On Geometric Prototype and Applications

Hu Ding
Department of Computer Science and Engineering, Michigan State University
East Lansing, USA; and
School of Computer Science and Technology, University of Science and Technology of China
Hefei, China
huding@msu.edu, huding@ustc.edu.cn

Manni Liu
Department of Computer Science and Engineering, Michigan State University
East Lansing, USA
liumanni@msu.edu

Abstract

In this paper, we propose to study a new geometric optimization problem called the "geometric prototype" in Euclidean space. Given a set of patterns, where each pattern is represented by a (weighted or unweighted) point set, the geometric prototype can be viewed as the "average pattern" minimizing the total matching cost to them. As a general model, the problem finds many applications in real-world, such as Wasserstein barycenter and ensemble clustering. The dimensionality could be either constant or high, depending on the applications. To our best knowledge, the general geometric prototype problem has yet to be seriously considered by the theory community. To bridge the gap between theory and practice, we first show that a small core-set can be obtained to substantially reduce the data size. Consequently, any existing heuristic or algorithm can run on the core-set to achieve a great improvement on the efficiency. As a new application of core-set, it needs to tackle a couple of challenges particularly in theory. Finally, we test our method on both image and high dimensional clustering datasets; the experimental results remain stable even if we run the algorithms on core-sets much smaller than the original datasets, while the running times are reduced significantly.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases prototype, core-set, Wasserstein barycenter, ensemble clustering

Digital Object Identifier 10.4230/LIPIcs.ESA.2018.23

Related Version A full version of the paper is available at [21], `https://arxiv.org/abs/1804.09655`.

Acknowledgements The research of this work was supported in part by NSF through grant CCF-1656905 and a start-up fund from Michigan State University. The authors also want to thank the anonymous reviewers for their helpful comments and suggestions for improving the paper.

1 Introduction

Given a set of points in Euclidean space, we can easily use the geometric mean or median point to represent them. However, if they are replaced by a set of point sets where each point set denotes a "pattern", the problem of finding their representation will be much more challenging. We call it the "geometric prototype" problem. Before introducing its formal definition, we need to define the matching cost between two patterns first.

26th Annual European Symposium on Algorithms (ESA 2018).
Editors: Yossi Azar, Hannah Bast, and Grzegorz Herman; Article No. 23; pp. 23:1–23:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

► **Definition 1** ($\mathcal{M}(A,B)$)**.** Given two point sets $A=\{a_1,a_2,\cdots,a_k\}$ and $B=\{b_1,b_2,\cdots,b_k\}$ in $\mathbb{R}^d$,

$$\mathcal{M}(A,B)=\min_{\pi\in\Pi}\sum_{j=1}^{k}||a_j-b_{\pi(j)}||^2 \tag{1}$$

where Π contains all the possible permutations of $\{1,2,\cdots,k\}$.

$\mathcal{M}(A,B)$ is in fact the problem of geometric matching which can be optimally solved by the Hungarian algorithm [16]. When the dimensionality is constant, a number of efficient approximation algorithms have been developed in past years (see more discussion in Section 1.1).

► **Definition 2** (Geometric Prototype)**.** Given a set of point sets $\mathbb{P}=\{P_1,P_2,\cdots,P_n\}$ with each P_i containing k points $\{p^i_1,p^i_2,\cdots,p^i_k\}\subset\mathbb{R}^d$, the geometric prototype is a new point set $g(\mathbb{P})$ having k points such that

$$\sum_{i=1}^{n}\mathcal{M}(P_i,g(\mathbb{P})) \tag{2}$$

is minimized. Note that $g(\mathbb{P})$ can be a new pattern not from $\mathbb{P}$. Also, any k-point set achieving at most c times the minimum value of (2) is called a c-approximation with $\forall c\geq 1$.

► Remark. It is easy to see that when $k=1$, the geometric prototype is simply the mean point. Actually, the problem of geometric prototype can be viewed as a "chromatic k-means clustering". The kn points of $\cup_{i=1}^{n}P_i$ form k clusters where the k points of each P_i should be assigned to the k clusters separately; to minimize the objective function (2), the k points of $g(\mathbb{P})$ should be the mean points of the resulting clusters.

In Definition 2, the dimension d could be either constant or high depending on the applications, and n usually is large (k could be not constant, but often much smaller than n in the applications). To our best knowledge, the general geometric prototype problem has never been systematically studied in the area of computational geometry (except some special cases; see Section 1.1), but finds many real-world applications recently. Below, we introduce two important applications in low and high dimension, respectively.

(1) Wasserstein Barycenter. Given a large set of images, finding their average yields several benefits in practice. For example, if all the images are taken from the same object but have certain extents of noise, their average image could serve as a robust pattern to represent them; also, this is an efficient way to compress large image datasets. In computer vision, Earth Mover's Distance (EMD) [39] is widely used to measure the difference between two images; the average image minimizing the total EMDs to all the images is defined as the *Wasserstein Barycenter* [9,10,17,29,46]. In addition, Ding and Xu [20,24] considered the case allowing rigid/affine transformations for each image. Wasserstein Barycenter can also be applied to Bayesian inference [43]. Note that the geometric prototype defined above is not exactly equivalent to Wasserstein Barycenter, because the latter one requires each point having a non-negative weight and EMD is to minimize the max flow cost; however, the techniques proposed in this paper can be easily extended to handle EMD and we will discuss it later.

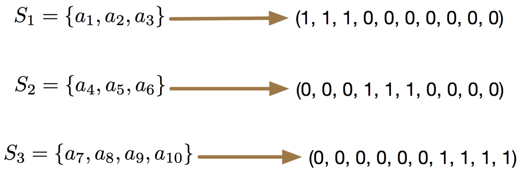

Figure 1 $d = 10$ and $k = 3$. The three clusters are mapped to 3 binary vectors in $\mathbb{R}^{10}$.

