Transportation Techniques for Geometric Clustering

by

Liang Mi

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2020 by the
Graduate Supervisory Committee:

Yalin Wang, Chair
Kewei Chen
Lina Karam
Baoxin Li
Pavan Turaga

ARIZONA STATE UNIVERSITY

May 2020

ABSTRACT

This thesis introduces new techniques for clustering distributional data according to their geometric similarities. This work builds upon the optimal transportation (OT) problem that seeks global minimum cost for matching distributional data and leverages the connection between OT and power diagrams to solve different clustering problems. The OT formulation is based on the variational principle to differentiate hard cluster assignments, which was missing in the literature. This thesis shows multiple techniques to regularize and generalize OT to cope with various tasks including clustering, aligning, and interpolating distributional data. It also discusses the connections of the new formulation to other OT and clustering formulations to better understand their gaps and the means to close them. Finally, this thesis demonstrates the advantages of the proposed OT techniques in solving machine learning problems and their downstream applications in computer graphics, computer vision, and image processing.

To my wife and my parents.

ACKNOWLEDGMENTS

This thesis marks not only the end of my doctoral journey, but it symbolizes the efforts of all the people who inspired, guided, supported, challenged, and most importantly, accompanied me during these years.

I would like to thank my advisor Yalin Wang and other committee members — Kewei Chen, Lina Karam, Baoxin Li, and Pavan Turaga – for their comments and guidance over the years. Professor Wang has always been an exemplar figure in my Ph.D. life that believes in my ideas, inspires and encourages me. Special thanks to Xianfeng Gu and José Bento for sharing their expertise in computational geometry and graph theories with me and their guidance in preparing my previous papers. I have also been lucky to collaborate with other talented people — Yonghui Fan, Dhruman Goradia, Eric M. Reiman, Tianshu Yu, Junwei Zhang, and Wen Zhang. None of my accomplishments would happen without them.

I spent wonderful summers with my mentors Zhixin Yan at Bosch, Wen P. Liu at Intuitive Surgical, and Darren Platt at Amyris, who equipped me with the skills that boosted my research. I also would like to thank all the members at the GSL lab at ASU for their continued help. Finally, my research has been supported by scholarships from NIH, NSF and fellowships from ASU. I am sincerely grateful.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Computer engineering engages with connecting computational theories and real-world problems by offering practical solutions with a mathematical foundation. Developing reliable numerical techniques is the ultimate goal of researchers in computer engineering. These computational techniques can be used to discover homogeneous groups, compress data for efficient storage, and measure similarities among different objects.

Partitioning distributional data into a fixed number of clusters according to their geometry is a fundamental task at the core of computer engineering with enormous applications in machine learning, computer graphics, computer vision, and image processing. This thesis investigates the connection between optimal transportation and geometric clustering and advances the computational methods for solving the optimal transportation problem with the goal of offering new perspectives and directions for solving different geometric clustering problems.

Gaspard Monge raised the optimal transportation problem more than 200 years ago, but because of its intractability, most researchers of the community have been following its relaxed version introduced in the 1940s by Leonid Kantorovich. While many works, including some of the recent focus on advancing the transportation techniques for aligning and clustering distributional data from Kantorovich's perspective, this thesis introduces techniques that solve OT problems from Monge's perspective. It builds upon the theoretical breakthrough in solving Monge's OT problem and closes the gap between Monge OT and several clustering problems. It develops the theoretical connections between Monge OT and different clustering problems and explores the advantage of using Monge's OT formulation over Kantorovich's formulation on solving these problems. The majority of the thesis has appeared in the following publications.

- Mi, Liang, Wen Zhang, Junwei Zhang, Yonghui Fan, Dhruman Goradia, Kewei Chen, Eric M. Reiman, Xianfeng Gu, and Yalin Wang. "An optimal transportation based univariate neuroimaging index." In Proceedings of the IEEE International Conference on Computer Vision, pp. 182-191. 2017.

- Mi, Liang, Wen Zhang, Xianfeng Gu, and Yalin Wang. "Variational Wasserstein Clustering." In Proceedings of the European Conference on Computer Vision (ECCV), pp. 322-337. 2018.

- Mi, Liang, Wen Zhang, and Yalin Wang. "Regularized Wasserstein Means Based on Variational Transportation." In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2020. (To appear)

- Mi, Liang, Tianshu Yu, Jose Bento, Wen Zhang, Baoxin Li, and Yalin Wang. "Variational Wasserstein Barycenters for Geometric Clustering." (In preparation).

We start with reiterating basic concepts of optimal transportation in Chapter 2 and introduce the variational principle to solve optimal transportation and the connection between OT and power Voronoi diagrams. We then discuss solving the K-means clustering problem with variational OT and its different applications. In chapter 3, we focus on aligning distributional data by variational OT. We present the practical difficulties of using the vanilla variational OT to align distributions and our solution to overcome it by inserting regularization techniques. In chapter 4, we broaden our discussion to solving the Wasserstein barycenter problems. By doing so, we achieve the full power of our clustering technique in solving optimal transportation-related problems. We further discuss the metric properties of Wasserstein barycenters and the advantages of our method over other OT solvers in solving geometric clustering problems.

Chapter 2

WASSERSTEIN CLUSTERING THROUGH POWER DIAGRAMS

In this chapter, we discuss the connection between optimal transportation (OT) and power diagrams and how we leverage the geometric properties of OT to aggregate distributional data according to their spatial similarities. Because OT produces the metric called the Wasserstein distance, we name the process Wasserstein clustering. We first present our motivation and related work on Wasserstein clustering and then provide the preliminaries on optimal transportation and K-means clustering. We then divide deep into the variational principle for solving semi-discrete optimal transportation, which is equivalent to solving a constrained K-means clustering problem. We demonstrate the use of our method with different experiments.

2.1 Introduction

Aggregating distributional data into clusters has ubiquitous applications in computer vision and machine learning. A continuous example is unsupervised image categorization and retrieval, where similar images reside close to each other in the image space or the descriptor space, and they are clustered together and form a specific category. A discrete example is document or speech analysis, where words and sentences that have similar meanings are often grouped together. K-means Lloyd (1982); Forgy (1965) is one of the most famous clustering algorithms, which aims to partition empirical observations into k clusters in which each observation has the closest distance to the *mean* of its own cluster. It was originally developed for solving quantization problems in signal processing, and in the early 2000s researchers have discovered its connection to another classic problem optimal transportation which seeks a transportation plan that minimizes the transportation cost between probability measures Graf and Luschgy (2007).

The optimal transportation (OT) problem has received great attention since its very birth. Numerous applications such as color transfer and shape retrieval have benefited from solving OT between probability distributions. Furthermore, by regarding the minimum transportation cost – *the Wasserstein distance* – as a metric, researchers have been able to compute the barycenter Agueh and Carlier (2011) of multiple distributions, e.g. Cuturi and Doucet (2014); Solomon *et al.* (2015), for various applications. Most researchers regard OT as finding the optimal coupling of the two probabilities, and thus each sample can be mapped to multiple places. It is often called the Kantorovich OT. Along with this direction, several works have shown their high performances in clustering distributional data via optimal transportation, .e.g. Ye *et al.* (2017); Solomon *et al.* (2015); Ho *et al.* (2017). On the other hand, some researchers regard OT as a measure-preserving mapping between distributions, and thus a sample cannot be split. It is called the Monge-Brenier OT.

In this chapter, we introduce a clustering method from the Monge-Brenier approach. Our method is based on Gu *et al.* Gu *et al.* (2013) who provided a variational solution to Monge-Brenier OT problem. We call it *variational optimal transportation* and name our method *variational Wasserstein clustering.* We leverage the connection between the *Wasserstein distance* and the clustering error function, and simultaneously pursue the Wasserstein distance and the K-means clustering by using a power Voronoi diagram. Given the empirical observations of a target probability distribution, we start from a sparse discrete measure as the initial condition of the centroids and alternatively update the partition and update the centroids while maintaining an optimal transportation plan. From a computational point of view, our method is solving a special case of the *Wasserstein barycenter* problem Agueh and Carlier

(2011); Cuturi and Doucet (2014) when the target is a univariate measure. Such a problem is also called the *Wasserstein means* problem Ho *et al.* (2017). We demonstrate the applications of our method to three different tasks – domain adaptation, remeshing, and representation learning. In domain adaptation on synthetic data, we achieve competitive results with D2 Ye *et al.* (2017) and JDOT Courty *et al.* (2017a), two methods from Kantorovich's OT. The advantages of our approach over those based on Kantorovich's formulation are that (1) it is a local diffeomorphism; (2) it does not require pre-calculated pairwise distances; and (3) it avoids searching in the product space and thus dramatically reduces the number of parameters.

2.2 Related Work

The optimal transportation (OT) problem was initially raised by Monge Monge (1781) in the 18th century, which sought a transportation plan for matching distributional data with the minimum cost. In 1941, Kantorovich Kantorovich (1942) introduced a relaxed version and proved its existence and uniqueness. Kantorovich also provided an optimization approach based on linear programming, which has become the dominant direction. Traditional ways of solving the Kantorovich's OT problem rely on pre-defined pairwise transportation costs between measure points, e.g., Cuturi (2013), while recently researchers have developed fast approximations that incorporate computing the costs within their frameworks, e.g., Solomon *et al.* (2015).

Meanwhile, another line of research followed Monge's OT and had a breakthrough in 1987 when Brenier Brenier (1991) discovered the intrinsic connection between optimal transportation and convex geometry. Following Brenier's theory, Mérigot Mérigot (2011), Gu *et al.* Gu *et al.* (2013), and Lévy Lévy (2015) developed their solutions to Monge's OT problem. Mérigot and Lévy's OT formulations are nonconvex, and they leverage damped Newton and quasi-Newton respectively to solve them. Gu *et al.* proposed a convex formulation of OT, particularly for convex domains where pure Newton's method works and then provided a variational method to solve it.

The Wasserstein distance is the minimum cost induced by the optimal transportation plan. It satisfies all metric axioms and thus is often borrowed for measuring the similarity between probability distributions. The transportation cost generally comes from the product of the geodesic distance between two sample points and their measures. We refer to p–Wasserstein distances to specify the exponent p when calculating the geodesic Givens *et al.* (1984). The 1–Wasserstein distance or earth mover's distance (EMD) has received great attention in image and shape comparison Rubner *et al.* (2000); Ling and Okada (2007). Along with the rising of deep learning in numerous areas, 1–Wasserstein distances have been adopted in many ways for designing loss functions for its superiority over other measures Lee *et al.* (2018); Arjovsky *et al.* (2017); Frogner *et al.* (2015); Gibbs and Su (2002). The 2–Wasserstein distance, although requiring more computation, are also popular in image and geometry processing thanks to its geometric properties such as barycenters Agueh and Carlier (2011); Solomon *et al.* (2015). In this paper, we focus on 2–Wasserstein distances.

The K-means clustering method goes back to Lloyd Lloyd (1982) and Forgy Forgy (1965). Its connections to the $1, 2$-Wasserstein metrics were leveraged in Ho *et al.* (2017) and Applegate *et al.* (2011), respectively. The essential idea is to use a sparse discrete point set to cluster denser or continuous distributional data with respect to the Wasserstein distance between the original data and the sparse representation, which is equivalent to finding a Wasserstein barycenter of a single distribution Cuturi and Doucet (2014). A few other works have also contributed to this problem by proposing fast optimization methods, e.g., Ye *et al.* (2017).

In this paper, we approach the K-means problem from the perspective of optimal transportation in the variational principle. Because we leverage power Voronoi diagrams to compute optimal transportation, we simultaneously pursue the Wasserstein distance and K-means clustering. We compare our method with others through empirical experiments and demonstrate its applications in different fields of computer vision and machine learning research.

2.3 Primer on Optimal Transportation and Notations

We begin by iterating key concepts of optimal transportation (OT) and Wasserstein barycenters (WBs). Suppose μ, ν are *Borel probability distributions* supported in *Polish spaces* $\mathcal{X}(x)$, $\mathcal{Y}(y)$, respectively. Let $\mathcal{P}(\mathcal{X} \times \mathcal{Y})$ be the set of all probability distributions on $\mathcal{X} \times \mathcal{Y}$. Then, we denote by $\Pi(\mu,\nu) = \{\pi \in \mathcal{P}(\mathcal{X} \times \mathcal{Y}) \mid \int_{\mathcal{X}} d\pi(x,y) = d\nu(y), \int_{\mathcal{Y}} d\pi(x,y) = d\mu(x)\}$ the set of all transportation maps π between μ and ν. Thus, π is also the joint distribution of μ and ν, and $d\pi(x,y)$ specifies the *mass* transported between x and y. In addition, we use $c(x,y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{\geq 0}$ to specify the transportation cost between x and y.

The OT problem is to minimize the total transportation cost:

$$\min_{\pi \in \Pi(\mu,\nu)} I_1[\pi] = \int_{\mathcal{X} \times \mathcal{Y}} c(x,y)^p d\pi(x,y),$$

where $p \in [1, \infty)$ indicates the moment of the cost function. Then, we call this minimum the *p-Wasserstein distance*:

$$\mathcal{W}_p(\mu,\nu) = \inf_{\pi \in \Pi(\mu,\nu)} \left(I_1[\pi]\right)^{1/p}.$$

The above is the well-known Kantorovich's OT formulation that admits a partial map that splits the measure $d\mu(x)$ during transportation. In Monge's original version, each location x has a unique correspondence y. If we define such a map as $T : \mathcal{X} \to \mathcal{Y}$, then we have $d\pi_T(x,y) \equiv d\mu(x)\delta_y(T(x))$ and Monge OT:

$$T^* = \underset{\pi_T \in \Pi(\mu,\nu)}{\arg\min}\; I_1[\pi_T] \equiv \int_{\mathcal{X}} c(x,T(x))^p d\mu(x) \tag{2.1}$$

T *pushes forward* μ to ν, i.e. $\nu = T\#\mu$; more rigorously, for any measurable set $B \subset \mathcal{Y}$, $\nu[B] = \mu[T^{-1}(B)]$. In other words, T preserves measure, or T is measure-preserving. We direct readers to Villani (2003); Peyré *et al.* (2019) for more on OT. In this paper, we compute Monge OT. In particular, we narrow our discussion to $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^n$, $c(x,y) = \|x-y\|_2$, and $p = 2$ unless specified otherwise. Hence, we compute $\mathcal{W}_2$.

The Wasserstein distance (WD) satisfies all metric properties Villani (2003). The *fréchet mean* of a collection of distributions $\boldsymbol{\mu}_{1:N} \overset{\text{def}}{=} \{\mu_i\}_{i=1}^N$ w.r.t the WD is called the *Wasserstein barycenter* (WB). It is the minimizer of the weighted average:

$$\nu^* = \underset{\nu \in \mathcal{P}(\mathcal{Y})}{\arg\min} \sum_{i=1}^{N} \lambda_i \mathcal{W}_2^2(\mu_i, \nu), \tag{2.2}$$

for $\lambda_i \in [0,1]$ and $\sum_i \lambda_i = 1$. We simplify (2.2) by assuming uniform weights and rewrite it as

$$\nu^* = \underset{\nu \in \mathcal{P}(\mathcal{Y})}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} \int_{\mathcal{X}_i} \|x - T_i^*(x)\|_2^2 d\mu_i(x), \tag{2.3}$$

s.t. $T_i^*\#\mu_i = \nu$ is OT for all i. Suppose the barycenter ν is supported on K discrete atoms $\boldsymbol{y} = \{y_k\}_{k=1}^K$. If we fix $\nu(y_k)$ and only allow updating $\boldsymbol{y}$, then readers can notice that (2.3) is simultaneously solving N *constrained K-means* problems using the same set of centroids with fixed capacity, $\boldsymbol{\nu} = \{\nu_k\}_{k=1}^K$. T_i^* serves as the optimal *assignment function* in each K-means problem. Note that $T_i^*(x)$ is a *hard assignment* that has only one target because we solve Monge OT.

To clarify notation, we use ν to denote a probability distribution, continuous or discrete. If it is discrete, i.e., a set of Dirac measures, then we use $\boldsymbol{y}$ and $\boldsymbol{\nu}$ to denote its support and measure. y_k and ν_k specify the support and measure of the

k^{th} Dirac measure. We use $|\cdot|$ to denote the cardinality of the discrete distribution. To simplify our discussion, we only consider positive Dirac measures when counting the cardinality, i.e. $\nu_k > 0 \ \forall \ k$.

Given the empirical observations $\{(x_k, \mu_k)\}$ of a probability distribution $X(x, \mu)$, the K-means clustering problem seeks to assign a cluster centroid (or prototype) $y_k = y(x_i)$ with label $j = 1, ..., k$ to each empirical sample x_i in such a way that the error function (2.4) reaches its minimum and meanwhile the measure of each cluster is preserved, i.e. $\nu_k = \sum_{y_k = y(x_i)} \mu_i$. It is equivalent to finding a partition $\mathcal{R} = \{(\mathcal{R}_k, y_k)\}$ of the embedding space M. If M is convex, then so is $\mathcal{R}_k$.

$$\arg\min_{\mathcal{R}} \sum_{x_i} \mu_i d(x_j, y(x_j))^p \ \equiv \ \arg\min_{\mathcal{R}} \sum_{k=1}^{K} \sum_{x_i \in \mathcal{R}_k} \mu_i d(x_j, y(\mathcal{R}_k))^p. \qquad (2.4)$$

Such a clustering problem (2.4), when ν is fixed, is equivalent to Monge's OT problem (2.1) when the support of y is sparse and not fixed because π and V induce each other, i.e. $\pi \Leftrightarrow V$. Therefore, the solution to (2.4) comes from the optimization in the search space $\mathcal{P}(\pi, y)$. Note that when ν is not fixed such a problem becomes *the Wasserstein barycenter* problem as finding a minimum in $\mathcal{P}(\pi, y, \nu)$, studied in Agueh and Carlier (2011); Cuturi and Doucet (2014); Ye *et al.* (2017).

2.4 Optimal Transportation through Variational Principles

Directly computing a Monge map is highly intractable and variational methods have been adopted by most researchers. De Goes *et al.* (2012); Gu *et al.* (2013); Lévy (2015) offer three variational formulations. We follow Gu *et al.* (2013) and refer to it as *variational OT* or VOT.

Suppose ν is supported on K discrete atoms $\boldsymbol{y} = \{y_k\}_{k=1}^{K} \subset \mathcal{Y}$, that is, ν is a set of K Dirac measures: $\nu = \{\nu_k \delta_y(y_k)\}_{k=1}^{K}$. The problem becomes *semi-discrete OT.*

We present the variational principle for solving the optimal transportation prob-

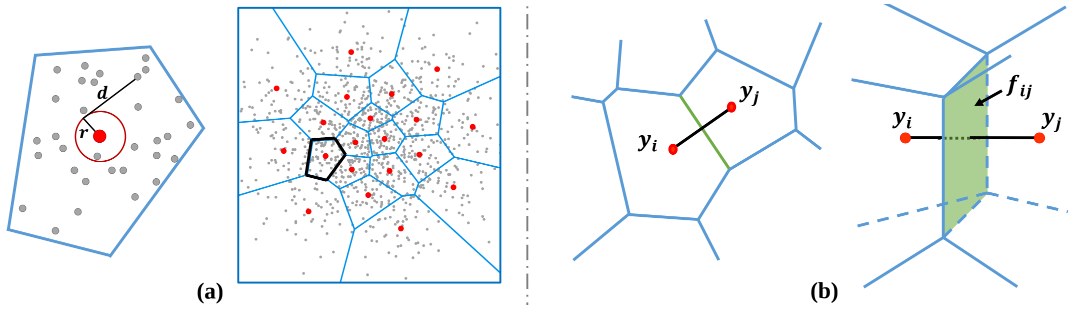


Figure 2.1: (a) Power Voronoi Diagram. Red Dots Are Centroids of the Voronoi Cells, or Clusters. The Power Distances Has an Offset Depending on the Weight of the Cell. (b) Intersection of Adjacent Cells in 2D and 3D for Computing Hessian.

lem. Given a metric space M, a Borel probability measure $X(x, \mu)$, and its compact support $\Omega = supp\ \mu = \{x \in M \mid \mu(x) > 0\}$, we consider a sparsely supported point set with Dirac measure $Y(y, \nu) = \{(y_k, \nu_k > 0)\}$, $j = 1, ..., k$. (Strictly speaking, the empirical measure $X(x, \mu)$ is also a set of Dirac measures but in this paper we refer to X as the empirical measure and Y as the Dirac measure for clarity.) Our goal is to find an optimal transportation plan or map (OT-map), $\pi : x \to y$, with the *push-forward* measure $\pi_{\#}\mu = \nu$. This is *semi-discrete OT*.

We introduce a vector $\boldsymbol{h} = (h_1, ..., h_k)'$, a hyperplane on M, $\gamma_k(\boldsymbol{h}) : \langle m, y_i \rangle + h_k = 0$, and a piece-wise linear function:

$$\theta_{\boldsymbol{h}}(x) = \max\{\langle x, y_k \rangle + h_k\},\ \ k = 1, ..., K.$$

Theorem 1. *(Alexandrov Alexandrov (2005)) Suppose Ω is a compact convex polytope with non-empty interior in $\mathbb{R}^n$ and $\{y_1, ..., y_K\} \subset \mathbb{R}^n$ are K distinct points and $\nu_1, ..., \nu_K > 0$ so that $\sum_{k=1}^{K} \nu_k = vol(\Omega)$. There exists a unique vector $\boldsymbol{h} = (h_1, ..., h_K)' \in \mathbb{R}^K$ up to a translation factor $(c, ..., c)'$ such that the piece-wise linear convex function $\theta_{\boldsymbol{h}}(x) = \max\{\langle x, y_k \rangle + h_k\}$ satisfies $vol(x \in \Omega \mid \nabla\theta_{\boldsymbol{h}}(x) = y_k) = \nu_k$.*

Furthermore, Brenier Brenier (1991) proved that the gradient map $\nabla\theta$ provides the solution to Monge's OT problem, that is, $\nabla\theta_{\boldsymbol{h}}$ minimizes the transportation cost $\int_{\Omega}\|x-\nabla\theta_{\boldsymbol{h}}(x)\|^2$. Therefore, given X and Y, $\boldsymbol{h}$ by itself induces OT.

From Aurenhammer (1987), we know that a convex subdivision associated to a piecewise-linear convex function $u_{\boldsymbol{h}}(x)$ on $\mathbb{R}^n$ equals a *power Voronoi diagram*, or *power diagram*. A typical power diagram on $M \subset \mathbb{R}^n$ can be represented as:

$$V_j \overset{\text{def}}{=} \{m \in M \mid \|m-y_k\|^2 - r_j^2 \leqslant \|m-y_\ell\|^2 - r_i^2\},\ \forall k \neq \ell.$$

Then, a simple calculation gives us

$$m \cdot y_k - \frac{1}{2}(y_k \cdot y_k + r_k^2) \leqslant m \cdot y_\ell - \frac{1}{2}(y_\ell \cdot y_\ell + r_\ell^2),$$

where $m \cdot y_k = \langle m, y_k\rangle$ and w_j represents the offset of the *power distance* as shown in Fig. 2.1 (a). On the other hand, the graph of the hyperplane $\gamma_k(\boldsymbol{h})$ is

$$U_i \overset{\text{def}}{=} \{m \in M \mid \langle m, y_k\rangle - h_k \geqslant \langle m, y_\ell\rangle - h_\ell\},\ \forall k \neq \ell.$$

Thus, we obtain the connection between $\boldsymbol{h}$ and the power diagram: $h_k = \frac{r_k^2 - |y_k|^2}{2}$.

We substitute $M(m)$ with the measure $X(x)$. In our formulation, Brenier's gradient map $\nabla\theta_{\boldsymbol{h}} : \mathcal{R}_k(\boldsymbol{h}) \to y_k$ "transports" each $\mathcal{R}_k(\boldsymbol{h})$ to a specific point y_k. The total mass of $\mathcal{R}_k(\boldsymbol{h})$ is denoted as: $w_j(\boldsymbol{h}) = \sum_{x\in\mathcal{R}_k(\boldsymbol{h})}\mu(x)$.

