of the cloud to the graph, which is much heavier. The average time of 5 trials by (Solomon and others 2015) was 1,200 seconds while ours took 15 seconds. CPU: Intel i5-7640x 4.0 GHz.

Conclusion

We have talked about the Wasserstein means problem and our method to regularize it. The results have shown that our method can well adapt to different problems by adopting different regularization terms. This work opens up a new perspective to look at the Wasserstein means problem, or the k-means problem, as well as regularizing them.

We expect further use of regularized optimal transportation techniques on aligning distributions in high-dimensional spaces. Future work in our line of research could also include regularizing the barycenters.

Acknowledgements This work was supported in part by NIH (RF1AG051710 and R01EB025032). Liang Mi is supported in part by ASU Completion Fellowship.

References

Agueh, M., et al. 2011. Barycenters in the Wasserstein space. *SIAM J. on Mathematical Analysis* 43(2):904–924.

Alexandrov, A. D. 2005. *Convex polyhedra*. Springer Science & Business Media.

Ambrosio, L., et al. 2008. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media.

Anderes, E., et al. 2016. Discrete Wasserstein barycenters: Optimal transport for discrete data. *Mathematical Methods of Operations Research* 84(2):389–409.

Arjovsky, M., et al. 2017. Wasserstein generative adversarial networks. In *ICML*.

Beecks, C., et al. 2011. Modeling image similarity by Gaussian mixture models and the signature quadratic form distance. In *ICCV*.

Bengio, Y., et al. 2013. Representation learning: A review and new perspectives. *IEEE TPAMI* 35(8):1798–1828.

Brenier, Y. 1991. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics* 44(4):375–417.

Chen, Y., et al. 2019. A gradual, semi-discrete approach to generative network training via explicit wasserstein minimization. In *ICML*.

Courty, N., et al. 2017a. Joint distribution optimal transportation for domain adaptation. In *NeurIPS*.

Courty, N., et al. 2017b. Optimal transport for domain adaptation. *IEEE TPAMI* 39(9):1853–1865.

Cuturi, M., et al. 2014. Fast computation of Wasserstein barycenters. In *ICML*.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*.

Ferradans, S., et al. 2014. Regularized discrete optimal transport. *SIAM J. on Imaging Sciences* 7(3):1853–1882.

Gibbs, A. L., et al. 2002. On choosing and bounding probability metrics. *Intl. statistical review* 70(3):419–435.

Gopalan, R., et al. 2011. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*.

Grippo, L., et al. 2000. On the convergence of the block nonlinear Gauss–Seidel method under convex constraints. *Operations research letters* 26(3):127–136.

Gu, X., et al. 2013. Variational principles for Minkowski type problems, discrete optimal transport, and discrete Monge-Ampere equations. *arXiv preprint arXiv:1302.5472*.

Ho, N., et al. 2017. Multilevel clustering via Wasserstein means. In *ICML*.

Kantorovich, L. V. 1942. On the translocation of masses. In *Dokl. Akad. Nauk SSSR*, volume 37, 199–201.

Kolouri, S., et al. 2016. Sliced wasserstein kernels for probability distributions. In *CVPR*.

Kolouri, S., et al. 2018. Sliced wasserstein distance for learning gaussian mixture models. In *CVPR*, 3427–3436.

Lévy, B. 2015. A numerical algorithm for L2 semi-discrete optimal transport in 3d. *ESAIM: Mathematical Modelling and Numerical Analysis* 49(6):1693–1715.

Ling, H., and Okada, K. 2007. An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE TPAMI* 29(5):840–853.

Ma, J., et al. 2016. Non-rigid point set registration by preserving global and local structures. *IEEE Trans. on image Processing* 25(1):53–64.

McCann, R. J. 1997. A convexity principle for interacting gases. *Advances in mathematics* 128(1):153–179.

Mérigot, Q. 2011. A multiscale approach to optimal transport. In *Computer Graphics Forum*, volume 30, 1583–1592.

Mi, L., et al. 2018. Variational Wasserstein clustering. In *ECCV*.

Monge, G. 1781. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*.

Myronenko, A., and Song, X. 2010. Point set registration: Coherent point drift. *IEEE TPAMI*.

Ng, M. K. 2000. A note on constrained k-means algorithms. *Pattern Recognition* 33(3):515–519.

Rabin, J., et al. 2011. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*.

Saenko, K., et al. 2010. Adapting visual category models to new domains. In *ECCV*.

Schroff, F., et al. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*.

Solomon, J., et al. 2015. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph.* 34(4):66.

Stark, J. A. 2000. Adaptive image contrast enhancement using generalizations of histogram equalization. *IEEE TIP*.

Sun, B., et al. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*.

Ulen, J., et al. 2015. Shortest paths with higher-order regularization. *IEEE TPAMI* 37(12):2588–2600.

Villani, C. 2003. *Topics in optimal transportation*. Number 58. American Mathematical Soc.

Weinberger, K. Q., et al. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research* 10(Feb):207–244.

Yang, J., et al. 2016. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE TPAMI* (11):2241–2254.

Ye, J., et al. 2017. Fast discrete distribution clustering using Wasserstein barycenter with sparse support. *IEEE Trans. on Signal Processing* 65(9):2317–2332.