(2) Ensemble Clustering. Given a number of different clustering solutions for the same set of items, the problem about finding a unified clustering solution minimizing the total differences to them is called *ensemble clustering* [28]. This problem has attracted a great deal of attention, especially for the applications in big data and crowdsourcing [27,35,42,44]. For example, due to the proliferation of networked sensing systems, we can use a large number of sensors to record the same environment and each sensor can generate an individual clustering for the same set of objects. However, most of existing approaches rely on algebraic or graphic models and need to solve complicated optimizations with high complexities (such as semi-definite programming [42]).

Recently, Ding et al. [22] presented a novel high dimensional geometric model for the problem of ensemble clustering: suppose there are d items and each clustering solution has k clusters on these items (if less than k, we can add some dummy empty clusters); then, each single cluster is mapped to a binary vector in $\mathbb{R}^d$ where each dimension indicates the membership of an individual item (see Figure 1); so each clustering solution is mapped to a k-point set in $\mathbb{R}^d$; the size of the symmetric difference between two clusters is equal to their squared distance in $\mathbb{R}^d$, and thus the difference between two clustering solutions is always equal to half of their matching cost (Definition 1) in Euclidean space. Therefore, finding the final clustering solution minimizing the total differences to the given solutions is equivalent to computing the geometric prototype of the resulting k-point sets in $\mathbb{R}^d$. Please find more details in [22]. Note that the obtained geometric prototype may result in fractional clustering memberships, because the points of the geometric prototype are not necessarily binary vectors. So the approximation result in [22] does not violate the APX-hardness for strict ensemble/consensus clustering [11]. Actually fractional clustering memberships are acceptable and make sense in practice; for instance, we may claim that one object belongs to class 1, 2, and 3 with probabilities of 70%, 20%, and 10%, respectively.

1.1 Our Main Contributions and Related Work

Due to the non-convex nature of the geometric prototype problem, most of the aforementioned approaches for Wasserstein barycenter [9,10,17,29,46] and large-scale ensemble clustering [22] are iterative algorithms, such as alternating minimization and Alternating Direction Method of Multipliers (ADMM) [12], which can converge to some local optimums. Those approaches could be very slow for large datasets, because they may run many rounds and each round usually needs to conduct some complicated update or optimization. This is also the main motivation of our work, that is, replacing the original large input by a small core-set to speed up the computation of existing algorithms.

In this paper, our contribution is twofold in the aspects of theory and applications. In theory, we show that a small core-set can be obtained for the problem of geometric prototype. More importantly, our core-set is independent of any geometric prototype algorithm; namely, we can run any available algorithm as a black box on the core-set, instead of the original instance $\mathbb{P}$, to achieve a similar result. Although core-set has been extensively studied for many applications before [3, 38], we still need to tackle several significant challenges when constructing the core-set for geometric prototype. In practice, we test our method for solving the applications Wasserstein barycenter and ensemble clustering. The experiment shows that running the existing algorithms on core-sets can achieve almost the same results while the running times are substantially reduced.

Related work. The general geometric prototype problem has yet to be seriously considered by the theory community (to our best knowledge), however, some special cases were studied before. Based on the remark below Definition 2, we know that finding the geometric prototype is also a chromatic clustering problem. Motivated by the application of managing traffic flows, Arkin et al. [8] studied a variety of chromatic 2-center clustering in 2D and gave both exact and approximate solutions. In addition, Ding and Xu [23, 25] studied chromatic clustering in high dimension; however, their method assumes that k is constant and thus it is unable to be extended to our general geometric prototype problem.

Computing the geometric matching $\mathcal{M}(A, B)$ is a sub-problem of geometric prototype. Besides the Hungarian algorithm [16], the computational geometry community has extensively studied its approximation algorithms for the case in constant dimension [2, 4, 7, 40, 41], and some of them can achieve nearly linear running time.

The rest of the paper is organized as follows. First, we introduce some basic results and useful tools in Section 2. Then we show our core-set construction and analysis in Section 3. Finally, we implement our algorithm and test it on multiple datasets in Section 4. Due to the space limit, we omit some proofs and the reader can find more details in the full version of our paper [21].

2 Preliminaries

The hardness. Actually, we are able to show that finding the optimal geometric prototype of a given instance is NP-hard and has no FPTAS even if $k = 2$ in high dimensional space, unless P=NP. Our proof makes use of the construction by Dasgupta for the NP-hardness proof of the 2-means clustering problem in high dimension [18].

The following lemma, which can be easily obtained via Definition 1, is repeatedly used in our analysis.

▶ **Lemma 3.** *Given three k-point sets A, B, and C in $\mathbb{R}^d$,*

$$\mathcal{M}(A, B) \leq 2\mathcal{M}(A, C) + 2\mathcal{M}(C, B). \tag{3}$$

Using Markov inequality and Lemma 3, Ding et al. [22] showed that a constant approximation can be achieved with constant probability.

▶ **Theorem 4** ([22]). *Let $\alpha > 1$. Given an instance $\mathbb{P}$ of the geometric prototype problem, if we randomly pick a point set P_{i_0} from $\mathbb{P}$, then with probability at least $1 - \frac{1}{\alpha}$, $\mathcal{M}(P_{i_0}, g(\mathbb{P}))$ is no larger than $\frac{\alpha}{n}\sum_{i=1}^{n}\mathcal{M}(P_i, g(\mathbb{P}))$ and P_{i_0} yields a $(2\alpha + 2)$-approximation.*

▶ Remark. To boost the success probability, we can try multiple times and select the one yielding the lowest objective value. For example, if we try t times, the success probability will be $1 - \frac{1}{\alpha^t}$.