Now, we introduce an energy function:

$$I_2[\boldsymbol{h}] \overset{\text{def}}{=} \int_{\boldsymbol{0}}^{\boldsymbol{h}} \sum_{k=1}^{K} \int_{\mathcal{R}_k} d\mu(x) dh_k - \sum_{k=1}^{K} \nu_k h_k, \tag{2.5}$$

whose gradient, $\left\{\int_{\mathcal{R}_k} d\mu(x) - \nu_k\right\}_k$, also integrates to

$$I_3[\boldsymbol{h}] \overset{\text{def}}{=} \int_{\mathcal{X}} \theta_{\boldsymbol{h}}(x) d\mu(x) - \sum_{k=1}^{K} \nu_k h_k. \tag{2.6}$$

The differentiability of I_2 w.r.t. $\boldsymbol{h}$ has been discussed in Gu *et al.* (2013). Its gradient and Hessian are then given by

Algorithm 1: Variational Optimal Transportation

```
Function Variational-OT($X(x, \mu)$, $Y(y, v)$, $\epsilon$)
    $\boldsymbol{h} \leftarrow \mathbf{0}$.
    repeat
        Update power diagram $V$ with $(y, \boldsymbol{h})$.
        Compute cell weight $w(\boldsymbol{h}) = \{\sum_{m \in V_j} \mu(m)\}$.
        Compute gradient $\nabla I_2(\boldsymbol{h})$ and Hessian $H$ using Equation (2.7) and
         (2.8).
        $\boldsymbol{h} \leftarrow \boldsymbol{h} - H^{-1}\nabla I_2(\boldsymbol{h})$. // Update the minimizer $\boldsymbol{h}$ according to (2.9)
    until $|\nabla I_2(\boldsymbol{h})| < \epsilon$.
    return $V, \boldsymbol{h}$.
end
```

$$\nabla I_2(\boldsymbol{h}) = (w_1(\boldsymbol{h}) - \nu_1, ..., w_k(\boldsymbol{h}) - \nu_k)^T, \tag{2.7}$$

$$H = \frac{\partial^2 E(\boldsymbol{h})}{\partial h_i \partial h_k} = \begin{cases} \sum_l \dfrac{\int_{f_{il}} \mu(x)dx}{\|y_l - y_\ell\|}, & i = j, \ \forall l, s.t. \ f_{il} \neq \emptyset, \\ -\dfrac{\int_{f_{ij}} \mu(x)dx}{\|y_k - y_\ell\|}, & i \neq j, \ f_{ij} \neq \emptyset, \\ 0, & i \neq j, \ f_{ij} = \emptyset, \end{cases} \tag{2.8}$$

where $\|\cdot\|$ is the $L1$–norm and $\int_{f_{ij}} \mu(x)dx = \text{vol}(f_{ij})$ is the volume of the intersection f_{ij} between two adjacent cells. Fig. 2.1 (b) illustrates the geometric relation. The Hessian H is positive semi-definite with only constant functions spanned by a vector $(1, ..., 1)^T$ in its null space. Thus, I_2 is strictly convex in the space of $\boldsymbol{h}$. By Newton's method, we solve a linear system,

Algorithm 2: Iterative Measure-Preserving Mapping

```
Function Iterative-Measure-Preserving-Mapping($X(x,\mu), Y(y,\nu)$)
    repeat
        $V(\boldsymbol{h}) \leftarrow$ Variational-OT($x, \mu, y, \nu$). // 1. Update Voronoi partition
        $y_k \leftarrow \sum_{x \in V_j} \mu_k x_k / \sum_{x \in V_j} \mu_i$. // 2. Update $y$
    until $y$ converges.
    return $y, V$.
end
```

$$H\delta\boldsymbol{h} = \nabla I_2(\boldsymbol{h}), \tag{2.9}$$

and update $\boldsymbol{h}^{(t+1)} \leftarrow \boldsymbol{h}^{(t)} + \delta\boldsymbol{h}^{(t)}$. The energy I_2 (2.5) is motivated by Theorem 1 which seeks a solution to $vol(x \in \Omega \mid \nabla\theta_{\boldsymbol{h}}(x) = y_k) = \nu_k$. Move the right-hand side to left and take the integral over $\boldsymbol{h}$ then it becomes I_2 (2.5). Thus, minimizing (2.5) when the gradient approaches $\mathbf{0}$ gives the solution. We show the complete algorithm for obtaining the OT-Map $\pi : X \to Y$ in Alg. 1. Later in the chapter, we discuss the connection between the energy of variational OT and the classic OT problem so that we better understand how we "variationally" minimize $I_2[\boldsymbol{h}]$, (2.5), for a *height vector* $\boldsymbol{h}$ and that will produce a Monge map T^*.

We now introduce in detail our method to solve clustering problems through variational optimal transportation. We name it *variational Wasserstein clustering* (VWC). We focus on the semi-discrete clustering problem which is to find a set of discrete sparse centroids to best represent a continuous probability measure, or its discrete empirical representation. Suppose M is a metric space and we embody in it an empirical measure $X(x, \mu)$. Our goal is to find such a sparse measure $Y(y, \nu)$ that minimizes (2.4).

We begin with an assumption that the distributional data are embedded in the same Euclidean space $M = \mathbb{R}^n$, i.e. $X, Y \in \mathcal{P}(M)$. We observe that if ν is fixed then (2.1) and (2.4) are mathematically equivalent. Thus, the computational approaches to these problems could also coincide. Because the space is convex, each cluster is eventually a Voronoi cell and the resulting partition $V = \{(V_j, y_k)\}$ is actually a power Voronoi diagram where we have $\|x - y_k\|^2 - r_j^2 \leq \|x - y_\ell\|^2 - r_i^2$, $x \in V_j$, $\forall j \neq i$ and r is associated with the total mass of each cell. Such a diagram is also the solution to Monge's OT problem between X and Y. From the previous section, we know that if we fix X and Y, the power diagram is entirely determined by the minimizer $\boldsymbol{h}$. Thus, assuming ν is fixed and y is allowed to move freely in M, we reformulate (2.4) to

$$f(\boldsymbol{h}, y) = \sum_{j=1}^{K} \sum_{x_i \in V_j(\boldsymbol{h})} \mu_i \|x_i - y_k\|^2, \tag{2.10}$$

where every V_j is a power Voronoi cell.

The solution to Eq. (2.10) can be achieved by iteratively updating $\boldsymbol{h}$ and y. While we can use Alg. 1 to compute $\boldsymbol{h}$, updating y can follow the rule:

$$y_k^{(t+1)} \leftarrow \sum \mu_i x_i^{(t)} \Big/ \sum \mu_i, \;\; x_i^{(t)} \in V_j. \tag{2.11}$$

Since the first step preserves the measure and the second step updates the measure, we call such a mapping an *iterative measure-preserving mapping.* Our algorithm repeatedly updates the partition of the space by variational-OT and computes the new centroids until convergence, as shown in Alg. 2. Furthermore, because each step reduces the total cost (2.10), we have the following propositions.

Proposition 1. *Alg. 2 monotonically minimizes the objective function (2.10).*

Proof. It is sufficient for us to show that for any $t \geq 0$, we have

$$f(\boldsymbol{h}^{(t+1)}, y^{(t+1)}) \leq f(\boldsymbol{h}^{(t)}, y^{(t)}). \tag{2.12}$$

Algorithm 3: Variational Wasserstein Clustering

```
Input   : Empirical measures $X_M(x, \mu)$ and $Y_N(y, \nu)$
Output: Measure-preserving Map $\pi : X \to Y$ represented as $(y, V)$.
begin
    $\nu \leftarrow$ Sampling-known-distribution. // Initialization.
    Harmonic-mapping: $M, N \to \mathbb{R}^n$ or $\mathbb{D}^n$. // Unify domains.
    $y, V \leftarrow$ Iterative-Measure-Preserving-Mapping($x, \mu, y, \nu$).
end
return $y, V$.
```

The above inequality is indeed true since $f(\boldsymbol{h}^{(t+1)}, y^{(t)}) \leq f(\boldsymbol{h}^{(t)}, y^{(t)})$ according to the convexity of our OT formulation, and $f(\boldsymbol{h}^{(t+1)}, y^{(t+1)}) \leq f(\boldsymbol{h}^{(t+1)}, y^{(t)})$ for the updating process itself minimizes the mean squared error. □

Corollary 1. *Alg. 2 converges in a finite number of iterations.*

Proof. We borrow the proof for K-means. Given N empirical samples and a fixed number K, there are K^N ways of clustering. At each iteration, Alg. 2 produces a new clustering rule only based on the previous one. The new rule induces a lower cost if it is different than the previous one, or the same cost if it is the same as the previous one. Since the domain is a finite set, the iteration must eventually enter a cycle whose length cannot be greater than 1 because otherwise it violates the fact of the monotonically declining cost. Therefore, the cycle has a length of 1 in which case the Alg. 2 converges in a finite number of iterations. □

Now, we introduce the concept of variational Wasserstein clustering. For a subset $M \subset \mathbb{R}^n$, let $\mathcal{P}(M)$ be the space of all Borel probability measures. Suppose $X(x, \mu) \in \mathcal{P}(M)$ is an existing one and we are to aggregate it into k clusters represented by

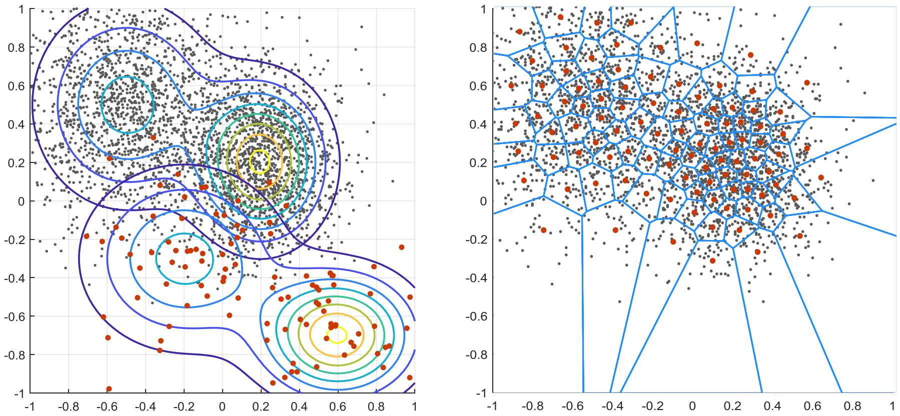


Figure 2.2: Given the Source Domain (Red Dots) and the Target Domain (Grey Dots), the Distribution of the Source Samples Are Driven Into the Target Domain and Form a Power Voronoi Diagram.

another measure $Y(y,\nu) \in \mathcal{P}(M)$ and assignment $y_k = \pi(x)$, $j = 1, ..., k$. Thus, we have $\pi \in \mathcal{P}(M \times M)$. Given ν fixed, our goal is to find such a combination of Y and π that minimize the object function:

$$Y_{y,\nu} = \underset{\substack{Y \in P(M) \\ \pi \in P(M \times M)}}{argmin} \sum_{j=1}^{k} \sum_{y_k = \pi(x_i)} \mu_i \|x_i - y_k\|^2, \; s.t. \; \nu_k = \sum_{y_k = \pi(x_i)} \mu_i. \tag{2.13}$$

(2.13) is not convex w.r.t. y as discussed in Cuturi and Doucet (2014). We solve it by iteratively updating π and y. When updating π, since y is fixed, (2.13) becomes an optimal transportation problem. Therefore, solving (2.13) is equivalent to approaching the infimum of the 2-Wasserstein distance between X and Y:

$$\inf_{\substack{Y \in P(M) \\ \pi \in P(M \times M)}} \sum_{j=1}^{k} \sum_{y_k = \pi(x_i)} \mu_i \|x_i - y_k\|^2 = \inf_{Y \in P(M)} W_2^2(X, Y). \tag{2.14}$$

Assuming the domain is convex, we can apply iterative measure-preserving mapping (Alg. 2) to obtain y and h which induces π. In case that X and Y are not in the same domain i.e. $Y(y,\nu) \in P(N)$, $N \subset \mathbb{R}^n$, $N \neq M$, or the domain is not necessarily convex, we leverage *harmonic mapping* Gu and Yau (2008); Wang *et al.* (2004) to map them to a convex canonical space. We wrap up our complete algorithm in Alg. 3. Fig. 2.2 illustrates a clustering result. Given a source Gaussian mixture (red dots) and a target Gaussian mixture (grey dots), we cluster the target domain with the source samples. Every sample has the same mass in each domain for simplicity. Thus, we obtain an unweighted Voronoi diagram. In the next section, we show examples that involve different mass. We implement our algorithm in C/C++ and adopt Voro++ Rycroft (2009) to compute Voronoi diagrams. The code is available at https://github.com/icemiliang/vot.

2.5 Connections between Variational OT and Monge OT

The *Lagrangian duality* of Monge OT (2.1) is

$$\begin{gathered}\max_{\boldsymbol{\varphi}} \ \min_{T} \ I_4[\boldsymbol{\varphi}, T] \overset{\text{def}}{=} \\ \int_{\mathcal{X}} \|x - T(x)\|_2^2 d\mu(x) + \sum_{k=1}^{K} \varphi_k \left(d\mu(x) - \nu_k\right), \end{gathered} \tag{2.15}$$

where $\boldsymbol{\varphi} = \{\varphi_k\}_{k=1}^{K}$. (2.15) simplifies to

$$\max_{\boldsymbol{\varphi}} \ I_4[\boldsymbol{\varphi}] = \sum_{k=1}^{K} \int_{\mathcal{R}'_k} \big(\|x - y_k\|_2^2 + \varphi_k\big) d\mu(x) - \sum_{k=1}^{K} \varphi_k \nu_k, \tag{2.16}$$

$\mathcal{R}'_k = \{x \in \mathcal{X} \mid \|x - y_k\|_2^2 + \varphi_k \leq \|x - y_\ell\|_2^2 + \varphi_\ell, \forall \ell \neq k\}$ which coincides with a *power Voronoi diagram.*

We prove their following connections which show that we "variationally" minimize $I_2[\boldsymbol{h}]$, (2.5), for a *height vector* $\boldsymbol{h}$ and that will produce a Monge map T^*.

Proposition 2. ***1****. The minimum point of* $I_2[\boldsymbol{h}]$, (2.5)*, also minimizes* $I_3[\boldsymbol{h}]$, (2.6)*.* ***2****.* $\mathcal{R}_k \equiv \mathcal{R}'_k$*.* ***3****.* $\boldsymbol{\mathcal{R}}$ *in* $I_2[\boldsymbol{h}]$, (2.5)*, induces the Monge map* $T : x \to y_k$*.* ***4****. Minimizing* $I_2[\boldsymbol{h}]$, (2.5)*, is equivalent to maximizing* $I_4[\boldsymbol{h}]$, (2.16)*.*

3. $\mathcal{R}_\ell \equiv \mathcal{R}_\ell^{'}$

Proof.

$$\min_T I_1[T] = \int_{\mathcal{X}} \|x - T(x)\|_2^2 d\mu(x) = \sum_{\ell=1}^{K} \int_{\mathcal{R}'_\ell} \|x - T(x)\|_2^2 d\mu(x),$$
$$s.t. \int_{\mathcal{R}'_\ell} d\mu(x) = \nu_\ell, \quad \mathcal{R}'_\ell \overset{\text{def}}{=} \{x \in \mathcal{X} \mid T(x) = y_\ell\}.$$

$$\begin{aligned}\max_{\boldsymbol{\varphi}} \min_T I_4[\boldsymbol{\varphi}, T] \overset{\text{def}}{=}& \sum_{\ell=1}^{K} \int_{\mathcal{R}'_\ell} \|x - T(x)\|_2^2 d\mu(x) + \sum_{\ell=1}^{K} \varphi_\ell \left(\int_{\mathcal{R}'_\ell} d\mu(x) - \nu_\ell \right) \\ =& \sum_{\ell=1}^{K} \int_{\mathcal{R}'_\ell} \left(\|x - T(x)\|_2^2 + \varphi_\ell \right) d\mu(x) - \sum_{\ell=1}^{K} \varphi_\ell \nu_\ell \\ =& \sum_{\ell=1}^{K} \int_{\mathcal{R}'_\ell} \left(\|x - y_\ell\|_2^2 + \varphi_\ell \right) d\mu(x) - \sum_{\ell=1}^{K} \varphi_\ell \nu_\ell.\end{aligned}$$

Note that $\boldsymbol{\mathcal{R}}$ or $\boldsymbol{\mathcal{R}}'$ is induced by T, but we omit T in the notation for simplicity.

Suppose T^* is the minimizer of $I_4[T]$, then T^* induces the graph $\mathcal{R}'_\ell = \{x \in \mathcal{X} \mid \|x - y_\ell\|_2^2 + \varphi_\ell \leq \|x - y_k\|_2^2 + \varphi_k, \ \forall k \neq \ell\}$ because otherwise there would exist $x \in \mathcal{R}_\ell$ such that $\|x - y_\ell\|_2^2 + \varphi_\ell > \|x - y_k\|_2^2 + \varphi_k$ and that is contradictory to the fact that T^* is the minimizer. Therefore, $\mathcal{R}_\ell \equiv \mathcal{R}_\ell^{'} \ \forall \ \ell$. □

Minimizing $I_2[\boldsymbol{h}]$ is equivalent to maximizing $I_4[\boldsymbol{h}]$.

$$\min \ I_3[\boldsymbol{h}] \overset{\text{def}}{=} \int_{\mathcal{X}} \theta_{\boldsymbol{h}}(x) d\mu(x) - \sum_{\ell=1}^{K} \nu(y_\ell) h_\ell.$$

$\mathcal{R}_\ell^{'} = \{x \in \mathcal{X} \mid \|x - y_\ell\|_2^2 + \varphi_\ell \leq \|x - y_k\|_2^2 + \varphi_k, \forall k \neq \ell\}$

Proof. Given that $\mathcal{R}_\ell = \mathcal{R}_\ell^{'}$,

$$\begin{aligned}\mathcal{R}_\ell =& \{x \in \mathcal{X} \mid xy_\ell + h_\ell \geq xy_k + h_k, \forall k \neq \ell\} \\ \mathcal{R}_\ell^{'} =& \{x \in \mathcal{X} \mid \|x - y_\ell\|_2^2 + \varphi_\ell \leq \|x - y_k\|_2^2 + \varphi_k, \forall k \neq \ell\} \\ =& \{x \in \mathcal{X} \mid 2xy_\ell - y_\ell^2 - \varphi_\ell \geq 2xy_k - y_k^2 - \varphi_k, \forall k \neq \ell\}\end{aligned}$$

$$\begin{aligned}\implies 2h_\ell &= -y_\ell^2 - \varphi_\ell \\ h_\ell &= -\frac{1}{2}\left(y_\ell^2 + \varphi_\ell\right) \\ \varphi_\ell &= -2h_\ell - y_\ell^2\end{aligned}$$

$$\begin{aligned}I_4[\boldsymbol{h}] =&\sum_{\ell=1}^{K}\int_{\mathcal{R}_\ell}\left(\|x-y_\ell\|_2^2+\varphi_\ell\right)d\mu(x)+\sum_{\ell=1}^{K}\varphi_\ell\nu(y_\ell) \\ =&\sum_{\ell=1}^{K}\int_{\mathcal{R}_\ell}x^2d\mu(x)-2\sum_{\ell=1}^{K}\int_{\mathcal{R}_\ell}xy_\ell d\mu(x)+\sum_{\ell=1}^{K}\int_{\mathcal{R}_\ell}y_\ell^2d\mu(x) \\ &-2\sum_{\ell=1}^{K}\int_{\mathcal{R}_\ell}h_\ell^2d\mu(x)-\sum_{\ell=1}^{K}\int_{\mathcal{R}_\ell}y_\ell^2d\mu(x)+2\sum_{\ell=1}^{K}h_\ell\nu(y_\ell)+\sum_{\ell=1}^{K}y_\ell^2\nu(y_\ell) \\ =&-2\sum_{\ell=1}^{K}\int_{\mathcal{R}_\ell}xy_\ell d\mu(x)-2\sum_{\ell=1}^{K}\int_{\mathcal{R}_\ell}h_\ell d\mu(x)+2\sum_{\ell=1}^{K}h_\ell\nu(y_\ell)+constants \\ =&-2\sum_{\ell=1}^{K}\int_{\mathcal{R}_\ell}\left(xy_\ell+h_\ell\right)d\mu(x)-2\sum_{\ell=1}^{K}h_\ell\nu(y_\ell)+constants \\ =&-2I_2[\boldsymbol{h}]+constants.\end{aligned}$$

Therefore, minimizing $I_2[\boldsymbol{h}]$ is equivalent to maximizing $I_4[\boldsymbol{h}]$. □

The minimum point of $I_2[\boldsymbol{h}]$ also minimizes $I_3[\boldsymbol{h}]$.

Proof.

$$\nabla I_3[\boldsymbol{h}] = \left\{\frac{\partial I_3[\boldsymbol{h}]}{\partial h_\ell}\right\}_{\ell=1}^{K} = \left\{\int_{\mathcal{X}}\frac{\partial\theta_{\boldsymbol{h}}}{h_\ell}d\mu(x)-\nu_\ell\right\}_{\ell=1}^{K} = \left\{\int_{\mathcal{R}_\ell}d\mu(x)-\nu_\ell\right\}_{\ell=1}^{K}$$

$$\nabla I_2[\boldsymbol{h}] = \left\{\frac{\partial I_2[\boldsymbol{h}]}{\partial h_\ell}\right\}_{\ell=1}^{K} = \left\{\int_{\mathcal{R}_\ell}d\mu(x)-\nu_\ell\right\}_{\ell=1}^{K}$$

Therefore, $\nabla I_2[\boldsymbol{h}] = \nabla I_3[\boldsymbol{h}]$. As per proved in Gu *et al.* (2013), both $I_2[\boldsymbol{h}]$ and $I_3[\boldsymbol{h}]$ are strictly convex which means their minimum points are $\nabla I_2[\boldsymbol{h}] \to \mathbf{0}$ or

$\nabla I_3[\boldsymbol{h}] \to \mathbf{0}$. In order to solve $I_3[\boldsymbol{h}]$, we can then instead solve $I_2[\boldsymbol{h}]$ for the optimal $\boldsymbol{h}^*$. □

$\boldsymbol{\mathcal{R}}$ in $I_2[\boldsymbol{h}]$ induces the Monge map $T : x \to y_\ell$

Proof.

$$\min \ I_2[\boldsymbol{h}] \overset{\text{def}}{=} \int_{\mathbf{0}}^{\boldsymbol{h}} \sum_{\ell=1}^{K} \int_{\mathcal{R}_\ell} d\mu(x) dh_\ell - \sum_{\ell=1}^{K} \nu(y_\ell) h_\ell,$$

$$\mathcal{R}_\ell = \{x \in \mathcal{X} \mid xy_\ell + h_\ell \geq xy_k + h_k, \forall k \neq \ell\}.$$

This is indeed true since $\boldsymbol{\mathcal{R}} = \boldsymbol{\mathcal{R}}'$ and $\boldsymbol{\mathcal{R}}'$ induces the Monge map. □

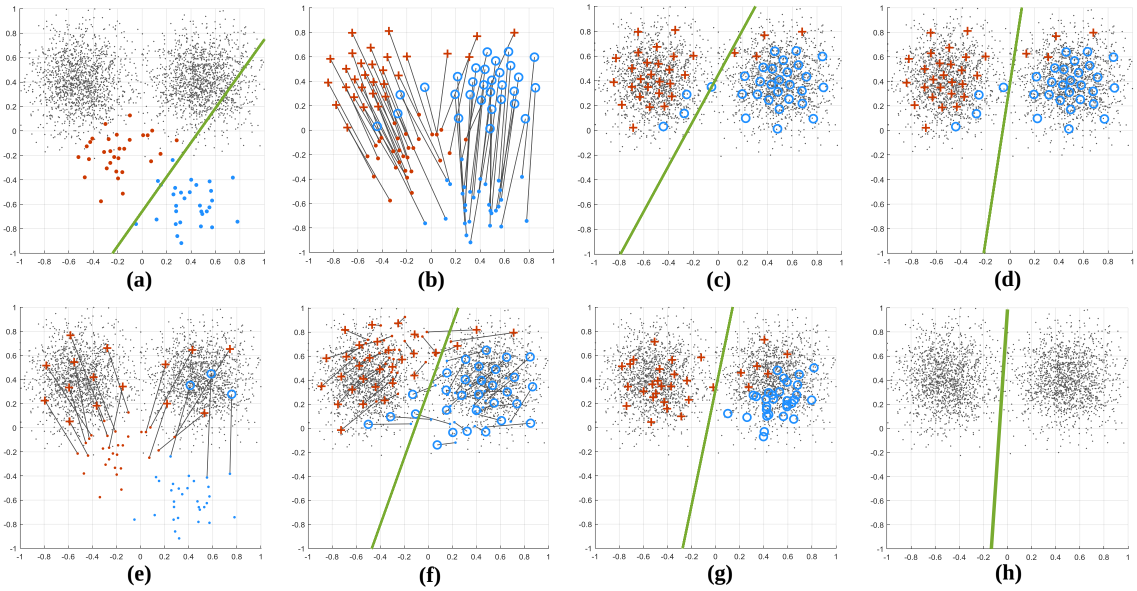
2.6 Experiments

While the K-means clustering problem is ubiquitous in numerous tasks in computer vision and machine learning, we present the use of our method in approaching domain adaptation, remeshing, and representation learning.