According to Theorem 4, the selected P_{i_0} could serve as a good initialization for the geometric prototype. To further improve the approximation ratio, the algorithm in [22] adopts a simple alternating minimization procedure, i.e., alternatively updating the prototype and matchings round by round. The main drawback of this algorithm is that it needs to repeatedly compute the matchings between the prototype and all the given point sets in each round, and thus the running time is high especially when some or all of n, k, and d are large (as discussed at the beginning of Section 1.1).

In addition, we are able to apply the well known Johnson-Lindenstrauss (JL) lemma [1] to reduce the dimensionality before running the algorithm; also, the obtained geometric prototype in the lower dimension can be efficiently mapped back to the original space [22].

▶ **Theorem 5** ([22]). *Let $0 < \epsilon < 1$ and $c \geq 1$. Suppose we randomly project a given instance $\mathbb{P}$ of the geometric prototype problem from $\mathbb{R}^d$ to $\mathbb{R}^{O(\log(nk)/\epsilon^2)}$ and obtain a new instance $\mathbb{P}'$ in the lower dimension. Then, with high probability, we can convert any c-approximation for $\mathbb{P}'$ to a $c(\frac{1+\epsilon}{1-\epsilon})^2$-approximation for $\mathbb{P}$ in $\mathbb{R}^d$, in $O(nkd)$ time.*

The following lemma is a key tool in our analysis. In fact, it can be viewed as an interesting supplement of Lemma 3.

▶ **Lemma 6.** *Let A, B, and C be three k-point sets in $\mathbb{R}^d$. Then for any $\epsilon > 0$,*

$$\Big|\mathcal{M}(A,B) - \mathcal{M}(A,C)\Big| \leq (1+\frac{1}{\epsilon})\mathcal{M}(B,C) + \epsilon\mathcal{M}(A,B) \tag{4}$$

3 Core-set for Reducing the Data Size

Langberg and Schulman [32] introduced a framework of core-set (it was called "ϵ-approximator" in their paper) to compress data for several geometric shape fitting problems; further, Feldman and Langberg [26] improved the core-set size for a large class of clustering problems. Here, we consider constructing a core-set of the instance $\mathbb{P}$ so as to reduce the data size and running time. Formally, our objective is to find a small sample $\mathbb{S} \subset \mathbb{P}$ and assign a weight w_l for each $P_l \in \mathbb{S}$, such that for any k-point set $Q \subset \mathbb{R}^d$,

$$\Big|\sum_{P_l\in\mathbb{P}}\mathcal{M}(P_l,Q) - \sum_{P_l\in\mathbb{S}} w_l\mathcal{M}(P_l,Q)\Big| \leq O(\epsilon)\sum_{P_l\in\mathbb{P}}\mathcal{M}(P_l,Q) \tag{5}$$

with certain probability and small enough $\epsilon > 0$. Moreover, we want to keep each weight w_l to be non-negative so as to easily run any existing algorithm or heuristic on the core-set.

Unfortunately, we cannot directly apply the existing ideas to the problem of geometric prototype, because the points from $\cup_{i=1}^n P_i$ are not independent from each other (due to the matching constraint in Definition 1; also see our remark below Definition 2) and it would be much more challenging to build the connection between the sampled core-set and $\mathbb{P}$. Instead, we regard each P_i as an "abstract point" and compute a core-set on these n abstract points. Though these abstract points can form some metric space with the matching costs being their pairwise (squared) distances, it is still quite different to metric clustering studied by [14, 26, 32], since the prototype $g(\mathbb{P})$ is not necessarily from $\mathbb{P}$ and could appear anywhere in the Euclidean space.

Conceptually, the core-set construction is a random sampling process: first, compute an upper bound on the sensitivity $\sigma_{\mathbb{P}}(P_i)$ of each P_i (we will formally define the sensitivity later); then take a sample from $\mathbb{P}$ with probabilities proportional to $\sigma_{\mathbb{P}}(P_i)$ to form the core-set. To implement this construction, we have to develop new ideas for resolving the following two issues. **(I)** How to compute $\sigma_{\mathbb{P}}(P_i)$, or its upper bound, so as to generate the probability distribution for sampling. **(II)** What sample size is needed to ensure our core-set yields a sufficient approximation. We consider these two issues in Section 3.1 and 3.2, respectively.

The final result for core-set construction of geometric prototype is presented in Theorem 13. We also discuss some extensions on other metrics (e.g., l_1 norm and earth mover's distance) and the time complexity in Section 3.3 and 3.4, respectively.

3.1 Solving Issue I

Following [32], the sensitivity of each $P_i \in \mathbb{P}$ is defined as follows:

$$\sigma_{\mathbb{P}}(P_i) = sup_Q \frac{\mathcal{M}(P_i, Q)}{\sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q)} \tag{6}$$

where Q is restricted to be k-point set in $\mathbb{R}^d$. Intuitively, the sensitivity measures the importance of each P_i among all the patterns of $\mathbb{P}$. Directly obtaining the value of $\sigma_{\mathbb{P}}(P_i)$ could be challenging and also needless, thus we often turn to compute an upper bound for it.

Recall that $g(\mathbb{P})$ is the optimal geometric prototype of $\mathbb{P}$, and let $\Delta = \sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, g(\mathbb{P}))$ for convenience.

▶ **Lemma 7.** *For any $P_i \in \mathbb{P}$, $\sigma_{\mathbb{P}}(P_i) \leq \frac{2\mathcal{M}(P_i, g(\mathbb{P}))}{\Delta} + \frac{16}{n}$.*

Proof. First, we consider $\frac{\mathcal{M}(P_i, Q)}{\sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q)}$ with a fixed Q in (6). Through Lemma 3, we know that the numerator $\mathcal{M}(P_i, Q)$ is bounded by $2\mathcal{M}(P_i, g(\mathbb{P})) + 2\mathcal{M}(g(\mathbb{P}), Q)$. Then, we consider two cases: **(1)** $\mathcal{M}(g(\mathbb{P}), Q) \leq \frac{8}{n}\Delta$ and **(2)** $\mathcal{M}(g(\mathbb{P}), Q) > \frac{8}{n}\Delta$.