2.6.1 Domain Adaptation on Synthetic Data

Domain adaptation plays a fundamental role in knowledge transfer and has benefited many different fields, such as scene understanding and image style transfer. Several works have coped with domain adaptation by transforming distributions to close their gap with respect to a measure. In recent years, Courty *et al.* Courty *et al.* (2014) took the first steps in applying optimal transportation to domain adaptation. Here we revisit this idea and provide our own solution to *unsupervised many-to-one domain adaptation* based on variational Wasserstein clustering.

Consider a two-class classification problem in the 2D Euclidean space. The source domain consists of two independent Gaussian distributions sampled by red and blue dots, as shown in Fig. 2.3 (a). Each class has 30 samples. The target domain has two other independent Gaussian distributions with different means and variances, each



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 2.3: SVM RBF Boundaries for Domain Adaptation. (a) Target Domain in Gray Dots and Source Domain of Two Classes in Red and Blue Dots; (b) Mapping of Centroids by Using VWC; (c, d) Boundaries From VWC With Linear and RBF Kernels; (e) K-means++ Arthur and Vassilvitskii (2007) Fails to Produce a Model; (f) After Recentering Source and Target Domains, K-Means++ Yields Acceptable Boundary; (g) D2 Ye *et al.* (2017); (h) JDOT Courty *et al.* (2017a), Final Centroids Not Available.

having 1500 samples. They are represented by denser gray dots to emulate the source domain after an unknown transformation.

We adopt support vector machine (SVM) with linear and radial basis function (RBF) kernels for classification. The kernel scale for RBF is 5. One can notice that directly applying the RBF classifier learned from the source domain to the target domain provides a poor classification result (59.80%). While Fig. 2.3 (b) shows the final positions of the samples from the source domain by VWC, (c) and (d) show the decision boundaries from SVMs with a linear kernel and an RBF kernel, respectively. In (e) and (f) we show the results from the classic K-means++ method Arthur and

Vassilvitskii (2007). In (e), K-means++ fails to cluster the unlabeled samples into the original source domain and produces an extremely biased model that has and accuracy of50%. Only after we recenter the source and the target domains yields K-means++ better results, as shown in (f).

For more comparison, we test two other methods – D2 Ye *et al.* (2017) and JDOT Courty *et al.* (2017a). The final source positions from D2 are shown in (g). Because D2 solves the general barycenter problem and also updates the weights of the source samples, it converges as soon as it can find them some positions when the weights can also satisfy the minimum clustering loss. Thus, in (g), most of the source samples dive into the right, closer density, leaving those moving to the left with larger weights. We show the decision boundary obtained from JDOT Courty *et al.* (2017a) in (h). JDOT does not update the centroids, so we only show its decision boundary. In this experiment, both our method for Monge's OT and the methods Courty *et al.* (2017a); Ye *et al.* (2017) for Kantorovich's OT can effectively transfer knowledge between different domains, while the traditional method Arthur and Vassilvitskii (2007) can only work after a prior knowledge between the two domains, e.g., a linear offset. Detailed performances are reported in Tab. 2.1.

Table 2.1: Classification Accuracy On Synthetic Data

	K-means++*	K-means++^{r}		D2		JDOT		VWC	
Kernel	Linear/RBF	Linear	RBF	Linear	RBF	Linear	RBF	Linear	RBF
Acc.	50.00	97.88	99.12	95.85	99.25	99.03	99.23	98.56	99.31
Sen.	100.00	98.13	98.93	99.80	99.07	98.13	99.60	98.00	99.07
Spe.	0.00	97.53	99.27	91.73	99.40	99.93	98.87	99.07	99.53

*: extremely biased model labeling all samples with same class; ^{r}: after recenterd.

2.6.2 Deforming Triangle Meshes

Triangle meshes are a dominant approximation of surfaces. Refining triangle meshes to best represent surfaces has been studied for decades, including Shewchuk (2002); Fabri and Pion (2009); Goes *et al.* (2014). Given limited storage, we prefer to use denser and smaller triangles to represent the areas with relatively complicated geometry and sparser and larger triangles for flat regions. We follow this direction and propose to use our method to solve this problem. The idea is to drive the vertices toward high-curvature regions.

We consider a genus-zero surface $\mathbb{S}^2$ with a boundary approximated by a triangle mesh $T_{\mathbb{S}^2}(v)$. To drive the vertices to high-curvature positions, our idea, in general, is to reduce the areas of the triangles in there and increase them in those locations of low curvature, producing a new triangulation $T'_{\mathbb{S}^2}(v)$ on the surface. To avoid computing the geodesic on the surface, we first map the surface to a *unit disk* $\phi : \mathbb{S}^2 \to \mathbb{D}^2 \subset \mathbb{R}^2$ and equip it with the Euclidean metric. We drop the superscripts 2 for simplicity. To clarify notations, we use $T_{\mathbb{S}}(v)$ to represent the original triangulation on surface $\mathbb{S}$; $T_{\mathbb{D}}(v)$ to represent its counterpart on $\mathbb{D}$ after harmonic mapping; $T'_{\mathbb{D}}(v)$ for the target triangulation on $\mathbb{D}$ and $T'_{\mathbb{S}}(v)$ on $\mathbb{S}$. Fig. 2.4 (a) and (b) illustrate the triangulation before and after the harmonic mapping. Our goal is to rearrange the triangulation on $\mathbb{D}$, and then the following composition gives the desired triangulation on the surface:

$$T_{\mathbb{S}}(v) \xrightarrow{\phi} T_{\mathbb{D}}(v) \xrightarrow{\pi} T'_{\mathbb{D}}(v) \xrightarrow{\phi^{-1}} T'_{\mathbb{S}}(v).$$

π is where we apply our method.

Suppose we have an original triangulation $T_{sub,\mathbb{S}}(v)$ and an initial downsampled version $T_{\mathbb{S}}(v)$ and we map them to $T_{sub,\mathbb{D}}(v)$ and $T_{\mathbb{D}}(v)$, respectively. The vertex area $A_{\mathbb{D}} : v \to a$ on $\mathbb{D}$ is the source (Dirac) measure. We compute the (square root of

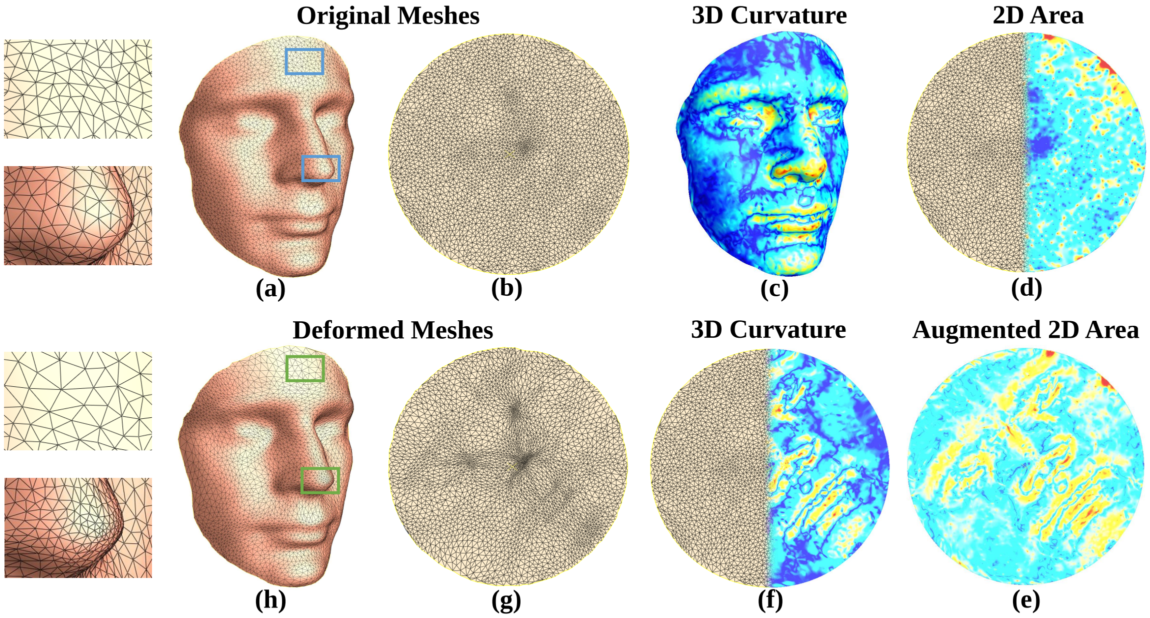


Figure 2.4: Redistribute Triangulation Based On Curvature. Original Mesh (a) Is Mapped To A Unit Disk (b). Mean Curvature On The 3D Mesh (c) Is Copied To The Disk (f). Design An "Augmented" Measure μ (e) On The Disk By Incorporating Curvature C Into 2D Vertex Area A (d), e.g. $\mu = 0.4a + 0.6C$. A Vertex y With a Large Curvature C, in Order to Maintain Its Original Measure A, Will Shrink Its Own Cluster. As a Result Vertices Collapse in High-Curvature Regions (g). Mesh Will Be Pulled Back to 3D (h) by Inverse Mapping.

absolute) mean curvature $C_{sub,\mathbb{S}} : v_{sub} \to c_{sub}$ on $\mathbb{S}$ and the area $A_{sub,\mathbb{D}} : v_{sub} \to a_{sub}$ on $\mathbb{D}$. After normalizing a and c, a weighted summation gives us the target measure, $\mu_{sub,\mathbb{D}} = (1 - \lambda)\ a_{sub,\mathbb{D}} + \lambda\ c_{sub,\mathbb{D}}$. We start from the source measure (v, a) and cluster the target measure (v_{sub}, μ_{sub}). The intuition is the following. If $\lambda = 0$, $\mu_{i,sub} = a_{i,sub}$ everywhere, then a simple unweighted Voronoi diagram which is the dual of $T_{\mathbb{D}}(v)$ would satisfy Eq. (2.14). As λ increases, the clusters $V_j(v_j, a_j)$ in the high-curvature $(c_{sub,\mathbb{D}})$ locations will require smaller areas $(a_{sub,\mathbb{D}})$ to satisfy $a_j = \sum_{v_{i,sub} \in V_j} \mu_{i,sub}$.

We apply our method on a human face for validation and show the result in

Fig. 2.4. On the left half, we show the comparison before and after the remeshing. The tip of the nose has more triangles due to the high curvature while the forehead becomes sparser because it is relatively flatter. The right half of Fig. 2.4 shows different measures that we compute for moving the vertices. (c) shows the mean curvature on the 3D surface. We map the triangulation with the curvature onto the planar disk space (f). (d) illustrates the vertex area of the planar triangulation and (e) is the weighted combination of 3D curvature and 2D area. Finally, we regard area (d) as the source domain and the "augmented" area (e) as the target domain and apply our method to obtain the new arrangement (g) of the vertices on the disk space. After that, we pull it back to the 3D surface (h). As a result, vertices are attracted to high-curvature regions. Note the boundaries of the deformed meshes (g,h) have changed after the clustering. We could restrict the boundary vertices to move *on* the unit circle if necessary. Rebuilding a Delaunay triangulation from the new vertices is also an optional step after.

2.6.3 Applications to MR Images for Alzheimer's Disease Analysis

We perform cross-sectional studies on sMRI to evaluate the performance of our framework on clinical datasets. The dataset is from Alzheimer's Disease Neuroimaging Initiative (ADNI) (adni.loni.usc.edu). The cohort information is detailed in Table 2.2. We perform 5-fold cross-validation on classification using the linear support vector machine (SVM).

Table 2.2: Demographic Information

Group	#	Sex (F/M)	Age	MMSE
AD	146	71 / 75	74.2 ± 7.0	22.6 ± 3.1
NC	175	96 / 79	74.4 ± 7.8	29.1 ± 1.3

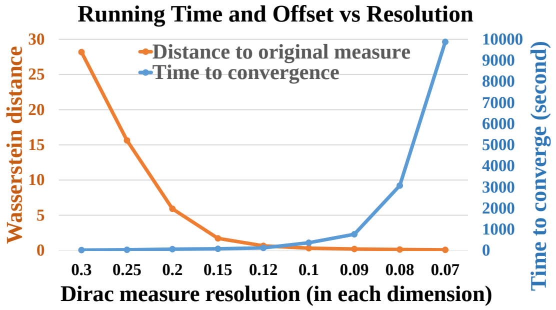


Figure 2.5: The Wasserstein Distance Between the Dirac Measure Template and the Original Image and the Time to Compute It Under Different Number of Dirac Measures.

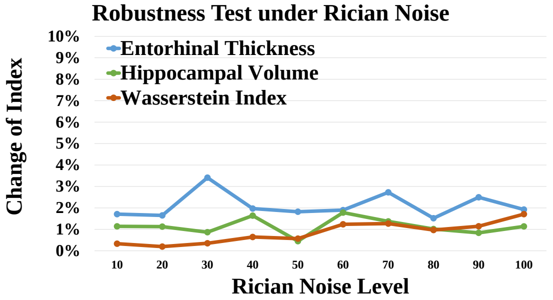


Figure 2.6: The Vibration Under Rician Noises of Three sMRI Indices -Wasserstein Index (Orange), Entorhinal Cortex Thickness (Blue), and Hippocampal Volume (Green).

Data used in this paper were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu) Jagust *et al.* (2010) Jack Jr *et al.* (2008). The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial MRI, PET, other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early AD.

We test the performance of our algorithm w.r.t. the number of Dirac measures. We cluster MNI152 into sparse Dirac measures and treat them as the template for

Table 2.3: Offset Wasserstein Distances In Different Resolutions

# of Dirac	171	257	515	1237	2401	4100	5695	8063	12020
Resolution	0.3	0.25	0.2	0.15	0.12	0.1	0.09	0.08	0.07
Offset WD	28.2	15.6	5.92	1.71	0.651	0.321	0.202	0.131	0.074

computing the Wasserstein index. Inevitably, there is an offset between the original brain template image and *our* template. We test the running time and this offset under different numbers of Dirac measures, or Voronoi cells. After resampling and harmonic mapping, MNI152 has $90,045$ samples inside a unit ball. The resolution of the Dirac ranges from 0.4 to 0.07, corresponding to the number of cells from 171 to 12020. We report the results in Table 2.3 and Figure 2.5. All the tests were run on a 3.40 GHz Intel i7-4770 CPU (single-core) with 8.00 GB RAM.

After resolution reaches 0.1, there is no significant improvement of the offset, but the computational cost boosts to thousands of seconds. To trade off between effectiveness and efficiency, we choose to use the Dirac template with a resolution of 0.1 for our following experiments. It contains $4,100$ measure points, or Voronoi cells.

We randomly select an sMRI image from our dataset to test the robustness of our proposed index to noises. We add noises to the image, producing several noisy samples and compute their Wasserstein indices (WIs). We expect the WIs to imply that the noisy samples are very close to the original sample. We also study the entorhinal cortex thickness (ECT) and hippocampal volume (HPV) for comparison. We removed the effect of brain size when calculating ECT and HPV.

Gudbjartsson and Patz (1995) suggested that the noise existing in MR images follows a Rician distribution. We follow Ridgway (2007) and add Rician noises to the sMRI image. Each time we add the noise of a different level ranging from 10 to 100 with an interval of 10, producing 10 noisy images. Then, we apply our method to these images and obtain the WIs of all the noisy images. In addition, we capture the changes of the ECT and HPV by using FreeSurfer. To make these indices comparable to each other, we calculate the change in percentage between each noisy image and the original one in terms of WI, ECT, and HPV, respectively, and then compare the relative changes under different levels of noises. We repeat the experiments for five times and take the average of the outcomes.

Figure 2.6 depicts the comparison. The Wasserstein index surpasses both HPV and ECT since its change stays closer to 0, while HPV suffers from a vibration around 1% and ECT has a larger change around 2%. The upward trend of the WI also suggests that our proposed index is sensitive to the global change of brain volumes when the noise becomes stronger enough to have a substantial impact on the original image. Compared with HPV under different levels of noises, the trajectory of the WD is smoother and more steady. From the figure, we find the WI is more robust than ECT and HPV when the noise level is low; under strong noises, ECT and HPV do not show major changes compared with under mild noises while WI starts to diverge from the ground truth.

To explore the practicality of our framework on sMRI images as well as its robustness over large brain image datasets, we apply it to images from the ADNI cohort, including those from 146 AD patients and 175 NC subjects, and compare the Wasserstein index (WI) with four frequently used single indices — average ECT, HPV, average cortical thickness (CT), and brain volume (BV) Cuingnet *et al.* (2011) — in terms of their classification accuracies. ECT, HPV, CT, and BV are measured by FreeSurfer Fischl (2012). All the indices are calculated on the left cerebral hemisphere.

We use the linear support vector machine (SVM) as the classifier and conduct 5-

Table 2.4: Classification Accuracy for Alzheimer's Distance on sMRI

Index	Accuracy	F1 Score
Wasserstein Index (WI)	80.1	83.6
Average Entorhinal Cortex Thickness (ECT)	79.1	82.0
Hippocampal Volume (HPV)	76.9	79.7
Average Corticle Thickness	73.2	76.5
Brain Volume	63.6	70.0

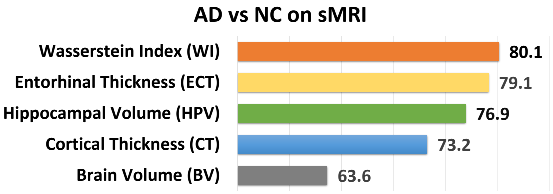


Figure 2.7: Results From a Linear SVM Classification Between Alzheimer's Disease and Normal Control. The Wasserstein Index (WI) Outperforms Traditional Indices—Entorhinal Cortex Thickness (ECT), Hippocampal Volume (HPV), Cortical Thickness (CT), and Brain Volume (BV).

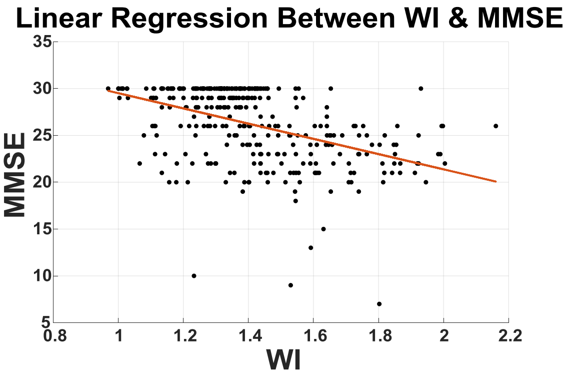


Figure 2.8: The Scatter Plot From the Linear Regression of Wasserstein Index (WI) and the Mini-Mental State Examination (MMSE). The Plot Suggests a Mild Negative Correlation Between WI and MMSE. The Root Mean Squared Error From the Linear Regression Model Is 3.55.

fold cross-validation. Figure 2.7 summarizes the classification accuracies of different indices. Among all the indices, WI achieves the highest accuracy of 80.1% while ECT and HPV provide 79.1% and 76.9%, respectively. Cortical thickness and brain volume yield 73.2% and 63.6%, respectively. We also report the F_1 scores in Table 2.4, which shows that our proposed index achieves relatively balanced results.

In addition, we test the correlation between the Wasserstein index and the clinical cognitive measure — the mini-mental state examination (MMSE) score. Figure 2.8 shows the result. The plot suggests a mild negative correlation between WI and MMSE, that is, subjects diagnosed as AD tend to have smaller MMSEs and larger WIs — which accords with other results. The model is significant at the 5% significance level with p-value $< 10^{-5}$. The root mean squared error is 3.55.

2.7 Summary

Optimal transportation has gained increasing popularity in recent years thanks to its robustness and scalability in many areas. In this paper, we have discussed its connection to K-means clustering. Built upon variational optimal transportation, we have proposed a clustering technique by solving iterative measure-preserving mapping and demonstrated its applications to domain adaptation, remeshing, and learning representations.

One limitation of our method at this point is computing a high-dimensional Voronoi diagram. It requires complicated geometry processing, which causes efficiency and memory issues. A workaround of this problem is to use gradient descent for variational optimal transportation because the only thing we need from the diagram is the intersections of adjacent convex hulls for computing the Hessian. The assignment of each empirical observation obtained from the diagram can be alternatively determined by nearest neighbor search algorithms. This is beyond the scope of this paper, but it could lead to more real-world applications.

The use of our method for remeshing could be extended to the general feature redistribution problem on a compact 2–manifold. Future work could also include adding regularization to the centroid updating process to expand its applicability to specific tasks in computer vision and machine learning. The extension of our formulation of Wasserstein means to barycenters is worth further study.

Chapter 3

REGULARIZING MONGE OPTIMAL TRANSPORTATION

In this chapter, we discuss regularizing Monge optimal transportation. Aligning distributional data often requires prior knowledge to be injected into the process to satisfy additional requirements. Regularizing OT is a common technique to achieve that. However, directly regularizing Monge OT is intractable due to two facts. One is that we use the variational method to solve Monge OT brings difficulties to regularizing it because we no longer have direct access to the OT map. The other fact is that the Monge OT map is binary and not differentiable. We introduce a method to work around these obstacles and regularize Monge OT maps.

3.1 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 *et al.* (2011), researchers have proposed various alignment techniques which benefit numerous fields including domain adaptation, e.g. Sun and Saenko (2016), and shape registration, e.g. Ma *et al.* (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 *et al.* (2013).

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

In this chapter, 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 *et al.* (2017b), focuses 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 *et al.* (2018a) 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 *et al.* (2017); Courty *et al.* (2017b); Mi *et al.* (2018a), we propose to regularize the mapping to cope with specific problems – domain adaptation and skeleton layout. The resulting mean, or centroid, can well represent the key property of the distribution while maintaining a small reconstruction error.

3.2 Related Work

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 *et al.* (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 *et al.* (2008) for averaging two distributions and generalized to multiple distributions in Agueh and Carlier (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 *et al.* (2017) defines it as the *Wasserstein means* problem. Before that, Cuturi and Doucet had discussed it in Cuturi and Doucet (2014) along with the connection of their algorithm to Lloyd's algorithm in that case. Mi *et al.* (2018a) proposes an OT-based clustering method which is very close to the Wasserstein means problem. Kolouri *et al.* (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 Doucet (2014) to compute

Wasserstein barycenters. Courty *et al.* (2017b) also leveraged class labels to regularize OT for domain adaptation. Ferradans *et al.* (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.

3.3 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 3. *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 $\lambda_i,\ i = 1, ..., N$, the Wasserstein mean ν_m of their average measure induces a lower*

Algorithm 4: 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 (3.3).
    $\pi^{(t+1)} \leftarrow$ Compute OT with fixed $y^{(t)}, \nu^{(t)}$.
    $y^{(t+1)} \leftarrow$ Update support according to (3.2).
    $t \leftarrow t + 1$.
until convergence.
return $\pi, y, \nu$.
```

bound of the average Wasserstein distance from the barycenter ν_b to them, provided that $|\Omega_{\nu_b}| \le |\Omega_{\nu_m}| \le k$ 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

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

Then, according to Wasserstein mean's definition,

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

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

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 *et al.* (2016) that $|\Omega_{\nu_b}| \le \sum_{i=1}^{N} |\Omega_{\mu_i}| - N + 1$, which also bounds $|\Omega_{\nu_m}|$ through $|\Omega_{\nu_m}| \le \sum_{i=1}^{N} |\Omega_{\mu_i}|$.

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

$$\begin{aligned} \min f(\pi, y, \nu) &\stackrel{\text{def}}{=} \min_{\pi, y_j, \nu_j} \sum_{j=1}^{k} \sum_{y_j = \pi(x)} \mu(x) \|y_j - x\|_2^2, \\ \text{s.t.}\ \ \nu_j &= \sum_{y_j = \pi(x)} \mu(x). \end{aligned} \tag{3.1}$$

Compared to OT, solving WM w.r.t. (3.1) introduces 2 additional parameters – measure ν and its support y. When y and ν are fixed, (3.1) becomes a classic optimal transportation problem and we adopt variational optimal transportation (VOT) Mi *et al.* (2018a) to solve it. Thus, (3.1) is minimizing the lower bound of the OT cost.

Then, it boils down to solving for y and ν. Certainly (3.1) 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

$$\tilde{y_j} = \frac{\int_{\Omega_\mu \cap S_j} x d\mu(x)}{\int_{\Omega_\mu \cap S_j} d\mu(x)}. \tag{3.2}$$

It is essentially to update the mean to the centroid of corresponding measures, adopted in for example Cuturi and Doucet (2014); Ye *et al.* (2017); Courty *et al.* (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 Doucet (2014), (3.1) 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

$$\begin{aligned} \tilde{\nu}(y_j) &= \int_{\Omega \cap S_j} d\mu(x), \\ \text{s.t. } S_j &= \{x \in M \mid \|x - y_j\|_2 \leq \|x - y_i\|_2, i \neq j\}. \end{aligned} \tag{3.3}$$

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. 4

As discussed in Cuturi and Doucet (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 Doucet (2014), where the update of the support is equivalent to re-centering it by our (3.2).

Complexity As we discuss in 2.4, we vectorize the computation with PyTorch because parameters in VOT 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. 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 $\mathcal{O}(KN)$.

Algorithm 5: 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 (3.2).
    repeat
        $y^{(t+1)} \leftarrow$ Update centroid by optimizing (3.6).
    until $y^{(t+1)}$ converges.
    $t \leftarrow t + 1$.
until $\pi$ and $y$ converge.
return $\pi, y$.
```

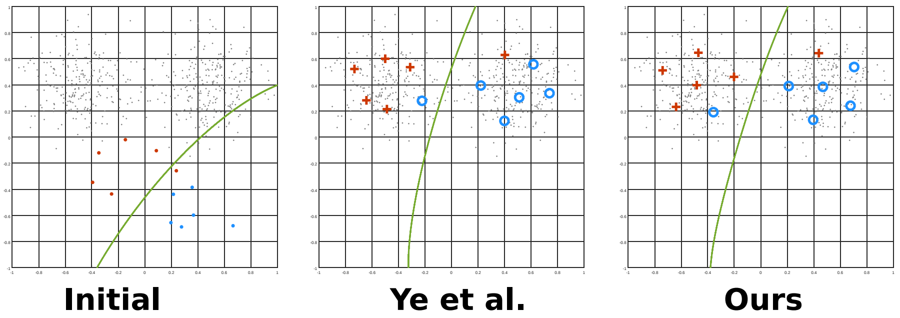


Figure 3.1: Matching Two Gaussian Mixtures With Ye *et al.* (2017) and Our Method. Updating Both Supports and Measures May Result in Centroids Not Evenly Distributed Into the Target Domain, Which Although May Not Affect the Classification Boundary in This Example.

Although both the supports $\boldsymbol{y}$ and the measures $\boldsymbol{\nu}$ are variable in the Wasserstein means problem, updating both of them complicates the optimization landscape and even problematic in some sense. In Figure 3.1, we show a comparison between our method and Ye *et al.* (2017), which updates both, on fitting a Gaussian mixture to the target domain. The two methods lead to similar decision boundaries, but our embedding is more evenly distributed into the target domain according to the density.

3.4 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:

$$\begin{gathered}\mathcal{L}(\pi, y) = \mathcal{L}_{ot}(\pi, y) + \lambda \mathcal{L}_{\text{reg}}(y), \\ \text{where } \mathcal{L}_{ot}(\pi, y) = \int_{\Omega} \|y - x\|_2^2 d\mu(x), \text{ where } y = \pi(x).\end{gathered} \tag{3.4}$$

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 (3.4) can also follow the block coordinate descent method. First, we fix the mean and compute the OT. Unlike in Alg. 4 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 (3.4).

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:

$$\begin{aligned}\int_{S_j} \|y_j - x\|_2^2 d\mu(x) &= (y_j^2 - 2y_j \int_{S_j} x d\mu(x) + C_1) \\ &= \|y_j - \int_{S_j} x d\mu(x)\|_2^2 + C_2 = \|y_j - \tilde{y}_j\|_2^2 + C_2.\end{aligned} \tag{3.5}$$

C_1, C_2 are some constants. $\tilde{y}_j$ is from (3.2) and S_j is the set in which x is mapped to y_j. It is defined by VOT as $S_j = \{x \ \in M | \langle y_j, x\rangle - h_j \geq \langle y_i, x\rangle - h_i\}, \forall i \neq j$. Thus, we re-write (3.4) as

$$\mathcal{L}(\pi, y) = \sum_j \|y_j - \tilde{y}_j\|_2^2 + \lambda \mathcal{L}_{\text{reg}}(y) \tag{3.6}$$

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. 5.

Citing the convergence proof from Grippo and Sciandrone (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 converges. In the 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.

3.4.1 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 *et al.* (2015), inspired by Weinberger *et al.* (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 (3.6) becomes

$$\mathcal{L}(y) = \sum_j \|y_j - \tilde{y}_j\|_2^2 + \lambda \mathcal{L}_{\text{triplet}}(y). \tag{3.7}$$

Fig. 3.2 shows an example of aligning Gaussian mixtures by (3.7). 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. 3.2 1st column shows the setups. The 2nd column shows the result

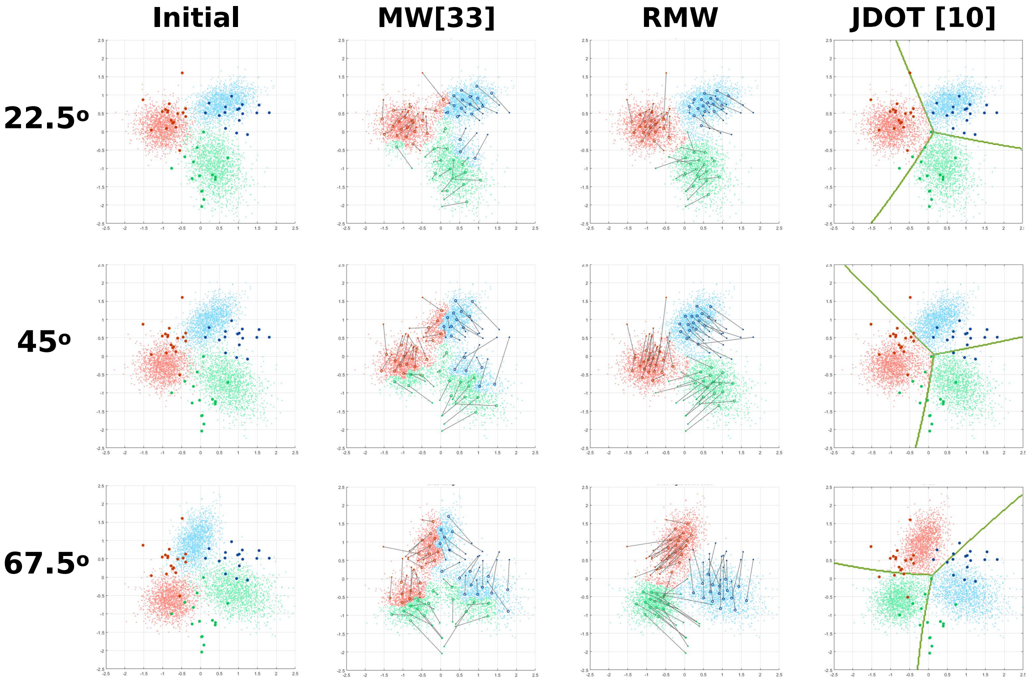
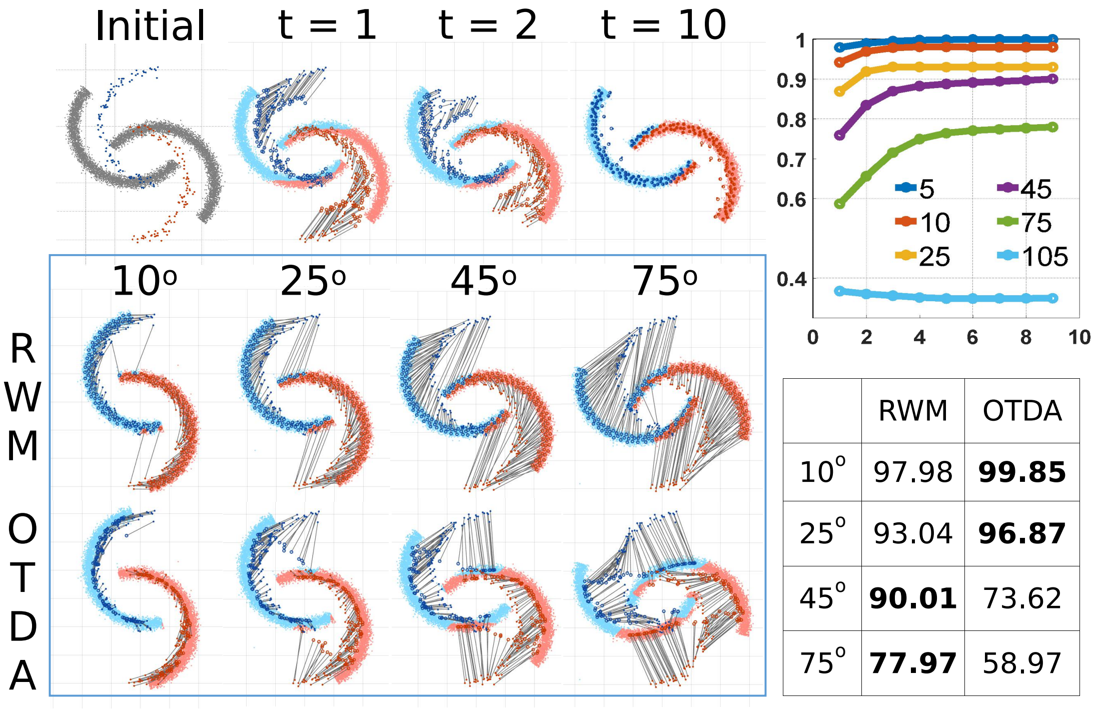


Figure 3.2: Regularizing the WM by the Intra-Class Triplets Can Adapt It to Domains That Suffer Unknown Rotations.

from computing the WM without regularization as in Mi *et al.* (2018a). 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 *et al.* (2018a). 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 *et al.* (2017a). It learns an RBF SVM classifier on the target samples.



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

Figure 3.3: 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.

3.4.2 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 *et al.* (2011); Courty *et al.* (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. 5. 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

$$\mathcal{L}(\pi, y) = \sum_j \|y_j - \tilde{y}_j\|_2^2 + \lambda \sum_j \|y_j - y_j^{\mathcal{T}}\|_2^2. \tag{3.8}$$

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

We demonstrate (3.8) with *two moons* in Fig. 3.3. 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$^{\text{nd}}$ row shows the result under different degrees of rotation. We weight in OTDA-GL's result Courty *et al.* (2017b) in the 3$^{\text{rd}}$ 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.

3.4.3 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 *et al.* (2015). Thus, the regularization on the length and curvature can express itself as follows:

$$\lambda \mathcal{L}_{\text{reg}} = \lambda_1 \sum_{1 \le i < k} g(\gamma'(y_i)) + \lambda_2 \sum_{1 < i < k} l(\gamma''(y_i)). \tag{3.9}$$

where $g(\cdot)$ and $l(\cdot)$ are some functions computed out of the length and curvature based on y, which are both convex making (3.9) 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 has branches, we can easily extend (3.9) 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:

$$\begin{aligned} \mathcal{L}(\pi, y) = &\sum_j \|y_j - \tilde{y}_j\|_2^2 \\ &+ \sum_{\gamma \in \Gamma} \big(\lambda_1 \sum_{1 \le i < k} g(\gamma'(y_i)) + \lambda_2 \sum_{1 < i < k} l(\gamma''(y_i)) \big). \end{aligned} \tag{3.10}$$

3.5 Applications

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

3.5.1 Domain Adaptation

We evaluate our method on the office-31 dataset Saenko *et al.* (2010). Office-31 consists of three subsets – Amazon, DSLR, 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. Fig. 3.4 shows some sample images.

We use the Decaf-fc6 and Decaf-fc7 features provided along with the dataset. Each sample is now encoded into a vector of 4,096 dimensions. The setup is similar to OTDA Courty *et al.* (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

Table 3.1: Classification Results (%) 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

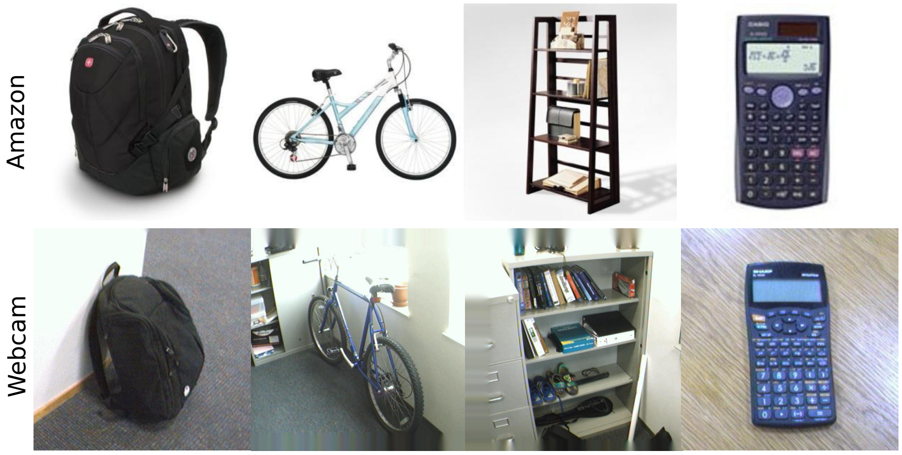


Figure 3.4: Sample Images From the Office-31 Dataset.

baselines. The experiments are repeated 10 times, and Tab. 3.1 summarizes the averaged results. RWM outperforms other methods by a large margin. We also show the resulting t-SNE embeddings in Fig. 3.5. 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'.

Adapting from W to A is challenging for RWM because W has too few samples for each class, and the target, in this case, should have even fewer samples. Adapting from, for example, 5 samples to 10 samples is not practical because 5 samples can hardly represent a domain. This is a limitation of Monge-based approaches. A workaround might be augmenting the target domain, creating more samples around the original samples, but it may not be practical either in high-dimensional spaces. Papadakis proposed, in a recent work Papadakis (2019), a new way of solving discrete optimal transportation by finding a few relays in between the source and the target

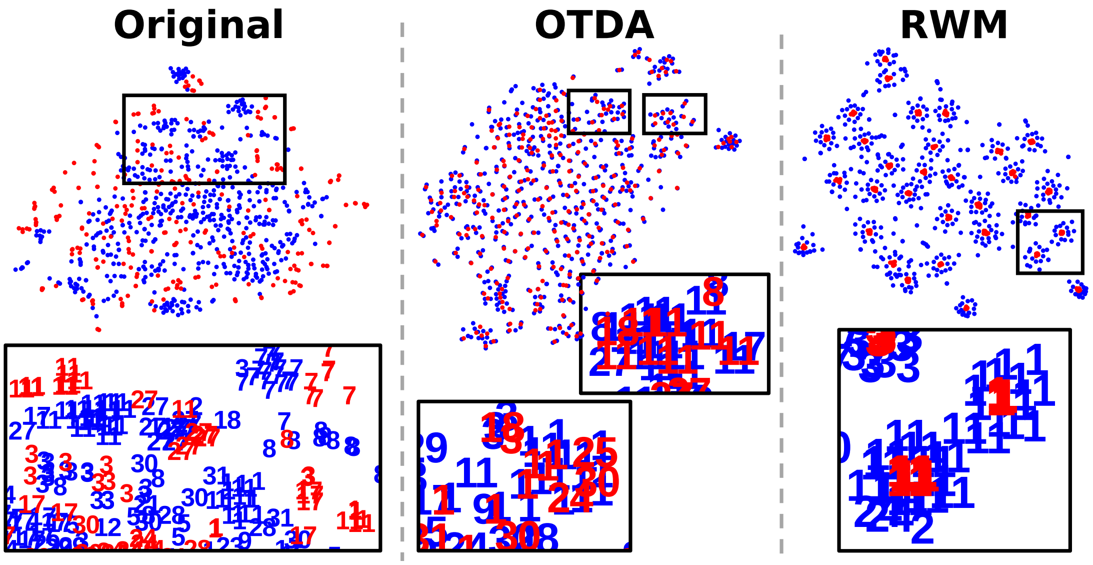


Figure 3.5: T-SNE Embeddings of the Office-31 Samples Before and After OTDA and RWM.

samples, which may bring insights to this problem of adapting too few samples. We leave it to future work.

3.5.2 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 3.6 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 (3.8) to recovering the transformation. With this example, we also test our algorithm under extreme conditions 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

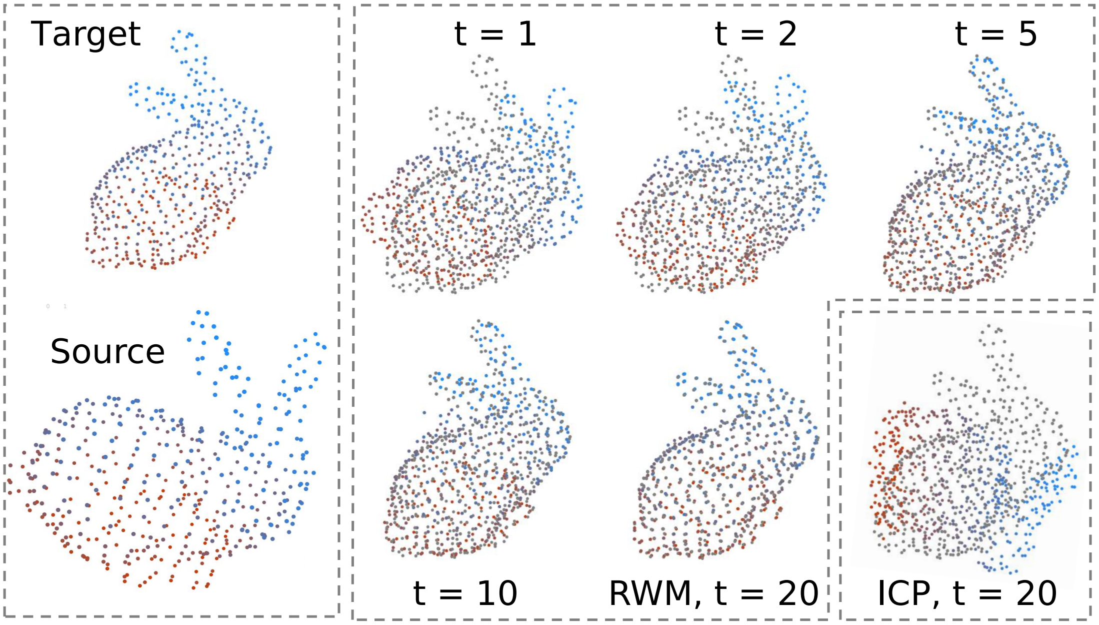


Figure 3.6: Alignment of Translationally and Rotationally Shifted Bunnies After RWM and ICP. T Indicates the Number of Iterations.

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 *et al.* (2016) for globally optimal alignment.

3.5.3 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. 3.7 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 (3.10). 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 (3.3). 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.

Top right of Fig. 3.7 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 the result from Stanford Armadillo. In the bottom right, we show the result from Solomon *et al.* (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 *et al.* (2015) was 1,200 seconds while ours took 15 seconds. CPU: Intel i5-7640x 4.0 GHz.

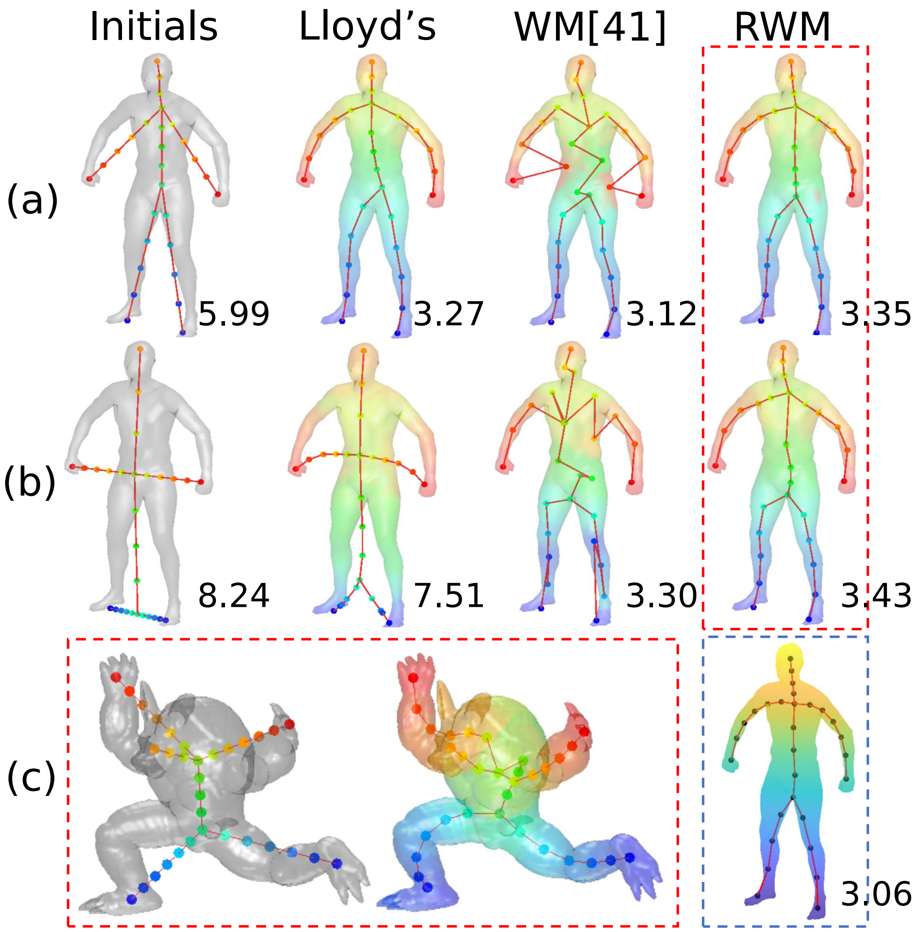


Figure 3.7: Skeleton Layout. RWM Embeds a Pre-Defined Graph Which Relates to the Shape of the Cloud. Numbers Indicating MSE Showing RWM Balances Between MSE and Topology.

3.6 Summary

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.

In this paper, we adopted VOT to obtain the OT map. In general, other OT solvers, e.g. Sinkhorn distances, could also work in our framework. 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.

Chapter 4

VARIATIONAL WASSERSTEIN BARYCENTERS FOR GEOMETRIC CLUSTERING

In this chapter, we advance our discussion to the Wasserstein barycenter problems. We generalize our methods for solving optimal transportation among multiple distributions. We expose the metric properties of the Wasserstein barycenter and its equivalence to pair-wise optimal transportation. We also discuss the immunity of our variational solution to unbalanced measures and its generalization to spherical domains. Finally, we showcase the use of our methods in solving a variety of geometric clustering problems.

4.1 Introduction

0Clustering distributional data according to their spatial similarities has been a core issue in machine learning. Numerous theories and algorithms for clustering problems have been developed to help understand the structure of the data and to discover homogeneous groups in their embedding spaces. Clustering algorithms also apply to unsupervised learning problems that pass information from known centroids to unknown empirical samples. Occasionally, researchers regard clustering as finding the optimal semi-discrete correspondence between distributional data or vice versa.

Optimal transportation (OT) techniques have gained increasing popularity in the past two decades for measuring the distance between distributional data as well as aligning them together. Rooted in the OT theories, several OT-based clustering algorithms have emerged in recent years as alternatives, thanks to their efficiency and robustness. In these works, the researchers discovered the connections between different clustering problems and the OT problem through the Wasserstein barycenter (WB) formulation which computes a "mean" of one or multiple distributions. However, most of them deliver the results as *soft assignments* that need to be further discretized.

In this paper, we propose to compute the Wasserstein barycenter based on Monge OT and explore its natural connections to different clustering problems that prefer *hard assignments*. We base our OT solver on variational principles and coin our method as variational Wasserstein barycenters. We study the metric properties of WBs and use them to explain and solve different clustering-related problems such as regularized K-means clustering, co-clustering, and vector quantization and compression. We also show its immunity to unbalanced measures and its extension to measures on spherical domains. We discuss our method from different angles through comparison with other barycenter methods. We show the advantages of Monge OT-based barycenters in solving geometric clustering problems. We are among the first few that compute Monge barycenters and discover its connections to clustering problems.