Since $\Delta \leq \sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q)$, we directly have

$$\begin{aligned} \frac{\mathcal{M}(P_i, Q)}{\sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q)} &\leq \frac{2\mathcal{M}(P_i, g(\mathbb{P})) + 2\mathcal{M}(g(\mathbb{P}), Q)}{\sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q)} \\ &\leq \frac{2\mathcal{M}(P_i, g(\mathbb{P})) + \frac{16}{n}\Delta}{\Delta} \\ &= \frac{2\mathcal{M}(P_i, g(\mathbb{P}))}{\Delta} + \frac{16}{n} \end{aligned} \tag{7}$$

for case (1).

Now, we assume that case (2) is true. Denote by $\mathbb{P}'$ the set $\{P_l \in \mathbb{P} \mid \mathcal{M}(P_l, g(\mathbb{P})) \leq \frac{2}{n}\Delta\}$, and Markov inequality implies $|\mathbb{P}'| \geq \frac{n}{2}$. Applying Lemma 3 again, we have

$$\begin{aligned} \sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q) &\geq \sum_{P_l \in \mathbb{P}'} \mathcal{M}(P_l, Q) \geq \sum_{P_l \in \mathbb{P}'} \Big(\frac{1}{2}\mathcal{M}(g(\mathbb{P}), Q) - \mathcal{M}(g(\mathbb{P}), P_l)\Big) \\ &\geq \sum_{P_l \in \mathbb{P}'} \Big(\frac{1}{2}\mathcal{M}(g(\mathbb{P}), Q) - \frac{2}{n}\Delta\Big) \geq \frac{n}{2}\Big(\frac{1}{2}\mathcal{M}(g(\mathbb{P}), Q) - \frac{2}{n}\Delta\Big) \\ &= \frac{n}{4}\mathcal{M}(g(\mathbb{P}), Q) - \Delta. \end{aligned} \tag{8}$$

As a consequence,

$$\frac{\mathcal{M}(P_i, Q)}{\sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q)} \leq \frac{2\mathcal{M}(P_i, g(\mathbb{P})) + 2\mathcal{M}(g(\mathbb{P}), Q)}{\frac{n}{4}\mathcal{M}(g(\mathbb{P}), Q) - \Delta}. \tag{9}$$

Since both $\mathcal{M}(P_i, g(\mathbb{P}))$ and Δ are independent of Q, the right-hand side of (9) can be viewed as a function on $\mathcal{M}(g(\mathbb{P}), Q)$. Through a simple calculation and the assumption of case (2) (i.e., $\mathcal{M}(g(\mathbb{P}), Q) > \frac{8}{n}\Delta$), we know that it is always less than $\frac{2\mathcal{M}(P_i, g(\mathbb{P}))}{\Delta} + \frac{16}{n}$.

Overall, we have $\sigma_{\mathbb{P}}(P_i) \leq \frac{2\mathcal{M}(P_i, g(\mathbb{P}))}{\Delta} + \frac{16}{n}$ for both cases. ◀

However, only Lemma 7 is not enough to compute the upper bound for $\sigma_{\mathbb{P}}(P_i)$, because neither $\mathcal{M}(P_i, g(\mathbb{P}))$ nor Δ is known. Therefore, we need to compute an approximation to replace the upper bound given by Lemma 7.

▶ **Lemma 8.** *Suppose P_{i_0} is randomly picked from $\mathbb{P}$, and let $\tilde{\Delta} = \sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, P_{i_0})$ and $\alpha > 1$. Then with probability $1 - \frac{1}{\alpha}$, for all $1 \leq i \leq n$, $\sigma_{\mathbb{P}}(P_i) \leq 8(\alpha+1)\frac{\mathcal{M}(P_i, P_{i_0})}{\tilde{\Delta}} + \frac{4\alpha+16}{n}$.*

Proof. According to Theorem 4, we know that $\mathcal{M}(P_{i_0}, g(\mathbb{P})) \leq \frac{\alpha}{n}\Delta$ and $\tilde{\Delta} \leq 2(\alpha+1)\Delta$ with probability at least $1 - \frac{1}{\alpha}$. Then we have

$$\begin{aligned} \sigma_{\mathbb{P}}(P_i) &\leq \frac{2\mathcal{M}(P_i, g(\mathbb{P}))}{\Delta} + \frac{16}{n} \leq \frac{4\mathcal{M}(P_i, P_{i_0}) + 4\mathcal{M}(P_{i_0}, g(\mathbb{P}))}{\Delta} + \frac{16}{n} \\ &\leq \frac{4\mathcal{M}(P_i, P_{i_0})}{\frac{1}{2(\alpha+1)}\tilde{\Delta}} + \frac{4\mathcal{M}(P_{i_0}, g(\mathbb{P}))}{\Delta} + \frac{16}{n} \\ &\leq 8(\alpha+1)\frac{\mathcal{M}(P_i, P_{i_0})}{\tilde{\Delta}} + \frac{4\alpha+16}{n}, \end{aligned} \tag{10}$$

where the first inequality comes from Lemma 7. So the proof is completed. ◀

Lemma 8 indicates that once P_{i_0} is selected, we can obtain an upper bound for each $\sigma_{\mathbb{P}}(P_i)$ by computing the values $\mathcal{M}(P_i, P_{i_0})$ and $\tilde{\Delta}$.

3.2 Solving Issue II

Let $t_{\mathbb{P}}(P_i)$ and T denote the obtained upper bound of $\sigma_{\mathbb{P}}(P_i)$ from Lemma 8 and their sum, respectively. It is easy to know that $T = \sum_{P_i \in \mathbb{P}} t_{\mathbb{P}}(P_i) \leq 8(\alpha+1) + 4\alpha + 16$ which is constant if α is constant. For the sake of simplicity, we always assume $T = O(1)$ in our analysis below.