4.2 Related Work and Our Contributions

Computational clustering algorithms date back to Lloyd (1982); Forgy (1965) for solving K-means problems. From then, researchers have proposed different formulations and algorithms such as spectral clustering and density-based clustering. Mixture modeling, especially Gaussian mixture modeling, is also considered to be a robust solution to clustering problems. Hierarchical clustering and co-clustering also attracted much attention in the machine learning community. Xu and Wunsch (2005) surveys some classic clustering algorithms. The term "*geometric clustering*" appeared in the early literature, such as Murtagh (1983); Quigley and Eades (2000), referring to clustering samples into subspaces according to their location in the metric space, usually the Euclidean space. In Applegate *et al.* (2011), the authors discuss the connection between K-means and another famous problem – the OT distance, or the Wasserstein distance.

The transportation problem has attracted many mathematicians since its very birth. Thanks to efficient OT solvers, e.g., Cuturi (2013), OT has become a popular tool in machine learning with which we compare distributional data. Meanwhile, by regarding the OT distance as a metric, we can interpolate in the space of probability measure. McCann (1997) laid the foundation; Agueh and Carlier (2011) developed the problem into a general scenario and coined the term "*Wasserstein barycenters*". Cuturi and Doucet (2014); Ho *et al.* (2017); Mi *et al.* (2018a) relate WBs to K-means like clustering problems and Leclaire and Rabin (2019); Lee *et al.* (2019) explored the use of OT for hierarchical clustering. Claici *et al.* (2018) is among the latest work on scalable semi-discrete Wasserstein barycenters. Most of them follow Kantorovich's static OT; few of them follow Monge's, or Brenier's, dynamic version that regards OT as a gradient flow in the probability space.

Compared to previous work, our contribution is three-fold: 1) We derive the WB based on Monge's OT formulation and explore its connections to different clustering problems; 2) We prove the metric properties of our WB and propose it as a metric for evaluating multi-marginal clustering algorithms; 3) We explore the advantages and disadvantages of Monge WB through empirical comparison with other methods.

4.3 Variational Wasserstein Distances

We discuss computing discrete Wasserstein barycenters based on the variational solution to Monge OT. Then, we show that the WB induces a generalized metric among all the marginal distributions, and our method can produce an approximation to the exact Wasserstein distance between these marginals. We call our approximation the *Variational Wasserstein Distance*, or VWD. After that, we show that VWDs are immune to unbalanced measures, and, at last, we discuss how our method adapts to the spherical domain.

4.3.1 Wasserstein Barycenters through Variational OT

Solving the WB problem relies on alternatively solving N OT problems and updating the barycenter, ν. Eventually, ν minimizes the average WD between the marginals and the barycenter. A discrete distribution ν consists of support and measure $(\boldsymbol{y}, \boldsymbol{\nu}) = \{(y_k, \nu_k)\}_{k=1}^K$. Updating both of them, e.g., in Ye *et al.* (2017), however, is difficult and even troublesome in some cases (see Appendix). Some researchers refer to these two scenarios as free-support WBs and fixed-support WBs, e.g., Cuturi and Doucet (2014); Álvarez-Esteban *et al.* (2016). Free-support WBs usually imply that the measure is fixed. In this paper, we focus on free-support WBs, only updating the supports.

We first solve N VOT problems:

$$\min_{\{\boldsymbol{h_i}\}_{i=1}^N} I_5[\{\boldsymbol{h_i}\}] \overset{\text{def}}{=} \frac{1}{N} \sum_{i=1}^N \left(\int_{\mathbf{0}}^{\boldsymbol{h_i}} \sum_{k=1}^K \int_{\mathcal{R}_{i,k}} d\mu_i(x) dh_{i,k} - \sum_{k=1}^K \nu_k h_{i,k} \right)$$

Its derivative w.r.t. the VOT optimizer $h_{i,k}$ is

$$\nabla I_5[\boldsymbol{h}_i] = \left\{ \frac{\partial I_5}{\partial h_{i,k}} = \int_{\mathcal{R}_{i,k}} d\mu_i(x) - \nu_k \right\}_{k=1}^K, \tag{4.1}$$

which, in practice, can be replaced by its stochastic version,

$$\frac{\partial I_5}{\partial h_{i,k}} \approx \sum_{x \in \mathcal{R}_{i,k}} \mu_i(x) - \nu_k,$$

where x's are now Monte Carlo samples. Then, we can naturally adopt the gradient descent (GD) update:

$$\boldsymbol{h}_i^{(t+1)} = \boldsymbol{h}_i^{(t)} - \eta \nabla I_5[\boldsymbol{h}_i]. \tag{4.2}$$

For completeness, we give the second-order derivative later in the chapter. Its computation, however, involves integrating over the Voronoi facets and thus is intractable in general, especially for Monte Carlo samples. To accelerate the optimization, we adopt the momentum of the gradient in practice. We can optionally adopt modern first-order techniques to further accelerate the process. Another thing to notice is that since $\boldsymbol{h}_i$ is directly added to the Euclidean distance to determine $\boldsymbol{\mathcal{R}}_i$; thus we should scale the step size to compensate the difference in the Euclidean distance brought by dimensionality. If we use d to denote the dimensionality, i.e. $\mathbb{R}^d$, then we scale the step size as $\eta \leftarrow \eta \times d$.

To solve for ν^*, we rewrite the objective of the WB (2.3) as

$$\begin{aligned} \min_{\nu \in \mathcal{P}(\mathcal{Y})} I_6[\nu] &\overset{\text{def}}{=} \frac{1}{N} \sum_{i=1}^{N} \int_{\mathcal{X}_i} \|x - T_i^*(x)\|_2^2 d\mu_i(x) \\ &= \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \int_{\mathcal{R}_{i,k}} \|x - y_k\|_2^2 d\mu_i(x), \end{aligned} \tag{4.3}$$

s.t. $y_k = T_i^*(x),\ \forall x \in \mathcal{R}_{i,k}$. The critical point of this quadratic energy w.r.t. each y_k has a closed form:

$$y_k^* = \frac{\sum_{i=1}^N \int_{\mathcal{R}_{i,k}} x d\mu_i(x)}{N \sum_{i=1}^N \int_{\mathcal{R}_{i,k}} d\mu_i(x)} \approx \frac{\sum_{i=1}^N \sum_{x \in \mathcal{R}_{i,k}} x\mu_i(x)}{N \sum_{i=1}^N \sum_{x \in \mathcal{R}_{i,k}} \mu_i(x)}, \tag{4.4}$$

which is the center of mass of its correspondence across all measures. The latter expression is the "stochastic" version.

The last step is to derive the update rule for the measure $\boldsymbol{\nu}$. (4.3) is not differentiable w.r.t. $\boldsymbol{\nu}$. Still, we follow Cuturi and Doucet (2014); Mi *et al.* (2018b) and

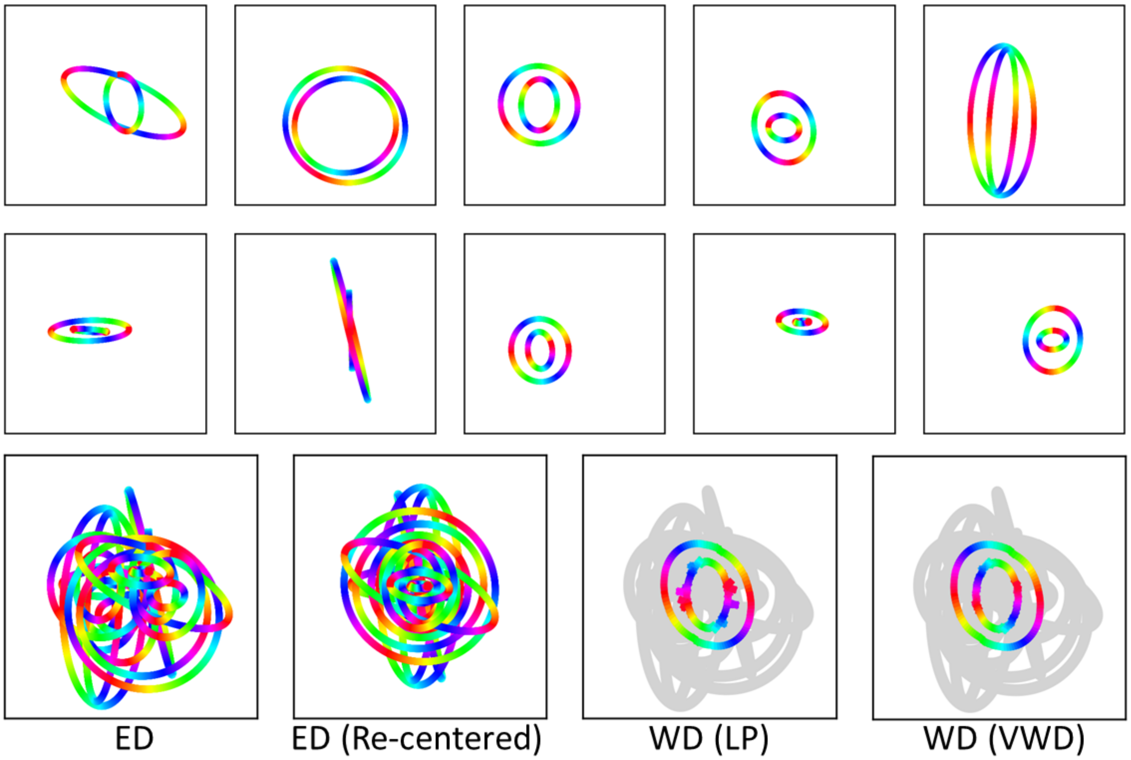


Figure 4.1: Ten Random Nested Ellipses Averaged According to the Euclidean Distance (ED) and the Wasserstein Distance (WD). For a Better Visual, We Use Euclidean Sums Instead. Compared With the Linear Programming (LP) Solver, Using Our Method (VWD) Leads to a Smoother Barycenter. Both Solvers Preserve the Topology (Rainbow Colors) of the Ellipses.

directly give the critical point and include the derivation in Appendix.

$$\nu_k^* = \frac{1}{N}\sum_{i=1}^{N}\int_{\mathcal{R}_{i,k}^*} d\mu_i(x) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{x\in\mathcal{R}_{i,k}^*}\mu_i(x), \tag{4.5}$$

where $\mathcal{R}_{i,k}^* = \{x \in \mathcal{X}_i \mid \|x - y_k\|_2^2 < \|x - y_\ell\|_2^2 \ \forall \ell \neq k\}$. ν_k^* coincides with the result of Lloyd's K-means algorithm in which the measure on each centroid accumulates all its assigned empirical measures.

Algorithm Now that we have derived the rules for updating T and ν, we summarize our algorithm for computing the discrete Wasserstein barycenter of a collection of measures $\{\mu_i\}_i$ in Appendix. We name the resulting barycenter the *variational Wasserstein barycenter* or VWB. As for the initial guess of the VWB, if not predefined, we provide three options: 1) run Lloyd's algorithm on all the marginals as a whole and adopt the resulting K centroids; 2) uniformly sample the space $\mathcal{Y}$ with K atoms; and 3) randomly choose one of the marginal. The choice of the measure on the centroids depends on the specific application. A ubiquitous choice is uniform Dirac measures, i.e. $\nu = \{\frac{1}{K}\delta_y(y_k)\}$. In Fig 4.1, we compare the barycenter w.r.t. the Euclidean distance and the Wasserstein distance. It suggests that by regarding the WD as the metric, we can find a mean shape on the same manifold (if there exists one). We also show the barycenter computed by using the classic linear programming (LP) method as implemented in Flamary and Courty (2017). The result is roughly the same as ours, but some centroids in there are not on the inner circle. The reason is that LP is solving the Kantorovich OT that allows partial mapping. Consequently, some empirical samples on the outer circles are partially mapped to the centroids that are supposed to be on the inner circle. As a result, when updating the centroids according to the center of the mass, some centroids on the inner circle are "dragged outward".

Time Our method does converge since we follow coordinate descent and every step is convex Grippo and Sciandrone (2000), given the assumption we made in 2.3 that $\mathcal{X}, \mathcal{Y} \subset \mathbb{R}^n$, $c(x,y)^2 = \|x-y\|_2^2$.

Memory There are in total $\mathcal{O}(K \cdot N)$ variables for computing N Monge maps $\{T_i\}_{i=1}^N$, and $\mathcal{O}(K)$ variables as the support $\boldsymbol{y}$ and $\mathcal{O}(K)$ variables as the measure $\boldsymbol{\nu}$. Taking into account the dimensionality, the complexity would be $\mathcal{O}(K \cdot n)$, but we omit n for simplicity. There are two ways to update the height vectors. The first is to compute each OT problem separately; the second is to concatenate all the

Algorithm 6: Variational Wasserstein Barycenters

```
Input: $\{\mu_i\}_{i=1}^{N}$, $K$
Initialize $\nu \in \mathcal{P}(\mathcal{Y})$.
repeat
  Compute $T_i$ between $\nu$ and each $\mu_i$ by solving (4.1).
  Update partition or assignment (if discrete), $\boldsymbol{\mathcal{R}}$, according to $\boldsymbol{T}$.
  if Free support then
    Update $y_\ell \ \forall \ \ell$ according to (4.4)
  end if
  if Free measure then
    Update $\nu_\ell \ \forall \ \ell$ according to (4.5)
  end if
until $\nu$ converges
```

minimizers and parallelize the computation on feasible hardware. The trade-off is between the time and the memory – the memory consumption for the second option is $\mathcal{O}(K \cdot N \cdot M)$.

Another source of memory consumption is the pre-computed pair-wise distance matrices between the centroids and all the marginals. Computing pair-wise distances beforehand is optional but preferable because it dramatically reduces the time to compute VWB. The majority of the computation is to find the nearest centroid for every empirical sample in every marginal distribution so that we can compute the total mass for every Voronoi cell and then update the Voronoi diagrams. Pre-computing distance matrices allows us to vectorize the computation and simultaneously determine the assignments for all the empirical samples through matrix and vector operations. Moreover, because the power distance that induces the power diagram is a

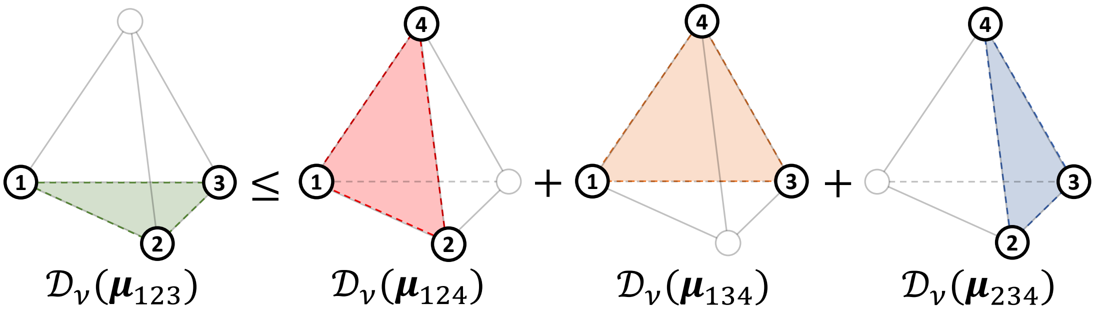

1
2
3
≤
4
1
2
+
4
1
3
+
4
3
2
$\mathcal{D}_\nu(\boldsymbol{\mu}_{123})$
$\mathcal{D}_\nu(\boldsymbol{\mu}_{124})$
$\mathcal{D}_\nu(\boldsymbol{\mu}_{134})$
$\mathcal{D}_\nu(\boldsymbol{\mu}_{234})$

Figure 4.2: The Triangle Inequality for $\mathcal{D}_\nu(\boldsymbol{\mu}_{1:3})$ Which Is a 3-Metric.

straight summation of the squared quadratic Euclidean distance and the 'height', precomputing a matrix of the squared quadratic Euclidean distances allow us to use it as a placeholder for incorporating the newly updated height vector and then directly index empirical samples with the power distance.

Code We implemented VWB in Python with PyTorch Paszke *et al.* (2019). The code to reproduce the figures in this paper is at https://github.com/icemiliang/pyvot.

4.3.2 The Metric Properties of Wasserstein Barycenters

Despite the extensive studies on the metric properties of the WD over the past century, the metric properties of Wasserstein barycenters have yet been fully explored. Papadakis (2019); Auricchio *et al.* (2018) are Some pioneer work. However, they all focus on the barycenter of two measures ($N = 2$). We show in the following that the WB in general ($N \geq 2$) induces a *generalized metric* (*n-metric*) among all the marginals.

First, let us denote the total Wasserstein distance between the barycenter and all the marginal as follows:

$$\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N}) \overset{\text{def}}{=} \inf_{\nu \in \mathcal{P}(\mathcal{Y})} \frac{1}{N} \sum_{i=1}^{N} \mathcal{W}_2^2(\mu_i, \nu), \tag{4.6}$$

Then, we raise the following two propositions.

Proposition 4. *$\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N})$ is a generalized metric among $\boldsymbol{\mu}_{1:N}$, $N \geq 2$. Specifically, $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N})$ satisfies the following properties.*

1) Non-negativity: $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N}) \geq 0$.

2) Symmetry: $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{\sigma_1(1:N)}) = \mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{\sigma_2(1:N)})$, where $\sigma_1(1:N)$ and $\sigma_2(1:N)$ are different permutations of the set $1:N$.

3) Identity: $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N}) = 0 \iff \mu_i = \mu_j, \forall i \neq j$.

4) Triangle inequality: $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N}) \leq \sum_{i=1}^{N} \mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N+1\backslash i})$.

Proposition 5. *The bound of the triangle inequality in Proposition 4 can be tightened by a linear factor. Specifically, we have $(N/2)\ \mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N}) \leq \sum_{i=1}^{N} \mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N+1\backslash i})$.*

If we regress to arbitrary weights, i.e.

$$\hat{\mathcal{D}}_{\nu^*}(\boldsymbol{\mu}_{1:N}) \overset{\text{def}}{=} \inf_{\nu \in \mathcal{P}(\mathcal{Y})} \sum_{i=1}^{N} \lambda_i \mathcal{W}_2^2(\mu_i, \nu), \tag{4.7}$$

for $\lambda_i \in [0,1]$ and $\sum_i \lambda_i = 1$, then, this total distance $\hat{\mathcal{D}}_{\nu^*}(\boldsymbol{\mu}_{1:N})$ still satisfies the metric properties in Proposition 4 and 5. For the triangle inequality to hold, we need to assume that $\lambda_{N+1} = \lambda_i$ in each $\hat{\mathcal{D}}_{\nu}(\boldsymbol{\mu}_{1:N+1\backslash i})$ for all i.

Proposition 6. *$\hat{\mathcal{D}}_{\nu^*}(\boldsymbol{\mu}_{1:N})$ is a generalized metric among $\boldsymbol{\mu}_{1:N}$, $N \geq 2$ and satisfies all metric properties as $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N})$ does in Proposition 4 and 5.*

We prove the above three propositions in Appendices. Figure 4.2 illustrates the triangle inequality for $\mathcal{D}_{\nu}(\boldsymbol{\mu}_{1:3})$ which is a 3-metric among $\boldsymbol{\mu}_{1:3}$.

The VWB $\nu^* = \sum_k \nu_k \delta_y(y_k) \in \mathcal{P}(\mathcal{Y})$, as a special case of the WB that consists of a set of Dirac measures, certainly inherits the metric properties since there is not a restriction on the continuity of the support y. We write this point in the following corollary.

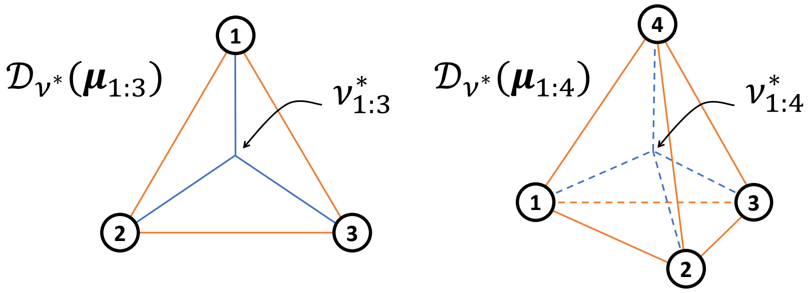

$\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:3})$
1
$\nu^*_{1:3}$
2
3
4
$\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:4})$
$\nu^*_{1:4}$
1
3
2

Figure 4.3: Using VWDs (Blue) and the Total Pairwise WDs (Orange) for Measuring the Compactness of Multiple Probability Distributions. Left and Right Illustrate the Case of 3 and 4 Marginals, Respectively.

Corollary 2. *Suppose, $\nu^* = \sum_k \nu_k \delta_y(y_k)$. Then, Proposition 4 and 5 still hold for $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N})$ and Proposition 6 for $\hat{\mathcal{D}}_{\nu^*}(\boldsymbol{\mu}_{1:N})$. In particular, the equal signs in 1) non-negativity and 4) inequality hold only if all μ_i's and ν^* have the same number of supports $|\mu_i| = |\nu^*| = K,\ \forall i \in \{1, ..., N\}$.*

The condition on the cardinality ensures the feasibility for all the discrete distributions to be equal to each other.

4.3.3 A Distance among Multiple Distributions

A distance among multiple distributions can measure their compactness or closeness together. It is particularly useful when the distributions are embedded in topological spaces, e.g., the Wasserstein space, where we do not have exact locations but only pair-wise connections. Computing the total pair-wise distance is a straightforward approach. With the WD as an embodiment of the distance, the summation becomes

$$\frac{2}{N(N-1)} \sum_{i,j} \mathcal{W}_2^2(\mu_i, \mu_j),\ \forall\ 1 \leq i < j \leq N. \tag{4.8}$$

Its computational complexity, however, increases quadratically to the number of marginals, i.e., $\mathcal{O}(N^2)$. This brings hardness given the already high expense of computing WDs. Mérigot *et al.* (2019) discusses linearizing the Wasserstein space by projecting samples to the tangent space so that, instead of computing pair-wise WDs, we can now compute the WDs from the marginals to their respective projections on the tangent spaces and then use Euclidean distances afterward. It brings down the number of WDs from $N(N-1)$ to N at the expense of the linearization error. Courty *et al.* Courty *et al.* (2018) propose to learn a vector space that approximates the Wasserstein space in terms of the pair-wise WDs to avoid computing WDs whatsoever. These approaches are all based on the idea of finding the approximation.

We instead propose to use VWD, $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N})$, as an alternative that reduces the complexity to linear to N. To differentiate these two alternatives, we refer to our formulation, VWD, as a barycentric distance and refer to (4.8) as a pair-wise distance. Fig 4.3 depicts the relationship between the original summation and our VWD. Similar to K-means, measuring the barycentric distance is equivalent to measuring the pair-wise distance. This conclusion comes immediately after replacing the squared quadratic Euclidean distance in the K-means formulation to squared 2-WD. For completeness, we state this point below and include the derivation in Appendix.

Remark 1. *VWD* (4.6) *is equivalent to* (4.8)*.*

The two approaches also connect in such a way as stated in the following proposition.

Proposition 7. *Suppose,* $\nu^* = \sum_k \nu_k \delta_y(y_k)$*. Then,* $\hat{\mathcal{D}}_{\nu^*}(\boldsymbol{\mu}_{1:N})$ *is lower-bounded by* (4.8)*. The bound is achieved if and only if all the marginals equal to each other. For a discrete barycenter, that also implies that* $|\mu_i| = |\nu^*| = K, \ \forall i \in \{1, ..., N\}$*.*

Compared to the pair-wise formulation, ours also allow us to introduce nonuniform weights to the marginals ("nodes"), instead of to the joint distributions ("edges"), so as to promote or demote certain marginals when computing the compactness of the marginals as a whole.

When $N = 2$, the problem relates to a transshipment problem which is finding the least expensive map between two probability distributions that passes through a set of relays. Apart from the applications of optimal transshipment to trading and networking, it has been proposed as an approximation to the true WD between the two marginals Papadakis (2019). In this case, we are seeking an optimal relay distribution that induces the minimum total transportation cost across the two marginals. Plugging $N = 2$ into Proposition 7, we obtain the connection of the VWD and the true WD for the transshipment problem as $\mathcal{D}_{\nu^*}(\mu_1, \mu_2) \leq \frac{1}{4}\mathcal{W}_2^2(\mu_1, \mu_2)$. The equal sign holds when $|\mu_1| = |\mu_2| = |\nu^*|$. In this case, each supporting atom of the ν^* lies in the middle of its two corresponding atoms in the two marginals. Papadakis (2019) studies this case in detail from the perspective of Kantorovich OT. Fig 4.4 illustrates the transshipment map. From the figure, we can see that our method can produce sparse, binary correspondence which is preferable for clustering tasks.

4.3.4 On Unbalanced Measures

Distributional data are not necessarily probabilities. When their measures do not integrate to the same total, we are solving *unbalanced OT*. Some researchers may refer to unbalanced OT as *generalized OT* Piccoli and Rossi (2014), but in this paper, we use the term generalized OT for OT among multiple distributions and the term unbalanced OT for OT among two or more distributions that do not integrate to the same total. Benamou (2003) first explored the problem. Researchers since then have offered various formulations and perspectives to approach it, e.g., in Liero *et al.* (2018) by adding f-divergences as regularizers instead of constraints on the marginalization.

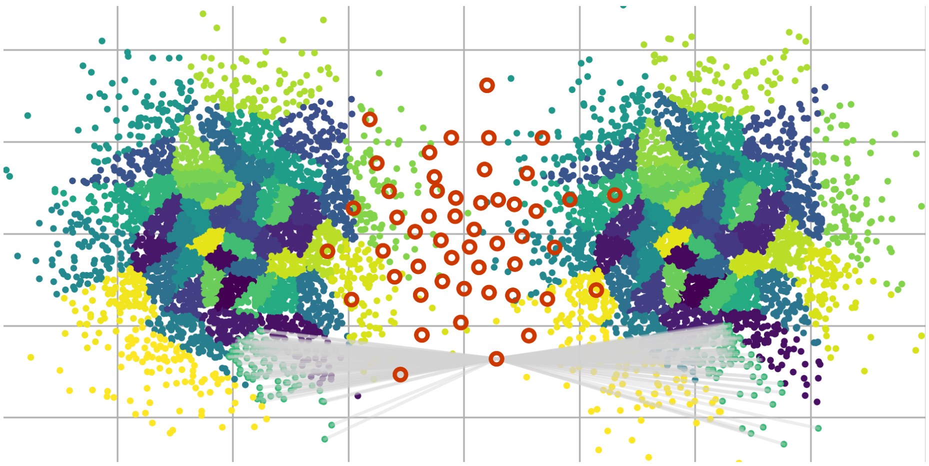


Figure 4.4: Transshipment: Transporting Measures Through a Set of Discrete Relays. Colors on the Measures Specify Correspondences.

Most of the existing work is from the perspective of Kantorovich OT. In that case, the constraints on the marginalization cannot be satisfied anymore, but be relaxed to a regularizer. Here, we show that VOT, which solves Monge OT, is robust to unbalanced measures, and subsequently, the VWD is too.

Without loss of generality, let us assume $\int_{\mathcal{X}} d\mu(x) = w$ and $\sum_{k=1}^{K} \nu_k = 1$. We denote the total mass of μ in each power Voronoi cell by $w_k = \int_{\mathcal{R}_k} d\mu(x)$; thus $w = \sum_{k=1}^{K} w_k$. We split the problem into two cases: a special case where $\boldsymbol{\mu}$ is uniform, i.e. ν $\nu_k = \frac{1}{K}$, and a more general one where $\nu_k \in (0,1)$, $\sum_{k=1}^{K} \nu_k = 1$. We only consider non-negative ν_k; otherwise a zero measure effectively changes K which complicate the problem. We copy the VOT formulation (2.5) below for ease of reading.

$$I_2[\boldsymbol{h}] \overset{\text{def}}{=} \int_{\mathbf{0}}^{\boldsymbol{h}} \sum_{k=1}^{K} \int_{\mathcal{R}_k} d\mu(x) dh_k - \sum_{k=1}^{K} \nu_k h_k, \tag{2}$$

Case 1: $\nu_k = \frac{1}{K}$, $\forall\ k \in \{1, ..., K\}$.

Our first intuition is that $w_k = \frac{1}{K} w$, $\forall\ k$ is at the optimal point because that

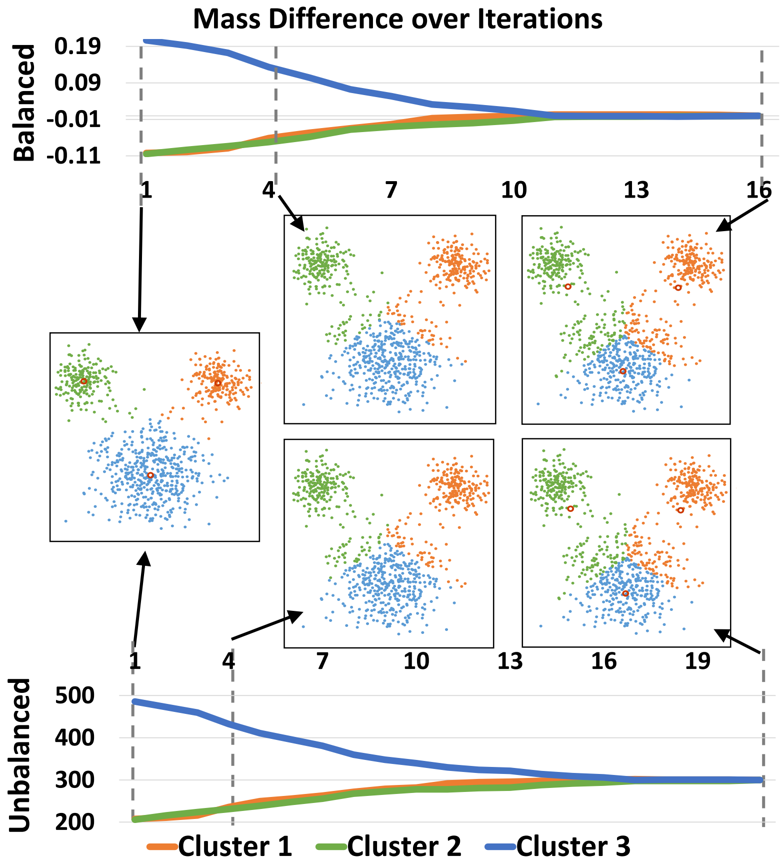


Figure 4.5: Mass Difference Over Iterations for VOT on Balanced and Unbalanced Measures. They Follow the Same Trend and Converge at Almost the Same Rate. The Resulting Clusters Are Exactly the Same .

means equal mass for all the cells, which is what $\nu_k = \frac{1}{K}$ seeks. The question now is whether $w_k = \frac{1}{K}w$ indeed minimizes (2.5).

When $w_k = \frac{1}{K}w$, the gradient $\nabla I_2[\boldsymbol{h}] = \{w_k - \frac{1}{K}\}_k$ becomes constant $\frac{1}{K}(w-1)\cdot\mathbf{1}$. Thus $\boldsymbol{h}$ is being translated, or lifted, at the same speed in all directions, which does not change the graph of the piece-wise linear function $\theta_{\boldsymbol{h}}(x) = \max_k\{xy_k + h_k\}$ Alexandrov (2005); Gu *et al.* (2013). Given the equivalence between the graph of $\theta_{\boldsymbol{h}}(x)$ and the graph of the power Voronoi diagram as specified in (2.16), the variational energy $I_2[\boldsymbol{h}]$

saturates to the point where $w_k = \frac{1}{K}w$. For any other graph, or partition, such that $\exists\ \mathcal{R}'_k$, where $w'_k \neq \frac{1}{K}w$, we have

$$\sum_{k=1}^{K} \int_{\mathcal{R}_k} \left(\|x - y_k\|_2^2 + h_k\right) d\mu(x)$$
$$\leq \sum_{k=1}^{K} \int_{\mathcal{R}'_k} \left(\|x - y_k\|_2^2 + h_k\right) d\mu(x).$$

When the optimal point is not reached yet, there exist at least two adjacent cells k, ℓ such that $w_k > w_\ell$, the corresponding derivatives being $\frac{\partial I_2}{\partial h_k} = w_k - \frac{1}{K} > w_\ell - \frac{1}{K} = \frac{\partial I_2}{\partial h_\ell}$. The boundary which is induced by $xy_k + h_k$ will be shifting to k, promoting a relative smaller w_k compared to w_ℓ.

Case 2: $\nu_k \in (0, 1),\ \sum_{k=1}^{K} \nu_k = 1$.

Similarly, $w_k = \nu_k w$ (replacing $\frac{1}{K}$ in Case 1 with ν_k) is at the optimal point of $I_2[\boldsymbol{h}]$.

Therefore, $w_k = \frac{1}{K}w,\ \forall\ k \in \{1, ..., N\}$ indeed minimizes the total transportation cost (2.1), whether the two distributions have the same total measure.

We illustrate the convergence in Fig 4.5. Suppose we have a Gaussian mixture of three components having 500, 200, and 200 samples, respectively. We initialize three centroids by K-means++ and then compute VOT from the Gaussian mixture to the 3 centroids. The top half shows the convergence of the first trial in which we normalize the data so that both the Gaussian mixture and the 3 centroids have the total measure of 1; the bottom half shows the convergence of VOT on the data without normalization, i.e., $w = 900, \nu_k = \frac{1}{3},\ k = 1, 2, 3$. Note that the gradient of the VWB, (4.1), correlates to the absolute measure values. Thus, we should scale the step size, η in (4.2), for each VOT according to the difference of the measure, i.e. $\eta_i/(w - 1)$, assuming the total for ν is 1. Fig 4.5 shows that under the same (scaled) GD step size, VOT in two cases follows the same trend and converge to the same map.

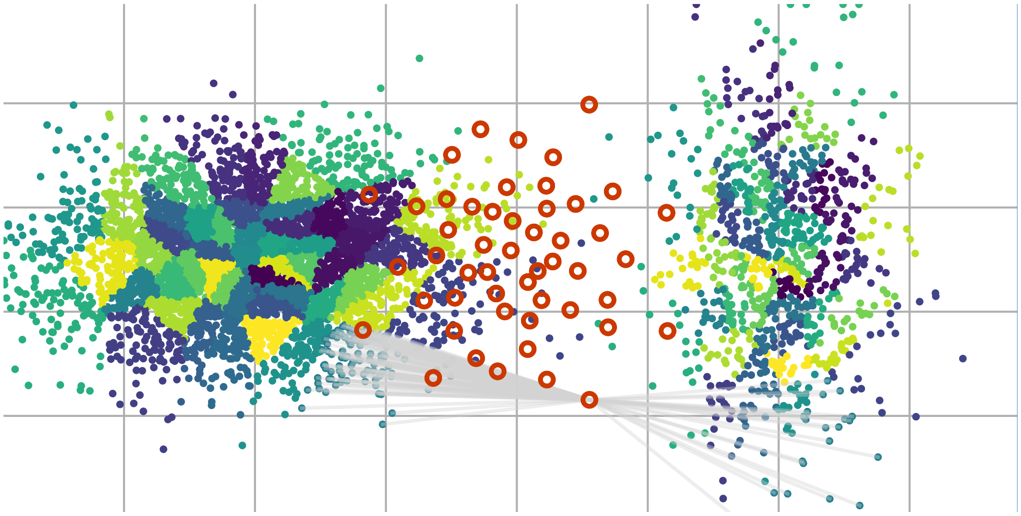


Figure 4.6: Interpolating Two Gaussian's of Different Number of Samples by Computing the VWB Results in a Mean Isotropic Gaussian.

We compute the VWB of two unbalanced Gaussian distributions and show the result in Fig 4.6. The Left Gaussian has 5k samples, and the right one has 1k samples. Each has a diagonal covariance matrix, and the value is swapped. Each of the samples, including the 50 samples of the barycenter, has the same weight. As we expect, the resulting barycenter is an isotropic Gaussian distribution which is exactly what it would be for two balanced Gaussian distributions. Different numbers of samples in two corresponding clusters are connected through a single relay, as shown in the figure.

In terms of VWD under unbalanced measures,

4.3.5 On the Spherical Domain

Optimal transportation on geometric domains other than the Euclidean domain extends its applicability Solomon *et al.* (2015); Staib *et al.* (2017); Cui *et al.* (2019).

Cui *et al.* (2019) relates the *spherical power Voronoi diagram* Sugihara (2002) to OT on unit spheres and applies it to computing an area-preserving map on spheres. Inspired by that, we study our VWD on spherical domains and its metric properties.

Let us define a new ground metric on a unit sphere, $\mathbb{S}^2 \times \mathbb{S}^2 \to \mathbb{R}^{\geq 0}$, as $c(x, y_k) = -\ln\langle x, y_k\rangle$ and the OT distance:

$$\mathcal{W}_1' = \inf_{T \in \Pi_T(\mu,\nu)} I_8[\pi] \overset{\text{def}}{=} -\int_{\mathbb{S}^2} \ln\langle x, T(x)\rangle d\mu(x) \tag{4.9}$$

s.t. $\int_{\mathbb{S}^2}(\psi \circ T)d\mu(x) = \int_{\mathbb{S}^2} \psi d\nu(y)$ for all non-negative ψ. Following Cui *et al.* (2019), we define the *power distance* on a sphere as $c'(x, y_k) = -\ln\langle x, y_k\rangle / \cos r_k$ and thus the power Voronoi diagram in the spherical domain $\mathcal{R}_k \overset{\text{def}}{=} \{x \in \mathbb{S}^2 \; | c'(x, y_k) \leq c'(x, y_\ell), \forall \ell \neq k\}$. r_k is the weight of each power cell, it relates to the VOT minimizers by $\cos r = e^h$. Then, the derivation in 2.4 gives us the Monge map.

$-\ln\langle x, y_k\rangle$ does not satisfy triangle inequality but the other three metric properties. Thus, $\mathcal{W}_1'$ inherits those properties. We notice that the proof for Proposition 4 does not leverage the triangle inequality of the WD. Therefore, if we define the VWD on spherical domains as the following, it is also a generalized metric.

$$\mathcal{D}_{\nu^*}'(\boldsymbol{\mu}_{1:N}) \overset{\text{def}}{=} \inf_{\nu \in \mathcal{P}(\mathcal{Y})} \frac{1}{N} \sum_{i=1}^{N} \mathcal{W}_1'(\mu_i, \nu) \tag{4.10}$$

Although $\mathcal{W}_1'$ is not a true metric, we can still find a "mean" of multiple marginals by minimizing the VWD as in 4.3.1. Fig 4.7 shows an example where the VWB simultaneously partitions two Gaussian distributions on the sphere. For simplicity, we draw connections with straight lines.

4.4 Geometric Clustering through VWDs

In this section, we further connect VWBs to several clustering problems. We consider a fixed number of clusters, K, the quadratic Euclidean distance as the

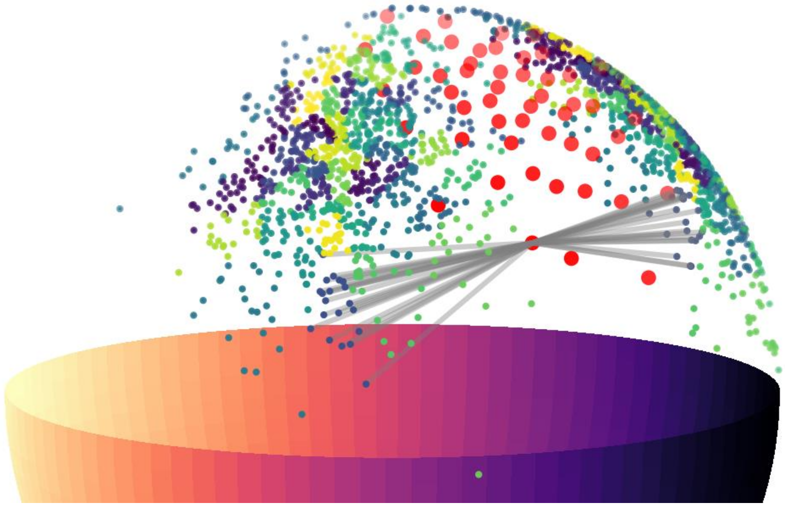


Figure 4.7: Interpolating Two Gaussian Distributions on a Sphere by Minimizing the VWD. At the Same Time, We Build Sparse Connections Between the Two Distributions via a Few Discrete Relays.

ground metric, and mainly the spatial relation between samples. We refer to this scenario as *geometric clustering*. We discretize the measures: $\nu = \sum_{k=1}^{K} \nu_k \delta[y_k], \mu_i = \sum_{j=1}^{n_i} \mu(x_j)\delta[x_j], \forall\ i \in \{1, ..., N\}$ for further discussion and assume that $n_i \gg K,\ \forall i$.

4.4.1 Regularized K-Means Clustering

In light of the discovery of VWDs for unbalanced measures in 4.3.4, we now introduce a relaxed version of the constrained K-means clustering problem. We call it the *regularized K-means* problem.

The classic K-means problem has the following objective:

$$\min_{\boldsymbol{\mathcal{R}}} \sum_{k=1}^{K} \sum_{x \in \mathcal{R}_k} \|x - y_k\|_2^2, \quad y_k = \frac{1}{|\mathcal{R}_k|} \sum_{x \in \mathcal{R}_k} x, \tag{4.11}$$

where $|\mathcal{R}_k|$ is the number of samples supported in $\mathcal{R}_k$. By adding the marginal constraint $\nu_k = \sum_{x \in \mathcal{R}_k} \mu(x)$ with pre-defined, fixed measures $\{\nu_k\}_{k=1}^{K}$, we turn (4.11) into the *constrained K-means* problem Bradley *et al.* (2000); Cuturi and Doucet (2014), or the *Wasserstein Means* problem Ho *et al.* (2017). We propose that when the total measures do not equal, we relax such constraints and turn them into regularizers for the clustering energy. Then, we define the objective of the regularized K-means problem as:

$$\min_{\boldsymbol{\mathcal{R}}} \sum_{k=1}^{K} \sum_{x \in \mathcal{R}_k} \|x - y_k\|_2^2 + \lambda \sum_{k=1}^{K} \left(\nu_k - w_k\right)^2 , \tag{4.12}$$

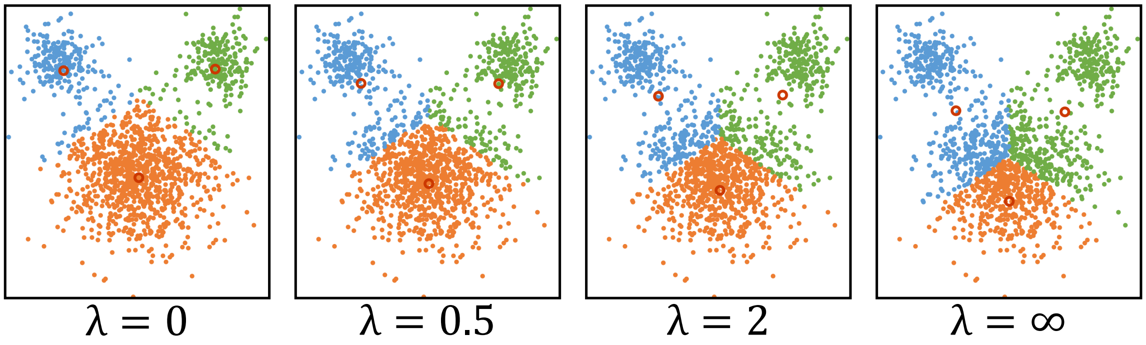
where $w_k = \sum_{x \in \mathcal{R}_k} \mu(x)$. If $\lambda = 0$, (4.12) regresses to the classic K-means problem; if $\lambda \to \infty$, then (4.12) becomes the constrained K-means problem which coincides with the semi-discrete Monge OT problem. In the rest of 4.4.1, we discuss the solution to (4.12).

When $\lambda \to \infty$, using our OT solver, we can obtain the optimal height vector $\boldsymbol{h}^*$ inducing the optimal power Voronoi diagram that produces the minimum transportation cost while satisfying the marginal constraints. When $\lambda = 0$, the solution to the classic K-means problem comes from a regular, or an unweighted, Voronoi diagram that is being iteratively updated until the center of every Voronoi cell matches the center of the mass enclosed in their cells. A natural thought is to combine the two different results because we expect that the solution to (4.12) lies in between the solution to the classic K-means and the solution to the constrained K-means problem. We prove in Appendix that the following Voronoi diagram indeed solves (4.12).

$$\mathcal{R}_k = \left\{ \|x - y_k\|_2^2 + \frac{\lambda_k}{1 + \lambda_k} h_k^* \leq \|x - y_\ell\|_2^2 + \frac{\lambda_\ell}{1 + \lambda_\ell} h_\ell^* \right\} \tag{4.13}$$

Remark 2. (4.13) *is the minimum point of* (4.12).

The solution to (4.12) itself simplifies the algorithm to achieve it because we can solve for $\lambda = 0$ and $\lambda \to \infty$ separately and then directly combine them to obtain the resulting Voronoi diagram. Still, to complete the problem formulation, we prove



$\lambda = 0$ $\lambda = 0.5$ $\lambda = 2$ $\lambda = \infty$

Figure 4.8: Results From Different Regularization Strength λ in (4.12). Left Is Traditional K-Means and Right Is Constrained K-Means.

in Appendix that the optimizer h, as being updated by solving VOT, monotonically decreases (4.12).

Figure 4.8 illustrates the regularized K-means clustering results at different regularization strengths. Informally, they look like the interpolations between K-means and constrained K-means.

4.4.2 *Co-Clustering*

Extending Wasserstein clustering to multiple targets induces the co-clustering problem. In this section, we discuss the connection between co-clustering problems and VWDs. In particular, we focus on co-clustering spatial features with the quadratic Euclidean distance as the ground metric. In this paper, we generalize the use of the term "co-clustering" not only for two distributions but for multiple, $N \geq 2$.

Given multiple domains, the goal of co-clustering is to simultaneously partition all the domains to 1) minimize the pair-wise variance in the same cluster in each domain and 2) minimize the pair-wise variance for the same cluster across domains. By doing so, we presume that all the domains have the same structure and the same orientation – they are only shifted by a translation. The objective is as follows:

$$\begin{aligned}\min_{\boldsymbol{\mathcal{R}}_i} I_9[\boldsymbol{\mathcal{R}}_i] &\stackrel{\text{def}}{=} \sum_{i=1}^{N}\sum_{k=1}^{K} \frac{1}{2|\mathcal{R}_{i,k}|} \sum_{x,x' \in \mathcal{R}_{i,k}} \|x - x'\|_2^2 \\ &+ \sum_{k=1}^{K} \sum_{1 \leq i < j \leq N} \frac{\lambda_{i,j,k}}{|\mathcal{R}_{i,k}| + |\mathcal{R}_{j,k}|} \sum_{\substack{x \in \mathcal{R}_{i,k} \\ x' \in \mathcal{R}_{j,k}}} \|x - x'\|_2^2.\end{aligned}$$

where $|\mathcal{R}_{i,k}|$ is the number of samples in $\mathcal{R}_{i,k}$; $\lambda_{i,j,k} \in \{0, 1\}$ specifies the correspondence of the clusters across different domains. Thus, $\sum_i \lambda_{i,j,k} = 1$ and $\sum_j \lambda_{i,j,k} = 1$. Similarly to K-means, we can simplify the pairwise variance with the mean of each cluster at each domain, $\alpha_{i,k}$:

$$\begin{aligned}\min_{\boldsymbol{\mathcal{R}}_i} I_9[\boldsymbol{\mathcal{R}}_i] &\equiv \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{x \in \mathcal{R}_{i,k}} \|x - \alpha_{i,k}\|_2^2 \\ &+ \sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j \neq i} \lambda_{i,j,k} \sum_{x \in \mathcal{R}_{i,k}} \|x - \alpha_{j,k}\|_2^2.\end{aligned} \tag{4.14}$$

where $\alpha_{i,k} = \frac{1}{|\mathcal{R}_{i,k}|} \sum_{x \in \mathcal{R}_{i,k}} x$ is the cluster center for each cluster at each domain. The first term of (4.14) is solving N K-means problems. The second term is solving $N(N-1)$ K-means problems but with the cluster centroids at other domains. Thus, we can further simplify the problem into:

$$\min_{\boldsymbol{\mathcal{R}}_{\boldsymbol{i}}} I_9[\boldsymbol{\mathcal{R}}_{\boldsymbol{i}}] \equiv \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{x \in \mathcal{R}_{i,k}}\sum_{j=1}^{N} \|x - \alpha_{j,k}\|_2^2 \tag{4.15}$$

Solving (4.15) involves alternatively updating partition $\{\boldsymbol{\mathcal{R}}_{\boldsymbol{i}}\}_i$ and the centroid $\{\alpha_{i,k}\}_{i,k}$. When updating $\{\boldsymbol{\mathcal{R}}_{\boldsymbol{i}}\}_i$ with fixed $\{\alpha_{i,k}\}_{i,k}$, we can rewrite (4.15) as

$$\begin{aligned}I_{11}[\boldsymbol{\mathcal{R}}_i] &= \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{x \in \mathcal{R}_{i,k}} \left[x - \left[\sum_{j=1}^{N} \alpha_{j,k} \right] \right]^2 + C \\ &\stackrel{\text{def}}{=} \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{x \in \mathcal{R}_{i,k}} (x - \hat{\alpha_k})^2 + C.\end{aligned} \tag{4.16}$$

C is some constant. Thus, we convert co-clustering to N K-means problems with the same set of centroids.