We have the following theorem from [32,45] (we slightly modify their statements to fit our problem better).

▶ **Theorem 9** ([32,45]). *Let Q be any fixed k-point set in $\mathbb{R}^d$. **i.** If we take a sample P_i from $\mathbb{P}$ according to the distribution $\frac{t_{\mathbb{P}}(P_i)}{T}$, the expectation of $\frac{T}{t_{\mathbb{P}}(P_i)}\mathcal{M}(P_i, Q)$ is $\sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q)$. **ii.** If we take a sample $\mathbb{S}$ of size of r from $\mathbb{P}$ according to the same distribution, and let $\epsilon > 0$,*

$$Pr\Big[\big|\sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q) - \frac{1}{r}\sum_{P_l \in \mathbb{S}} \frac{T}{t_{\mathbb{P}}(P_l)}\mathcal{M}(P_l, Q)\big| \leq \epsilon \sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q)\Big] \geq 1 - 2e^{-\frac{2r\epsilon^2}{T^2}}. \tag{11}$$

In particular, (11) is an application of Hoeffding's inequality because each $\frac{T}{t_{\mathbb{P}}(P_i)}\mathcal{M}(P_i, Q)$ is a random variable between 0 and $T\sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q)$ (see Lemma 2.2 of [45] for more details). Moreover, (11) shows that the sample $\mathbb{S}$ together with the weight $w_l = \frac{1}{r}\frac{T}{t_{\mathbb{P}}(P_l)}$ for each $P_l \in \mathbb{S}$ will form a core-set of $\mathbb{P}$ with respect to the fixed Q (see (5)). But (5) should hold for an infinite number of possible candidates for the geometric prototype, rather than one single Q, in the space. Hence, we need to determine an appropriate sample size (i.e., issue **(II)**).

Our basic idea is to discretize the space and generate a finite number of representations for them; then we can take a union bound for the final success probability through (11). Note [45] also used discretization to determine the sample size for projective clustering integer points; but our idea and analysis are quite different due to the different natures of the problems. Also, [26,32] defined the "dimension" of the clustering problems so as to bound their sample sizes. Here, we avoid using their approach due to two reasons: first, it will be very complicated to define and compute the dimension of the geometric prototype problem; second, the framework in [26] would result in a more complicated sampling process and even may cause negative weights, however, we prefer to keep our sampling process simple as described in Theorem 9 (especially when using any available algorithm or heuristic as a black box on the core-set). We elaborate on our analysis below.

By Theorem 4, we assume that a randomly picked P_{i_0} yields a $(2\alpha+2)$-approximation and denote by L the resulting cost $\sum_{P_l\in\mathbb{P}}\mathcal{M}(P_l, P_{i_0})$. The following lemma reveals that we just need to consider the k-point sets which are not too far from P_{i_0}.

▶ **Lemma 10.** *For any k-point set Q with $\mathcal{M}(Q, P_{i_0}) > \frac{4L}{n}$, the resulting cost $\sum_{P_l\in\mathbb{P}}\mathcal{M}(P_l, Q)$ is always higher than $\sum_{P_l\in\mathbb{P}}\mathcal{M}(P_l, P_{i_0})$.*

Proof. Using Lemma 3, we have

$$\sum_{P_l\in\mathbb{P}}\mathcal{M}(P_l, Q) \geq \sum_{P_l\in\mathbb{P}}(\frac{1}{2}\mathcal{M}(Q, P_{i_0}) - \mathcal{M}(P_l, P_{i_0})) > \frac{1}{2}4L - L = \sum_{P_l\in\mathbb{P}}\mathcal{M}(P_l, P_{i_0}). \tag{12}$$

So the proof is completed. ◀

Because we already have the initial solution P_{i_0}, we are only interested in the solutions having lower costs. Thus, we focus on the k-point set Qs with $\mathcal{M}(Q, P_{i_0}) \leq \frac{4L}{n}$ based on Lemma 10. Let $Q = \{q_1, q_2, \cdots, q_k\} \subset \mathbb{R}^d$ and $R = L/n$. W.l.o.g, we assume the induced permutation of $\mathcal{M}(Q, P_{i_0})$ in Definition 1 is $\pi(j) = j$ for $1 \leq j \leq k$. The constraint $\mathcal{M}(Q, P_{i_0}) \leq \frac{4L}{n}$ directly implies that $||q_j - p^{i_0}_j|| \leq 2\sqrt{R}$ for each $1 \leq j \leq k$. We use $B(x, \rho)$ to denote the ball centered at the point x with the radius ρ. Then we draw k balls $B(p^{i_0}_j, 2\sqrt{R})$ for each $1 \leq j \leq k$; inside each ball, we build a uniform grid G_j with the grid side length $\epsilon\sqrt{\frac{R}{kd}}$. Let Γ be the Cartesian product $G_1 \times G_2 \times \cdots \times G_k$. It is easy to know that Γ contains $O\Big((\frac{4\sqrt{kd}}{\epsilon})^{kd}\Big)$ k-point sets in total. Therefore, we can apply (11) of Theorem 9 to obtain a union bound over all the k-point sets of Γ (recall $T = O(1)$).