Then, we can naturally impose a constraint on the weights, i.e $\int_{\mathcal{R}_{i,k}} d\mu_i(x) = \nu_k,\ \forall i,\ k$, to turn the problem into a VWB problem which is also an N constrained K-means problem. Note, that it is trivial to extend it into a generalized VWB problem, by instead inserting the weighted constraint into the main objective as we did in 4.4.1.

4.4.3 Regularizing VWBs for Aligning Distributions

For all the domains to have the same structure and orientation is a strong underlying assumption in 4.4.2 that limits our method from solving real-world problems. Here we further discuss regularizing the clustering process so that we can also recover the shift in the orientation should there be any. And we show the process is equivalently aligning marginal distributions altogether.

In addition to purely clustering feature domains according to Wasserstein losses, we can regularize the correspondences based on prior knowledge. Inspired by Alvarez-Melis *et al.* (2019); Mi *et al.* (2018b); Lee *et al.* (2019), we regularize the correspondence by global invariances under the widely accepted assumption that different related distributional data share the same structure but may differ by orientation in their embedding spaces. Directly regularizing Monge correspondences is highly intractable because we use the variational method and thus do not have direct access to the map. The other variable of our clustering process is the centroids each of which is being updated to the critical point, which is the center of mass of its correspondences. Thus, a natural thought is regularizing the centroid update to follow an isometry that preserves the structure of the domain. However, because each marginal may have its own orientation and because isometry is invertible, we transform all the marginals toward a common "barycenter". An isometry consists of a translation and rotation. Estimating the translation can be easily done by shifting the center of mass. Then, the problem boils down to estimating the rotation.

Suppose we have a collection of marginal distributions $\boldsymbol{\mu}_{1:N}$ that share the same structure and differ by orientation in $\mathbb{R}^n$. Thus, there exists a rotation $R_{i,j}$ between μ_i and μ_j. Given a canonical origin, all $R_{i,j}$'s belong to the same rotation group $SO(n) \overset{\text{def}}{=} \{R \in \mathbb{R}^{n\times n} \mid R^T R = RR^T = I_n\}$. Now, we vectorize our notation. We use $X_i \in \mathbb{R}^{N_i \times n}$ to denote the support of the Monte Carlo samples of μ_i and $Y \in \mathbb{R}^{K\times n}$ to denote the support of ν. We use R_i to denote the rotation from μ_i to ν^*. Computing VOT from μ_i to ν produces the partition of X_i into K clusters each with its center of mass. Let us denote the centers as $Y_i \in \mathbb{R}^{K\times n}$ which is a function $Y_i\left(X_i, Y | \boldsymbol{\mathcal{R}}_i\right)$ where $\boldsymbol{\mathcal{R}}_i$ is the VOT partition. Then, we have $Y^* = R_i Y_i$, assuming ν^* share the same structure with all μ_i's. A rotation group is closed w.r.t. composition. Thus, $R_{i,j} = R_j^T R_i$. The problem now is to achieve ν^* which narrows down to Y^* because we fix the measures on the support.

Finally, we are ready to introduce the objective function:

$$\begin{aligned} \min_{Y, \{R_i\}_{i=1}^N} &= \frac{1}{N}\sum_{i=1}^{N} \|Y - R_i Y_i\|^2, \text{ s.t. } R_i \in SO(n) \\ \text{where } Y_i &= Y_i\left(X_i, Y; \boldsymbol{\mathcal{R}}_i\right) = \frac{\int_{\mathcal{R}_{i,k}} x d\mu_i(x)}{\int_{\mathcal{R}_{i,k}} d\mu_i(x)}. \end{aligned} \tag{4.17}$$

We alternatively solve for the optimal support of the barycenter, Y, and associated rotation for each marginal, R_i. We use VWB to solve for Y as in 4.3.1. Once we obtain the center of mass for all marginals, $\{Y_i\}_{i=1}^N$ and Y, we then use singular value decomposition (SVD) to solve for the rotation. The correspondences between $\{Y_i\}_{i=1}^N$ and Y comes from the nature of using a common set of centroids to clustering multiple targets.

4.5 Numerical Evaluation

We evaluate our algorithm from two perspectives – timing and approximation error – against 1) the number of dimensions and 2) the number of centroids. We skip the evaluation of the performance against the number of marginals because, as discussed in 4.3.1, we can either solve for all the VWB variables as a whole which increases the time in the same way as the number of centroids does or solve for each OT problem separately which increases the time linearly to the number of marginals. As for the approximation error, we only test for two marginals because the numerical difference between the pair-wise WD among multiple marginals and VWD has a less practical use. We include the results from linear programming (LP) and the Sinkhorn algorithm as implemented in the POT library Flamary and Courty (2017) as baselines. For the timing, we include a comparison between using one core of a CPU (Intel Core i5-8400 CPU @ 2.80GHz) and a GPU (NVIDIA GeForce RTX 2070). The experiments for testing the approximation error are run either on CPU or GPU, whichever is faster according to the timing experiments.

For all the experiments, we randomly generate two isotropic Gaussian distributions, each having $10,000$ samples. We set the means of the two distributions to differ by $n^{-0.5}$, and the diagonal values of both the covariance matrices to be $0.02 \times n^{-0.5}$. A simple calculation would show that the exact WD between these two Gaussians is "1". The small variances are inherited from previous experiments in 4.3.3 to keep the two distributions separate from each other for better visualization. For each experiment, we record the results from 5 trials.

For each trial, we early-stop the OT after 100 iterations to save the overall time. Fig. 4.9 shows the average time for updating the OT optimizer for one iteration. The top row and the bottom row show the time for CPU and GPU, respectively; the top 1st column and 2nd column show the time for different K's and n's, respectively.

(a) reflects a quadratic relationship between the time and K using CPU, which matches our expectation. (c) indicates that, in general, K does not have a significant impact on the run time when we parallelly update the height vectors for the power

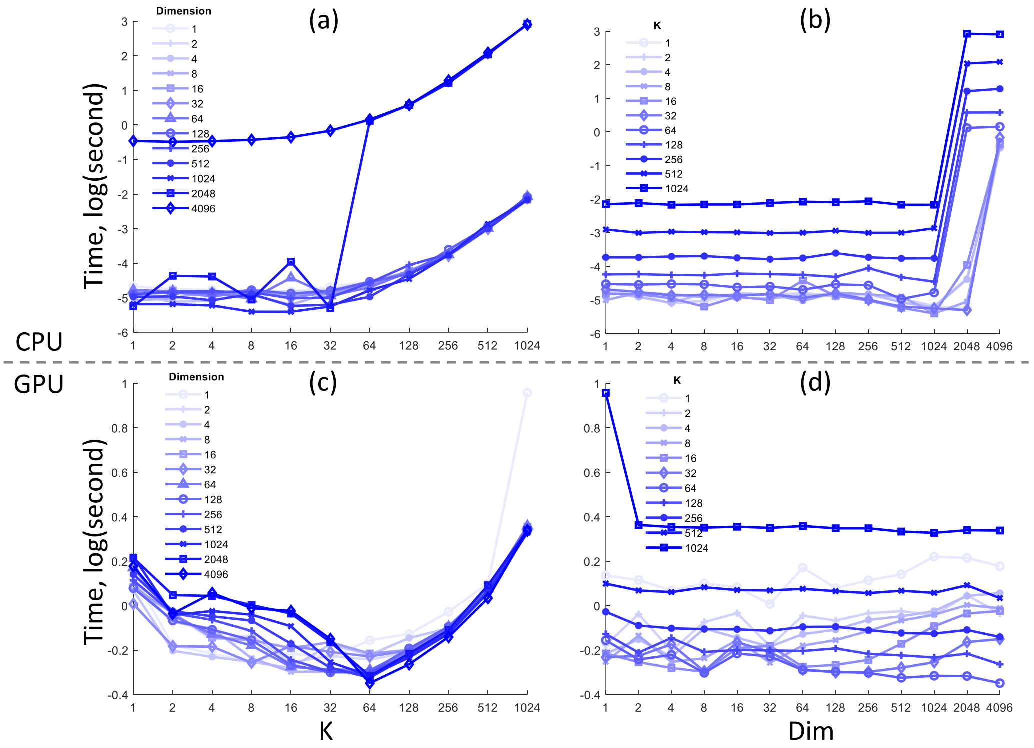


Figure 4.9: Time to Compute VWBs

Voronoi diagrams using GPU. However, as K increases, there is a slight incremental trend. We suspect the reason being related to the implementation of matrix operations in PyTorch and also the capacity of our GPU. We leave more comprehensive experiments to the future with more computational resources. By comparing the absolute time in (a) and (c), we find that computing VWB on GPUs is hugely beneficial when $K > 32$. (b) and (d) indicate that the number of dimensions, if it is not high enough (≤ 1024), does not have a substantial impact on the running time of our method. This is also expected because we use Monte Carlo samples and then vectorize all the computation. Similarly, when the number of dimensions is very high, matrix operations might be slower due to memory management issues.

Although computing VWB on GPUs saves time on a large-scale dataset, transmitting data between GPU memory and hard drive usually takes a significant amount of time. In practice, that could have a certain impact on choosing between GPU and CPU.

4.6 Applications

We demonstrate the use of variational Wasserstein barycenters for image compression and point cloud registration.

4.6.1 Vector Quantization and Data Compression

Lloyd's K-means algorithm was initially proposed for vector quantization and has been a fundamental choice and baseline for data compression. It centers at using fewer samples to approximate the entire distribution. In light of the connection between VOT and K-means, we propose to use VOT to compress distributional data in $\mathbb{R}^n$ and use VWBs to summarize multiple distributions and potentially compress the summary at the same time. In this case, the VWD quantitatively measures the compression error.

By using VOT, we obtain a surjection from each domain to the barycenter. Because we optimize over the height vector $\boldsymbol{h}_i$ (4.2), given empirical samples and the barycenter, we can fully recover the surjection by only using $\boldsymbol{h}_i$ at the negligible expense of computing the power distance as in (2.16). In this way, for a barycenter of size K of N empirical distributions each having M samples, we reduce the burden for storing the barycenter and the correspondence from $\mathcal{O}(NMK)$, as it would be for Sinkhorn distance-based or LP-based methods, to $\mathcal{O}(NK)$. This is particularly useful when M is large and when we need to store multiple interpolations between marginals.

Furthermore, with the VWB, we do not even need the original distributions to

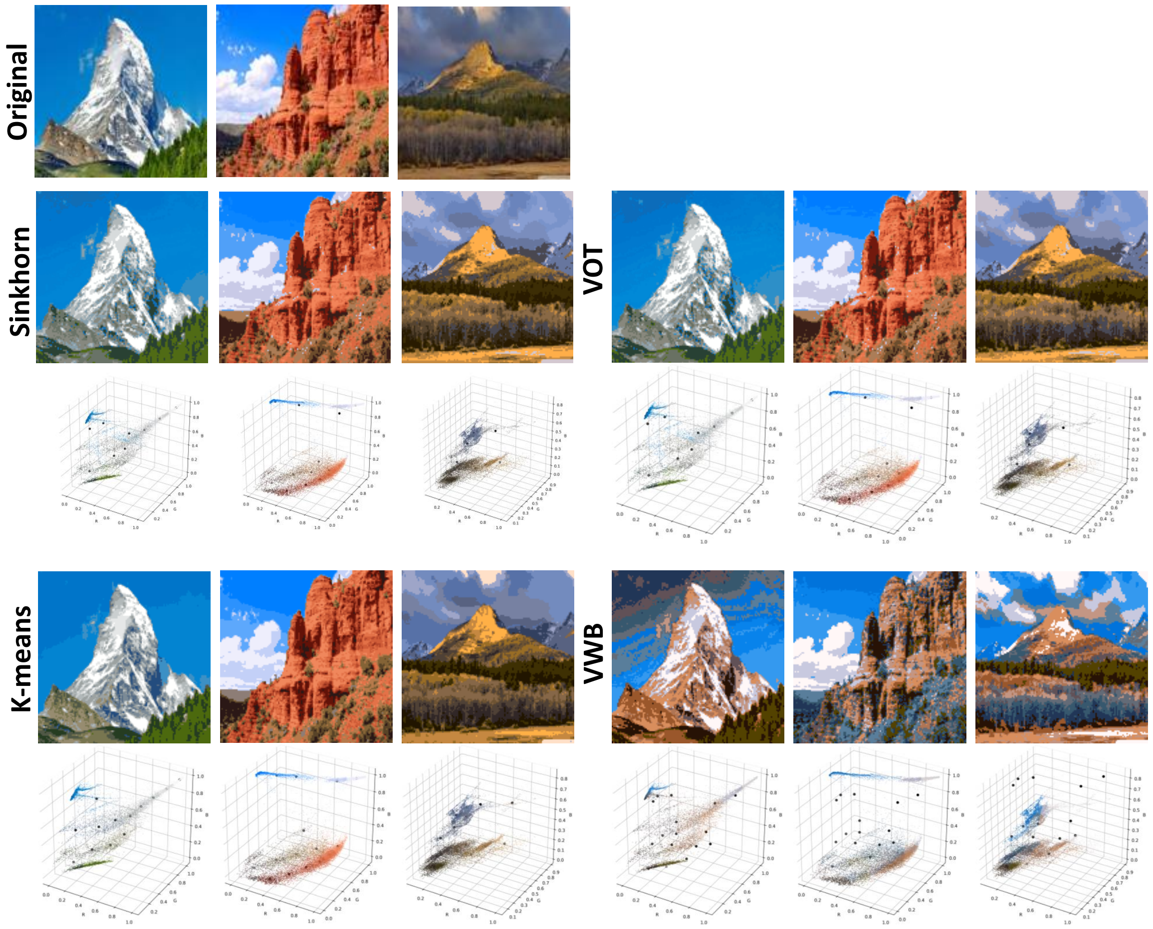


Figure 4.10: Quantizing RGB Values From 24 Bits to 4 Bits by Solving K-Means, OT, and the WB. Solving OT Results in Smoother Images; Solving WBs Can Cluster and Merge Colors at the Same Time.

parameterize the compression maps because our method is based on the geometry of the data and given the height vector $\boldsymbol{h}_i$ and barycenter supports $\boldsymbol{y}$ we can uniquely partition each original domain with a power Voronoi diagram $\boldsymbol{\mathcal{R}}_i$; or, equivalently, the graph of the piece-wise linear function $\theta_{\boldsymbol{h}}(x) = \max_k \{xy_k + h_k\}$.

We demonstrate the use of our method with quantizing the RGB colors of three images into a fixed number of clusters, or bins. See Figure 4.10 for the results. The top row shows the original images of dimension $128^2 \times 3$ whose pixels are embedded in the RGB color space $\mathcal{X} = \{x \in \mathbb{R}^3 \mid \|x\|_\infty \leq 1\}$. Our goal is to compute, for example, $K = 16$ centroids that partition all the pixels into their clusters, so that we reduce the storage for each pixel from 24 bits to 4 bits.

From the second row downward, we show the quantization result from K-means, Sinkhorn OT, and our VOT. Clearly, OT techniques better distribute the centroids so that each quantized color roughly has the same number of pixels, resulting in a smoother image. Sinkhorn OT and VOT deliver very similar results, but the soft assignments from Sinkhorn OT require further discretization for clustering and quantization purposes.

The second row in Figure 4.10 shows the resulting images of using Lloyd's K-means(++) algorithm, and the third row shows the results of using our VOT solver. Compared to Lloyd's, VOT well distributes the centroids into the pixel domain, resulting in a smoother transition from color to color. The correspondences in the color space we show in Appendix also confirm this. Finally, we simultaneously merge and compress the colors from all three images by using VWB. The last row shows the resulting images sharing the same color distribution that only consists of 16 discrete centroids. It has the same $\mathcal{W}_2$ to each original color distribution (marginal). In Appendix, we further show the results that comes from the centroids having different $\mathcal{W}_2$'s to each marginals, i.e. $\lambda_i \neq \frac{1}{N}$ in (4.6).

4.6.2 Multi-marginal Distributional Alignment

Aligning distributional data plays an essential role in many machine learning problems. We have seen enormous discussions in the past literature on aligning two distributions. In recent years, more researchers focus on aligning multiple distributions simultaneously to find a common anchor, and it has potentials on domain adaptation

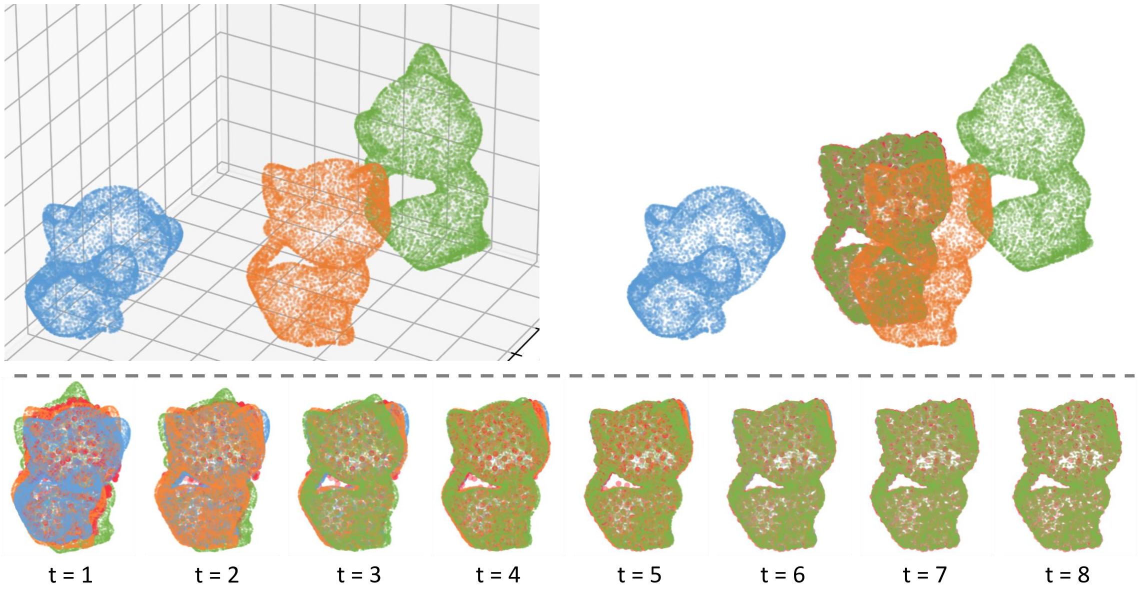


Figure 4.11: Aligning Three Point Clouds While Preserving Their Structure.

and generative modeling in a more general setting, e.g., Cao *et al.* (2019). Here we demonstrate the use of our method by registering multiple point clouds.

After we partition each domain into K clusters as we discussed in 4.4.3, the center of mass for each cluster is a linear average. Rotation is a linear operation. Therefore, applying the rotation to all the samples of each marginal distribution creates the same movement as to the centers of masses. In this way, we can align multiple marginal distributions altogether. Fig. 4.11 shows an example where we align three kittens, each in a different position and orientation. The top half shows the final barycentric centroids depicted in red dots in the middle that roughly lie in the center of the three original kittens and with a "averaged" orientation. The bottom four sub-figures show the process of the alignment. After 8 iterations of updating the centroids and the orientation of the marginals, eventually, all the marginals are perfectly aligned together with the centroids.

We are given three Kittens off by an unknown rigid transformation. Our goal is to interpolate, by computing a regularized Wasserstein barycenter, a new Kitten in between that is rigid to the original Kittens, and the amount of translation and rotation is linear to the weights of the two original Kittens.

The marginal Kittens each have $7,805$ sample points. We assume all the samples have equal weights. They are apart from each other by a rigid transformation composed by a random translation and a random rotation.

The barycenter Kitten w.r.t. the VWD (variational Wasserstein distance) has 780 supporting Dirac measures. The regularization strength, λ, is 10. One of the post-processing options to transport all the samples from the marginals is that for each sample, find its nearest 3 or more cluster centers and use inverse barycenter coordinates to find its new location on the target Kitten in the middle.

4.7 Discussion

We conclude this chapter by discussing the advantages and disadvantages of VWBs and several future directions.

Algorithms solving K-means like clustering problems are in general sensitive to initial choices. Typical solutions include using a subset of samples and spreading the seeds across the domain, e.g., K-means++. We tried the results from K-means++ as the initial choice for our barycenters and also tried a pre-defined Gaussian distribution whose mean is the average of the means of the marginals as prior knowledge. We did not find visible differences.

Monge maps between discrete measures may not exist, e.g. transporting 3 Dirac points $\{\frac{1}{3}\delta[x_j]\}_{j=1}^{3}$ to 2 Dirac points $\{\frac{1}{2}\delta[y_j]\}_{k=1}^{2}$. In this case, splitting the mass becomes necessary Wang *et al.* (2013). Moreover, there might be multiple solutions, and variational solvers cannot recover any of them. An example is transporting

$\{\frac{1}{2}\delta[x_1 = (0,-1)], \frac{1}{2}\delta[x_2 = (0,1)]\}$ to $\{\frac{1}{2}\delta[y_1 = (1,0)], \frac{1}{2}\delta[y_2 = (2,0)]\}$. There exist two one-to-one maps but VOT cannot recover either because the target measures cannot be distinguished by the piece-wise linear function $\theta_{\boldsymbol{h}}(x) = \max_k\{xy_k + h_k\}$, in 2.4. Therefore, when dealing with stochastic GD, having sufficient samples to represent the domain is key to stabilize VWBs. Luckily, increasing the empirical samples adds little computational burden if we parallelly update the correspondence for each empirical according to its nearest neighbor. On the other hand, Sinkhorn iteration-based OT methods produce soft correspondences that unavoidably result from the entropic regularization, making them robust for discrete measures. Occasionally, the soft correspondences are even desirable because they make the correspondences differentiable Cuturi *et al.* (2019); Monge correspondences, however, are basically permutations which are not differentiable. In summary, our VWB producing Monge maps is suitable for clustering or partitioning problems that require binary, sparse correspondence while Sinkhorn distance-based barycenters have been tested in numerous applications in machine learning for producing robust interpolations.

There are several future directions: 1) In the current implementation, we use exhaustive search to find the nearest centroid for each empirical sample, which takes about 80% of our run time. A faster alternative for nearest neighbor search based on the power distance, which is not a Minkowski distance, will significantly reduce the run time of the VWB; 2) Whether VWBs or WBs for unbalanced measures still induce a generalized metric deserves an answer; 3) Whether our discussion still holds for $1 \leq p < 2$ and $p > 2$ deserves an answer; 4) Another branch of computing Monge OT is the multi-scale approach, e.g., Mérigot (2011); Schmitzer (2016); Gerber and Maggioni (2017). It also partitions the target domain into sub-domains. Computing barycenters with multi-scale OT for clustering purposes is worth exploring.

4.8 Appendix

Critical Point of VWBs w.r.t. $\{\nu_\ell\}_{\ell=1}^K$:

$$\nabla I_5[\boldsymbol{h_i}] = \left\{ \frac{\partial I_5}{\partial h_{i,\ell}} = \int_{\mathcal{R}_{i,\ell}} d\mu_i(x) - \nu_\ell \right\}_{\ell=1}^K,$$

$$\boldsymbol{h_i}^{(t+1)} = \boldsymbol{h_i}^{(t)} - \eta \nabla I_5[\boldsymbol{h_i}].$$

$$y_\ell^* = \frac{\sum_{i=1}^N \int_{\mathcal{R}_{i,\ell}} x d\mu_i(x)}{N \sum_{i=1}^N \int_{\mathcal{R}_{i,\ell}} d\mu_i(x)} \approx \frac{\sum_{i=1}^N \sum_{x \in \mathcal{R}_{i,\ell}} x \mu_i(x)}{N \sum_{i=1}^N \sum_{x \in \mathcal{R}_{i,\ell}} \mu_i(x)},$$

$$\nu_\ell^* = \frac{1}{N} \sum_{i=1}^N \int_{\mathcal{R}_{i,\ell}^*} d\mu_i(x) \approx \frac{1}{N} \sum_{i=1}^N \sum_{x \in \mathcal{R}_{i,\ell}^*} \mu_i(x),$$

We update ν_ℓ so that $\mathcal{R}_i \ \forall \ i$ induced by the optimal T_i^* forms an unweighted Voronoi diagram where each empirical sample x is mapped to its nearest y_ℓ w.r.t. the quadratic Euclidean distance. Now, by contradiction, suppose we update ν_ℓ in such as way that the OT map does not form an unweighted Voronoi diagram, i.e., $T_i' \neq T_i^*$. Then, there must be some sample x that is mapped to some $y_k \neq y_\ell$ which is not the nearest centroid. In this case, the total transportation cost increases because $\|x - y_k\|_2^2 > \|x - y_\ell\|_2^2$. Therefore, the minimum total cost w.r.t. ν_ℓ is achieved when ν_ℓ induces the OT map that forms an unweighted Voronoi diagram. When we update ν_ℓ, we only need to construct an unweighted Voronoi diagram according to the current supports $\boldsymbol{y}$ and assign the total mass within each cell to its corresponding support. This Voronoi diagram itself is also the OT map, like Lloyd's algorithm.

Derivatives of VWBs.

$$\min_{\{\boldsymbol{h_i}\}_{i=1}^N} I_5[\nu] \overset{\text{def}}{=} \frac{1}{N} \sum_{i=1}^N \left(\int_{\boldsymbol{0}}^{\boldsymbol{h_i}} \sum_{\ell=1}^K \int_{\mathcal{R}_{i,\ell}} d\mu_i(x) dh_{i,\ell} - \sum_{\ell=1}^K \nu(y_\ell) h_{i,\ell} \right)$$

The gradient of $I_5[\boldsymbol{h}]$ is

$$\nabla I_5[\boldsymbol{h}] = \left\{ \left\{ \frac{\partial I_5}{\partial h_{i,\ell}} \right\}_{\ell=1}^{K} \right\}_{i=1}^{N} = \left\{ \left\{ \int_{\mathcal{R}_{i,\ell}} d\mu_i(x) - \nu(y_\ell) \right\}_{\ell=1}^{K} \right\}_{i=1}^{N}.$$

The Hessian of $I_5[\boldsymbol{h}]$ is then

$$H = \left(\frac{\partial^2 I_5[\boldsymbol{h}]}{\partial h_{i,\ell} \partial h_{j,k}} \right) = \begin{cases} \sum_k \dfrac{\int_{f_{i,\ell,k}} \mu(x)dx}{\|y_\ell - y_k\|}, & i = j, \ \forall k, s.t. \ f_{i,\ell,k} \neq \emptyset, \\ -\dfrac{\int_{f_{i,\ell,k}} \mu_i(x)dx}{\|y_\ell - y_k\|}, & i = j, \ f_{i,\ell,k} \neq \emptyset, \\ 0, & i \neq j. \end{cases}$$

where $f_{i,\ell,k} = \mathcal{R}_{i,\ell} \cap \mathcal{R}_{i,k}$.

This is a very sparse matrix since variables from different VOT problem, i, j, are excluded from each other, and in each VOT problem, one power Voronoi cell is only adjacent to a few other cells.

$$\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N}) \stackrel{\text{def}}{=} \inf_{\nu \in \mathcal{P}(\mathcal{Y})} \frac{1}{N} \sum_{i=1}^{N} \mathcal{W}_2^2(\mu_i, \nu),$$

Proposition 8. *$\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N})$ is a generalized metric among $\boldsymbol{\mu}_{1:N}$, $N \geq 2$. Specifically, $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N})$ satisfies the following properties.*

1) Non-negativity: $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N}) \geq 0$.

2) Symmetry: $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{\sigma_1(1:N)}) = \mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{\sigma_2(1:N)})$, where $\sigma_1(1:N)$ and $\sigma_2(1:N)$ are different permutations of the set $1:N$.

3) Identity: $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N}) = 0 \Longleftrightarrow \mu_i = \mu_j, \forall i \neq j$.

4) Triangle inequality: $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N}) \leq \sum_{i=1}^{N} \mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N+1\backslash i})$.

Proof. The first three properties are the immediate result of the metric nature of the Wasserstein distance.

1) Non-negativity: $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N}) \geq 0$.

Since $\mathcal{W}_2^2(\mu_i, \nu^*) \geq 0 \ \forall\ i$, $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N})$ is obviously not negative. The equal sign holds when $\mathcal{W}_2^2(\mu_i, \nu^*) = 0 \ \forall\ i$. When that happens, $\mu_i = \nu^*$ for all i. It also implies that all marginals are equal to each other, i.e. $\mu_i = \mu_j, \ \forall\ i \neq j$.

2) Symmetry: $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{\sigma_1(1:N)}) = \mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{\sigma_2(1:N)})$

This is true since summation is symmetric. Because the order does not matter, we use $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N})$ to omit it.

3) Identity: $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N}) = 0 \Longleftrightarrow \mu_i = \mu_j, \forall i \neq j$

This is true, according to our discussion in 1).

4) Triangle inequality: $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N}) \leq \sum_{i=1}^{N} \mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:N+1\backslash i})$

We split the proof into two cases: $N = 2$ and $N > 2$.

$N = 2$. Suppose we have three probability distributions μ_1, μ_2, μ_3 and their pairwise WBs $\nu^{(12)*}, \nu^{(23)*}, \nu^{(31)*}$. According to McCann (1997), $\nu^{(12)*}$ is on the geodesic between μ_1 and μ_2 and thus $\mathcal{W}_2(\mu_1, \nu^{(12)*}) = \mathcal{W}_2(\mu_2, \nu^{(12)*}) = \frac{1}{2}\mathcal{W}_2(\mu_1, \mu_2)$. Similarly, $\mathcal{W}_2(\mu_2, \nu^{(23)*}) = \mathcal{W}_2(\mu_3, \nu^{(23)*}) = \frac{1}{2}\mathcal{W}_2(\mu_2, \mu_3)$ and $\mathcal{W}_2(\mu_3, \nu^{(31)*}) = \mathcal{W}_2(\mu_1, \nu^{(31)*}) = \frac{1}{2}\mathcal{W}_2(\mu_3, \mu_1)$. Thus, $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:2}) = \frac{1}{4}\mathcal{W}_2^2(\mu_1, \mu_2)$, $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{2:3}) = \frac{1}{4}\mathcal{W}_2^2(\mu_2, \mu_3)$, and $\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{3:1}) = \frac{1}{4}\mathcal{W}_2^2(\mu_3, \mu_1)$. According to triangle inequality, $\mathcal{W}_2^2(\mu_1, \mu_2) \leq \mathcal{W}_2(\mu_1, \mu_2) + \mathcal{W}_2(\mu_1, \mu_2)$, then

$\mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{1:2}) \leq \mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{2:3}) + \mathcal{D}_{\nu^*}(\boldsymbol{\mu}_{3:1})$

Figure 4.12 reveals the triangle inequality for $N = 2$. Suppose $\boldsymbol{\mu}_{1:3}$ are three marginals and $\boldsymbol{\mu}_{4:6}$ are the corresponding Wasserstein barycenters of each two marginals. To ease the reading, we temporarily use $\mathcal{W}_{i,j}$ to represent WDs between marginals $\boldsymbol{\mu}_{i,j}$. It is straightforward to show that $\mathcal{W}_{1,4} + \mathcal{W}_{2,4} \leq \mathcal{W}_{2,5} + \mathcal{W}_{3,5} + \mathcal{W}_{1,6} + \mathcal{W}_{3,6}$. This is indeed true. First, $\mathcal{W}_{1,6} + \mathcal{W}_{3,6} + \mathcal{W}_{3,5} \geq \mathcal{W}_{1,6} + \mathcal{W}_{5,6} \geq \mathcal{W}_{1,5}$, thanks to the triangle inequality of the Wasserstein distance. Then, $\mathcal{W}_{1,5} + \mathcal{W}_{2,5} \geq \mathcal{W}_{1,4} + \mathcal{W}_{2,4}$ because of the definition of the Wasserstein barycenter where the barycenter induces

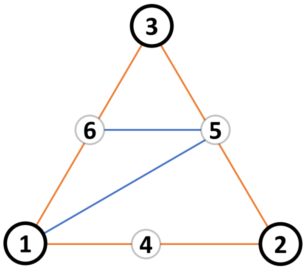


Figure 4.12: Triangle Inequality for $N = 2$.

the minimum of the total Wasserstein distance between the marginals which in this case are μ_1 and μ_2 .

$\square$

Chapter 5

CONCLUSION

In this thesis, we present a family of new transportation techniques for geometric clustering. We developed new computational methods to solve optimal transportation problems. Our methods built upon the variational principle and advanced the algorithmic development on regularizing Monge optimal transportation and the Monge Wasserstein barycenter problems. The natural connections we discovered between Monge problems and geometric clustering enables our methods to perform various tasks. We evaluated our methods from different perspectives and demonstrated their flexibility and reliability in remeshing, domain adaptation, vector quantization, point cloud registration, and medical image analysis.

Our variational approach closes the gaps between the Monge formulation and several clustering problems that have been exploited from the perspective of the Kantorovich optimal transportation formulation. It also opens up several new directions for future work. We are extending our methods to clustering distributional data with hierarchical structures. It is worth exploring computing Monge optimal transportation and Wasserstein barycenters on general manifolds with Riemannian metrics by defining generalized power Voronoi diagrams. We also expect more discussions on the metric properties of multi-marginal optimal transportation for unbalanced measures to generalize our framework.

REFERENCES

Agueh, M. and G. Carlier, “Barycenters in the Wasserstein space”, SIAM Journal on Mathematical Analysis **43**, 2, 904–924 (2011).

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

Álvarez-Esteban, P. C., E. Del Barrio, J. Cuesta-Albertos and C. Matrán, “A fixed-point approach to barycenters in Wasserstein space”, Journal of Mathematical Analysis and Applications **441**, 2, 744–762 (2016).

Alvarez-Melis, D., S. Jegelka and T. S. Jaakkola, “Towards optimal transport with global invariances”, in “Proceedings of Machine Learning Research”, edited by K. Chaudhuri and M. Sugiyama, vol. 89 of *Proceedings of Machine Learning Research*, pp. 1870–1879 (PMLR, 2019).

Ambrosio, L., N. Gigli and G. Savaré, *Gradient flows: in metric spaces and in the space of probability measures* (Springer Science & Business Media, 2008).

Anderes, E., S. Borgwardt and J. Miller, “Discrete Wasserstein barycenters: Optimal transport for discrete data”, Mathematical Methods of Operations Research **84**, 2, 389–409 (2016).

Applegate, D., T. Dasu, S. Krishnan and S. Urbanek, “Unsupervised clustering of multidimensional distributions using earth mover distance”, in “Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining”, pp. 636–644 (ACM, 2011).

Arjovsky, M., S. Chintala and L. Bottou, “Wasserstein generative adversarial networks”, in “International Conference on Machine Learning”, pp. 214–223 (2017).

Arthur, D. and S. Vassilvitskii, “k-means++: The advantages of careful seeding”, in “Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms”, pp. 1027–1035 (Society for Industrial and Applied Mathematics, 2007).

Aurenhammer, F., “Power diagrams: properties, algorithms and applications”, SIAM Journal on Computing **16**, 1, 78–96 (1987).

Auricchio, G., F. Bassetti, S. Gualandi and M. Veneroni, “Computing Kantorovich-Wasserstein distances on d-dimensional histograms using $(d+1)$-partite graphs”, in “Advances in Neural Information Processing Systems”, pp. 5793–5803 (2018).

Beecks, C., A. M. Ivanescu, S. Kirchhoff and T. Seidl, “Modeling image similarity by Gaussian mixture models and the signature quadratic form distance”, in “Computer Vision (ICCV), 2011 IEEE International Conference On”, pp. 1754–1761 (IEEE, 2011).

Benamou, J.-D., “Numerical resolution of an “unbalanced” mass transport problem”, ESAIM: Mathematical Modelling and Numerical Analysis **37**, 5, 851–868 (2003).

Bengio, Y., A. Courville and P. Vincent, “Representation learning: A review and new perspectives”, IEEE transactions on pattern analysis and machine intelligence **35**, 8, 1798–1828 (2013).

Bradley, P. S., K. P. Bennett and A. Demiriz, “Constrained k-means clustering”, Microsoft Research, Redmond **20**, 0, 0 (2000).

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

Cao, J., L. Mo, Y. Zhang, K. Jia, C. Shen and M. Tan, “Multi-marginal wasserstein gan”, in “Advances in Neural Information Processing Systems”, pp. 1774–1784 (2019).

Claici, S., E. Chien and J. Solomon, “Stochastic Wasserstein barycenters”, in “International Conference on Machine Learning”, pp. 999–1008 (2018).

Courty, N., R. Flamary and M. Ducoffe, “Learning wasserstein embeddings”, in “International Conference on Learning Representations”, (2018).

Courty, N., R. Flamary, A. Habrard and A. Rakotomamonjy, “Joint distribution optimal transportation for domain adaptation”, in “Advances in Neural Information Processing Systems”, pp. 3730–3739 (2017a).

Courty, N., R. Flamary and D. Tuia, “Domain adaptation with regularized optimal transport”, in “Joint European Conference on Machine Learning and Knowledge Discovery in Databases”, pp. 274–289 (Springer, 2014).

Courty, N., R. Flamary, D. Tuia and A. Rakotomamonjy, “Optimal transport for domain adaptation”, IEEE transactions on pattern analysis and machine intelligence **39**, 9, 1853–1865 (2017b).

Cui, L., X. Qi, C. Wen, N. Lei, X. Li, M. Zhang and X. Gu, “Spherical optimal transportation”, Computer-Aided Design **115**, 181 – 193 (2019).

Cuingnet, R., E. Gerardin, J. Tessieras, G. Auzias, S. Lehéricy, M.-O. Habert, M. Chupin, H. Benali and O. Colliot, “Automatic classification of patients with Alzheimer’s disease from structural MRI: a comparison of ten methods using the ADNI database”, Neuroimage **56**, 2, 766–781 (2011).

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

Cuturi, M. and A. Doucet, “Fast computation of Wasserstein barycenters”, in “International Conference on Machine Learning”, pp. 685–693 (2014).

Cuturi, M., O. Teboul and J.-P. Vert, “Differentiable ranking and sorting using optimal transport”, in “Advances in Neural Information Processing Systems”, pp. 6858–6868 (2019).

De Goes, F., K. Breeden, V. Ostromoukhov and M. Desbrun, “Blue noise through optimal transport”, ACM Transactions on Graphics (TOG) **31**, 6, 1–11 (2012).

Fabri, A. and S. Pion, “Cgal: The computational geometry algorithms library”, in “Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems”, pp. 538–539 (ACM, 2009).

Ferradans, S., N. Papadakis, G. Peyré and J.-F. Aujol, “Regularized discrete optimal transport”, SIAM Journal on Imaging Sciences **7**, 3, 1853–1882 (2014).

Fischl, B., “Freesurfer”, Neuroimage **62**, 2, 774–781 (2012).

Flamary, R. and N. Courty, “POT Python optimal transport library”, URL `https://github.com/rflamary/POT` (2017).

Forgy, E. W., “Cluster analysis of multivariate data: efficiency versus interpretability of classifications”, biometrics **21**, 768–769 (1965).

Frogner, C., C. Zhang, H. Mobahi, M. Araya and T. A. Poggio, “Learning with a Wasserstein loss”, in “Advances in Neural Information Processing Systems”, pp. 2053–2061 (2015).

Gerber, S. and M. Maggioni, “Multiscale strategies for computing optimal transport”, The Journal of Machine Learning Research **18**, 1, 2440–2471 (2017).

Gibbs, A. L. and F. E. Su, “On choosing and bounding probability metrics”, International statistical review **70**, 3, 419–435 (2002).

Givens, C. R., R. M. Shortt *et al.*, “A class of Wasserstein metrics for probability distributions.”, The Michigan Mathematical Journal **31**, 2, 231–240 (1984).

Goes, F. d., P. Memari, P. Mullen and M. Desbrun, “Weighted triangulations for geometry processing”, ACM Transactions on Graphics (TOG) **33**, 3, 28 (2014).

Gopalan, R., R. Li and R. Chellappa, “Domain adaptation for object recognition: An unsupervised approach”, in “Computer Vision (ICCV), 2011 IEEE International Conference on”, pp. 999–1006 (IEEE, 2011).

Graf, S. and H. Luschgy, *Foundations of quantization for probability distributions* (Springer, 2007).

Grippo, L. and M. Sciandrone, “On the convergence of the block nonlinear Gauss–Seidel method under convex constraints”, Operations research letters **26**, 3, 127–136 (2000).

Gu, X., F. Luo, J. Sun and S.-T. Yau, “Variational principles for minkowski type problems, discrete optimal transport, and discrete Monge-Ampere equations”, arXiv preprint arXiv:1302.5472 (2013).

Gu, X. D. and S.-T. Yau, *Computational conformal geometry* (International Press Somerville, Mass, USA, 2008).

Gudbjartsson, H. and S. Patz, “The Rician distribution of noisy MRI data”, Magn Reson Med. **34**, 6, 910–914 (1995).

Ho, N., X. Nguyen, M. Yurochkin, H. H. Bui, V. Huynh and D. Phung, “Multilevel clustering via Wasserstein means”, in “International Conference on Machine Learning”, pp. 1501–1509 (2017).

Jack Jr, C. R., M. A. Bernstein, N. C. Fox, P. Thompson, G. Alexander, D. Harvey, B. Borowski, P. J. Britson, J. L. Whitwell, C. Ward *et al.*, “The alzheimer’s disease neuroimaging initiative (adni): Mri methods”, Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine **27**, 4, 685–691 (2008).

Jagust, W. J., D. Bandy, K. Chen, N. L. Foster, S. M. Landau, C. A. Mathis, J. C. Price, E. M. Reiman, D. Skovronsky, R. A. Koeppe *et al.*, “The alzheimer’s disease neuroimaging initiative positron emission tomography core”, Alzheimer’s & Dementia **6**, 3, 221–229 (2010).

Kantorovich, L. V., “On the translocation of masses”, in “Dokl. Akad. Nauk SSSR”, vol. 37, pp. 199–201 (1942).

Kolouri, S., G. K. Rohde and H. Hoffmann, “Sliced Wasserstein distance for learning Gaussian mixture models”, in “Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition”, pp. 3427–3436 (2018).

Leclaire, A. and J. Rabin, “A fast multi-layer approximation to semi-discrete optimal transport”, in “International Conference on Scale Space and Variational Methods in Computer Vision”, pp. 341–353 (Springer, 2019).

Lee, J., M. Dabagia, E. Dyer and C. Rozell, “Hierarchical optimal transport for multimodal distribution alignment”, in “Advances in Neural Information Processing Systems”, pp. 13453–13463 (2019).

Lee, K., W. Xu, F. Fan and Z. Tu, “Wasserstein introspective neural networks”, in “The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)”, (2018).

Lévy, B., “A numerical algorithm for l2 semi-discrete optimal transport in 3d”, ESAIM: Mathematical Modelling and Numerical Analysis **49**, 6, 1693–1715 (2015).

Liero, M., A. Mielke and G. Savaré, “Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures”, Inventiones mathematicae **211**, 3, 969–1117 (2018).

Ling, H. and K. Okada, “An efficient earth mover’s distance algorithm for robust histogram comparison”, IEEE transactions on pattern analysis and machine intelligence **29**, 5, 840–853 (2007).

Lloyd, S., “Least squares quantization in PCM”, IEEE transactions on information theory **28**, 2, 129–137 (1982).

Ma, J., J. Zhao and A. L. Yuille, “Non-rigid point set registration by preserving global and local structures”, IEEE Transactions on image Processing **25**, 1, 53–64 (2016).

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

Mérigot, Q., “A multiscale approach to optimal transport”, Computer Graphics Forum **30**, 5, 1583–1592 (2011).

Mérigot, Q., A. Delalande and F. Chazal, “Quantitative stability of optimal transport maps and linearization of the 2-wasserstein space”, arXiv preprint arXiv:1910.05954 (2019).

Mi, L., W. Zhang, X. Gu and Y. Wang, “Variational Wasserstein clustering”, in “Proceedings of the European Conference on Computer Vision (ECCV)”, pp. 322–337 (2018a).

Mi, L., W. Zhang and Y. Wang, “Regularized Wasserstein means for aligning distributional data”, arXiv preprint arXiv:1812.00338 (2018b).

Monge, G., “Mémoire sur la théorie des déblais et des remblais”, Histoire de l’Académie Royale des Sciences de Paris (1781).

Murtagh, F., “A survey of recent advances in hierarchical clustering algorithms”, The computer journal **26**, 4, 354–359 (1983).

Myronenko, A. and X. Song, “Point set registration: Coherent point drift”, IEEE TPAMI (2010).

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

Papadakis, N., “Approximation of Wasserstein distance with transshipment”, arXiv preprint arXiv:1901.09400 (2019).

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, “Pytorch: An imperative style, high-performance deep learning library”, in “Advances in Neural Information Processing Systems”, pp. 8024–8035 (2019).

Peyré, G., M. Cuturi *et al.*, “Computational optimal transport”, Foundations and Trends® in Machine Learning **11**, 5-6, 355–607 (2019).

Piccoli, B. and F. Rossi, “Generalized wasserstein distance and its application to transport equations with source”, Archive for Rational Mechanics and Analysis **211**, 1, 335–358 (2014).

Quigley, A. and P. Eades, “Fade: Graph drawing, clustering, and visual abstraction”, in “International Symposium on Graph Drawing”, pp. 197–210 (Springer, 2000).

Ridgway, G., “Rice/Rician distribution”, http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/ (2007).

Rubner, Y., C. Tomasi and L. J. Guibas, “The earth mover’s distance as a metric for image retrieval”, International journal of computer vision **40**, 2, 99–121 (2000).

Rycroft, C., “Voro++: A three-dimensional voronoi cell library in c++”, Tech. rep., Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States) (2009).

Saenko, K., B. Kulis, M. Fritz and T. Darrell, “Adapting visual category models to new domains”, in “European conference on computer vision”, pp. 213–226 (Springer, 2010).

Schmitzer, B., “A sparse multiscale algorithm for dense optimal transport”, Journal of Mathematical Imaging and Vision **56**, 2, 238–259 (2016).

Schroff, F., D. Kalenichenko and J. Philbin, “Facenet: A unified embedding for face recognition and clustering”, in “Proceedings of the IEEE conference on computer vision and pattern recognition”, pp. 815–823 (2015).

Shewchuk, J. R., “Delaunay refinement algorithms for triangular mesh generation”, Computational geometry **22**, 1-3, 21–74 (2002).

Solomon, J., F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du and L. Guibas, “Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains”, ACM Transactions on Graphics (TOG) **34**, 4, 66 (2015).

Staib, M., S. Claici, J. M. Solomon and S. Jegelka, “Parallel streaming Wasserstein barycenters”, in “Advances in Neural Information Processing Systems”, pp. 2647–2658 (2017).

Stark, J. A., “Adaptive image contrast enhancement using generalizations of histogram equalization”, IEEE Transactions on image processing **9**, 5, 889–896 (2000).

Sugihara, K., “Laguerre voronoi diagram on the sphere”, Journal for Geometry and Graphics **6**, 1, 69–81 (2002).

Sun, B. and K. Saenko, “Deep coral: Correlation alignment for deep domain adaptation”, in “European Conference on Computer Vision”, pp. 443–450 (Springer, 2016).

Ulen, J., P. Strandmark and F. Kahl, “Shortest paths with higher-order regularization”, IEEE transactions on pattern analysis and machine intelligence **37**, 12, 2588–2600 (2015).

Villani, C., *Topics in optimal transportation*, no. 58 (American Mathematical Soc., 2003).

Wang, W., D. Slepčev, S. Basu, J. A. Ozolek and G. K. Rohde, “A linear optimal transportation framework for quantifying and visualizing variations in sets of images”, International journal of computer vision **101**, 2, 254–269 (2013).

Wang, Y., X. Gu, T. F. Chan, P. M. Thompson and S.-T. Yau, “Volumetric harmonic brain mapping”, in “Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on”, pp. 1275–1278 (IEEE, 2004).

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

Xu, R. and D. Wunsch, “Survey of clustering algorithms”, IEEE Transactions on neural networks **16**, 3, 645–678 (2005).

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

Ye, J., P. Wu, J. Z. Wang and J. Li, “Fast discrete distribution clustering using Wasserstein barycenter with sparse support”, IEEE Transactions on Signal Processing **65**, 9, 2317–2332 (2017).