▶ **Lemma 11.** *If the sample $\mathbb{S}$ in Theorem 9 has the size of $O(\frac{kd}{\epsilon^2}\log\frac{kd}{\epsilon})$, and each $P_l \in \mathbb{S}$ has the weight $w_l = \frac{1}{r}\frac{T}{t_{\mathbb{P}}(P_l)}$, then with constant probability the inequality (5) holds for each $Q \in \Gamma$.*

Next we consider the k-point set $Q = \{q_1, q_2, \cdots, q_k\} \notin \Gamma$. Again, w.l.o.g, we assume the induced permutation of $\mathcal{M}(P_{i_0}, Q)$ is $\pi(j) = j$ for $1 \leq j \leq k$. Also, due to our above assumption, we know that each q_j is covered by the ball $B(p^{i_0}_j, 2\sqrt{R})$. To help our analysis, we take its "nearest neighbor" from Γ, $\mathcal{N}(Q) = \{\mathcal{N}(q_1), \mathcal{N}(q_2), \cdots, \mathcal{N}(q_k)\}$ with each $\mathcal{N}(q_j)$ being the nearest grid point of q_j in G_j. So we have

$$||q_j - \mathcal{N}(q_j)|| \leq \epsilon\sqrt{\frac{R}{k}} \quad for \quad 1 \leq j \leq k. \tag{13}$$

For the sake of convenience, let $X_1 = \Big|\sum_{P_l\in\mathbb{P}}\mathcal{M}(P_l, Q) - \sum_{P_l\in\mathbb{P}}\mathcal{M}(P_l, \mathcal{N}(Q))\Big|$, $X_2 = \Big|\sum_{P_l\in\mathbb{P}}\mathcal{M}(P_l, \mathcal{N}(Q)) - \frac{1}{r}\sum_{P_l\in\mathbb{S}}\frac{T}{t_{\mathbb{P}}(P_l)}\mathcal{M}(P_l, \mathcal{N}(Q))\Big|$, $X_3 = \Big|\frac{1}{r}\sum_{P_l\in\mathbb{S}}\frac{T}{t_{\mathbb{P}}(P_l)}\mathcal{M}(P_l, \mathcal{N}(Q)) - \frac{1}{r}\sum_{P_l\in\mathbb{S}}\frac{T}{t_{\mathbb{P}}(P_l)}\mathcal{M}(P_l, Q)\Big|$. It is easy to see

$$\Big|\sum_{P_l\in\mathbb{P}}\mathcal{M}(P_l, Q) - \frac{1}{r}\sum_{P_l\in\mathbb{S}}\frac{T}{t_{\mathbb{P}}(P_l)}\mathcal{M}(P_l, Q)\Big| \leq X_1 + X_2 + X_3 \tag{14}$$

where X_2 is bounded by Lemma 11. So the remaining issue is to prove that the other two items X_1 and X_3 in (14) are small as well. That is, Lemma 11 can be extended from $\mathcal{N}(Q)$ to Q.

Note $X_1 \leq \sum_{P_l \in \mathbb{P}} \big|\mathcal{M}(P_l, Q) - \mathcal{M}(P_l, \mathcal{N}(Q))\big|$, so we consider each item $\big|\mathcal{M}(P_l, Q) - \mathcal{M}(P_l, \mathcal{N}(Q))\big|$ separately. Using Lemma 6, we have

$$\big|\mathcal{M}(P_l, Q) - \mathcal{M}(P_l, \mathcal{N}(Q))\big| \leq (1 + \frac{1}{\epsilon})\mathcal{M}(Q, \mathcal{N}(Q)) + \epsilon\mathcal{M}(P_l, Q). \tag{15}$$

In addition, we have $\mathcal{M}(Q, \mathcal{N}(Q)) \leq k\Big(\epsilon\sqrt{\frac{R}{k}}\Big)^2 = \epsilon^2 R$ by (13). Therefore, we have

$$\begin{aligned}
\Big|\sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q) - \sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, \mathcal{N}(Q))\Big| &\leq \sum_{P_l \in \mathbb{P}} \big|\mathcal{M}(P_l, Q) - \mathcal{M}(P_l, \mathcal{N}(Q))\big| \\
&\leq (1 + \frac{1}{\epsilon})n\mathcal{M}(Q, \mathcal{N}(Q)) + \epsilon \sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q) \\
&\leq O(\epsilon)nR + \epsilon \sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q) \\
&= O(\epsilon) \sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q), \qquad (16)
\end{aligned}$$

where the last equality comes from $nR = L$ which is a constant approximation of the optimal objective value. (16) also implies that

$$\big(1 - O(\epsilon)\big) \sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q) \leq \sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, \mathcal{N}(Q)) \leq \big(1 + O(\epsilon)\big) \sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, Q). \tag{17}$$

Next, we consider the last item X_3 in (14). It is a little more complicated because the coefficient $\frac{T}{t_{\mathbb{P}}(P_l)}$ could be large. We need the following lemma first.

▶ **Lemma 12.** *For each $P_l \in \mathbb{P}$, $t_{\mathbb{P}}(P_l) > \frac{1}{4n}$.*

Proof. Fix one $P_l \in \mathbb{P}$. We select $P_{l'}$ that has the largest matching cost to P_l, i.e., $\mathcal{M}(P_l, P_{l'}) = \max_{P_i \in \mathbb{P}} \mathcal{M}(P_l, P_i)$, and set $Q = P_{l'}$. Using Lemma 3, we have $\mathcal{M}(P_i, Q) \leq 2\mathcal{M}(P_i, P_l) + 2\mathcal{M}(P_l, Q) \leq 4\mathcal{M}(P_l, Q)$ for any $1 \leq i \leq n$. Therefore, based on the fact that $t_{\mathbb{P}}(P_l)$ is the upper bound of $\sigma_{\mathbb{P}}(P_l)$ in (6), we know that it should be at least $\frac{\mathcal{M}(P_l, Q)}{(1+4(n-1))\mathcal{M}(P_l, Q)} > \frac{1}{4n}$. ◀

Using Lemma 12 and the same idea for (16), we have

$$\begin{aligned}
&\Big|\frac{1}{r} \sum_{P_l \in \mathbb{S}} \frac{T}{t_{\mathbb{P}}(P_l)} \mathcal{M}(P_l, \mathcal{N}(Q)) - \frac{1}{r} \sum_{P_l \in \mathbb{S}} \frac{T}{t_{\mathbb{P}}(P_l)} \mathcal{M}(P_l, Q)\Big| \\
\leq\ & \frac{1}{r} \sum_{P_l \in \mathbb{S}} \frac{T}{t_{\mathbb{P}}(P_l)} \big|\mathcal{M}(P_l, \mathcal{N}(Q)) - \mathcal{M}(P_l, Q)\big| \\
\leq\ & \frac{1}{r} \sum_{P_l \in \mathbb{S}} \frac{T}{t_{\mathbb{P}}(P_l)} \Big((1 + \frac{1}{\epsilon})\mathcal{M}(\mathcal{N}(Q), Q) + \epsilon\mathcal{M}(P_l, \mathcal{N}(Q))\Big) \\
\leq\ & \max_{P_l \in \mathbb{S}}\{\frac{T}{t_{\mathbb{P}}(P_l)}\} \cdot (1 + \frac{1}{\epsilon})\mathcal{M}(\mathcal{N}(Q), Q) + \epsilon\frac{1}{r} \sum_{P_l \in \mathbb{S}} \frac{T}{t_{\mathbb{P}}(P_l)} \mathcal{M}(P_l, \mathcal{N}(Q)) \\
\leq\ & O(\epsilon)nR + \epsilon\frac{1}{r} \sum_{P_l \in \mathbb{S}} \frac{T}{t_{\mathbb{P}}(P_l)} \mathcal{M}(P_l, \mathcal{N}(Q)), \qquad (18)
\end{aligned}$$

where the last inequality comes from Lemma 12 and $T = O(1)$. In addition, Lemma 11 guarantees that $\epsilon\frac{1}{r} \sum_{P_l \in \mathbb{S}} \frac{T}{t_{\mathbb{P}}(P_l)} \mathcal{M}(P_l, \mathcal{N}(Q)) = O(\epsilon) \sum_{P_l \in \mathbb{P}} \mathcal{M}(P_l, \mathcal{N}(Q))$. Applying the triangle inequality (14) with the bounds (16), (17) and (18), we have

$$\begin{aligned}
& \Big|\sum_{P_l\in\mathbb{P}}\mathcal{M}(P_l,Q)-\frac{1}{r}\sum_{P_l\in\mathbb{S}}\frac{T}{t_{\mathbb{P}}(P_l)}\mathcal{M}(P_l,Q)\Big| \\
\leq\ & O(\epsilon)\sum_{P_l\in\mathbb{P}}\mathcal{M}(P_l,Q)+O(\epsilon)\sum_{P_l\in\mathbb{P}}\mathcal{M}(P_l,\mathcal{N}(Q)) \\
=\ & O(\epsilon)\sum_{P_l\in\mathbb{P}}\mathcal{M}(P_l,Q). \qquad (19)
\end{aligned}$$

Consequently, we have the final theorem for core-set.

▶ **Theorem 13.** *Let P_{i_0} be the k-point set randomly selected by Theorem 4, and $\mathbb{S}$ be the sample from $\mathbb{P}$ according to the distribution $\frac{t_{\mathbb{P}}(P_i)}{T}$. If the sample $\mathbb{S}$ has the size of $r=O(\frac{kd}{\epsilon^2}\log\frac{kd}{\epsilon})$ and each $P_l\in\mathbb{S}$ has the weight $w_l=\frac{1}{r}\frac{T}{t_{\mathbb{P}}(P_l)}$, then with constant probability the inequality (5) holds for any k-point set $Q\subset\mathbb{R}^d$ with $\mathcal{M}(Q,P_{i_0})\leq\frac{4L}{n}$.*

Recall that Theorem 5 tells us that the dimension can be reduced by Johnson-Lindenstrauss (JL)-transform. Thus, we directly have the following corollary.

▶ **Corollary 14.** *Given a high dimensional instance $\mathbb{P}$ (e.g., $d\gg\log n,\log k$), we can obtain a sample $\mathbb{S}$ having the size of $\tilde{O}(\frac{k}{\epsilon^4})$, where with constant probability the inequality (5) holds for any k-point set $Q\subset\mathbb{R}^d$ with $\mathcal{M}(Q,P_{i_0})\leq\frac{4L}{n}$. $\tilde{O}(\cdot)$ ignores the logarithmic factors $\log n$ and $\log k$.*

3.3 Some Extensions

Here, we briefly introduce some extensions of our core-set construction on other metrics.
(1). Our core-set construction can be extended to l_1 norm, i.e., the squared distances are replaced by absolute distances in the matching cost (1). Actually, the analysis for l_1 norm is even easier than that for l_2 norm, since we can directly use the triangle inequality rather than Lemma 3 or Lemma 6 when solving the aforementioned two issues, bounding the sensitivities and discretizing the space of candidates for geometric prototype.

A remaining issue for future work is that the dimension reduction result of Theorem 5 is not applicable to l_1 norm, due to the fact that it is much harder to compute geometric median (Fermat-Weber point) than mean point [15]. Fortunately, the high dimensional application, ensemble clustering, mentioned in Section 1 only uses l_2 norm, because the symmetric difference between two clusters corresponds to their squared distance in the space.
(2). We can also consider the case with weighted point sets for both l_1 and l_2 norm, i.e., each point of P_i has a non-negative weight. To make the problem meaningful in practice, we require that each P_i and the desired geometric prototype have the same total weight $W>0$; we can further assume W and all the weights are integers by scaling and rounding in practice. Thus, the computation on the matching between two point sets becomes the problem of earth mover's distance (EMD) [39]. Fortunately, the triangle inequality still holds for EMD because we assume they have equal total weight; as a consequence, we can bound the sensitivities for issue **(I)**. For issue **(II)**, we still discretize the space and build the set of k-point sets Γ with the same cardinality of the unweighted case; the only difference is that we need to consider the total $O(W^k)$ possible distributions of the total weight W over the k points of each k-point set, which increases the size of the core-set with an extra $O(\frac{k\log W}{\epsilon^2})$.

3.4 The Time Complexity

Suppose the complexity of computing $\mathcal{M}(A, B)$ is $h(k, d)$, then the running time for computing the core-set is simply $O(h(k, d) \cdot n)$ because we just need to compute each $\mathcal{M}(P_i, P_{i_0})$ so as to obtain the sensitivities for sampling (see Lemma 8). For simplicity, we can just use the Hungarian algorithm [16] so that $h(k, d) = O(k^2 d + k^3)$, where the term $k^2 d$ is for building the bipartite graph. In fact, this can be further improved by our following two observations. First, we just need to know the matching costs, rather than the matchings, for computing the upper bounds of the sensitivities in Lemma 8. Second, it is not necessary to always have the optimal matching costs. For example, if we compute a value $\mathcal{M}'(P_i, P_{i_0})$ for each $\mathcal{M}(P_i, P_{i_0})$ instead, such that $\mathcal{M}(P_i, P_{i_0}) \leq \mathcal{M}'(P_i, P_{i_0}) \leq c\mathcal{M}(P_i, P_{i_0})$ with some constant $c \geq 1$, the resulting T and each $t_{\mathbb{P}}(P_i)$ will increase by some appropriate constant factors correspondingly; in other words, the sample size in Theorem 13 will increase by only a constant factor. Some algorithms [13, 30, 31] are designed for approximately estimating the matching cost, and their running times can be nearly linear if the dimension d is constant; in practical fields, several heuristic algorithms [37] are also proposed for this purpose.

For the high dimensional case, we can apply JL-transform in advance, to reduce the dimensionality to be $O(\log(nk)/\epsilon^2)$ (Theorem 5 and Corollary 14). A naive implementation of the JL-transform by matrix multiplication has the complexity $O\big(\frac{1}{\epsilon^2} nkd \log(nk)\big)$ [19], and several even faster and practical algorithms have been studied before [1, 6, 34].

4 Experiments

To show the advantage of using core-sets for the problem of geometric prototype, we study the two important applications introduced in Section 1, Wasserstein barycenter and ensemble clustering. For each application, we run the existing algorithm on the original dataset and core-sets with different size levels. In general, our experiments suggest that running the algorithm on a small core-set can achieve very close performance and greatly reduce the running time. All of the experimental results were obtained on a Windows workstation with 2.4GHz Intel Xeon E5-2630 v3 CPU and 32GB DDR4 2133MHz Memory; the algorithms are implemented in Matlab R2016b.

Wasserstein barycenter. MNIST [33] is a popular benchmark dataset of handwritten digits from 0 to 9. For each digit, we generate a set of 3000 28×28 grayscale images including 10% noise (i.e., 300 images randomly selected from the other 9 digits). First, we represent the 28×28 pixels by 60 weighted $2D$ points via k-means clustering [36]: group the pixels into 60 clusters and each cluster is represented by its cluster center; each center has the weight equal to the total pixel values of the cluster. Therefore the problem of Wasserstein barycenter becomes an instance of geometric prototype with $n = 3000$, $k = 60$, and $d = 2$.

Ensemble clustering. To construct an instance of ensemble clustering, we generate a synthetic dataset of 2000 points randomly sampled from $k = 50$ Gaussian distributions in $\mathbb{R}^{100}$; we apply k-means clustering 1000 times, where each time has a different initialization for the k mean points, to generate 1000 different clustering solutions. According to the model introduced by [22], each instance is a geometric prototype problem with 1000 different 50-point sets in $\mathbb{R}^{2000}$. We apply JL-transform to reduce the dimensionality from 2000 to 100, before constructing the core-set and running the algorithm; we just use the simplest random matrix multiplication to implement JL-transform [19] (actually this step takes about only 5% of the whole running time in the experiments).

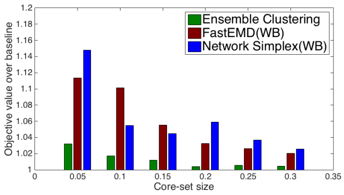

Figure 2 Normalized objective value.

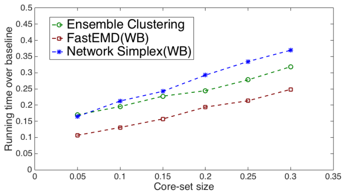

Figure 3 Normalized running time.

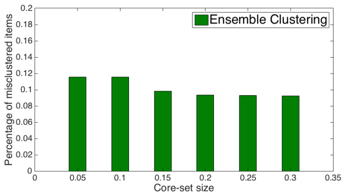

Figure 4 Percentage of misclustered items.

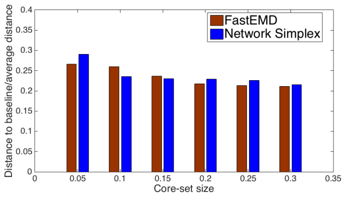

Figure 5 Matching cost to ground truth.

For both applications, we construct the core-sets using the method in Section 3; we vary the core-set size from 5% to 30% of the input size. To construct the core-set, we need to compute the matching cost $\mathcal{M}(P_i, P_{i_0})$ as discussed in Section 3.4: for the high dimensional application (i.e., ensemble clustering), we just use the Hungarian algorithm [16]; for the low dimensional application (i.e., Wasserstein barycenter), we use two existing popular algorithms for computing EMD, *Network simplex algorithm* [5] and the heuristic but faster algorithm *FastEMD* [37]. As the black box for computing the geometric prototype, we use the alternating minimization approach [22]. For each application, we consider three criteria: running time, objective value (in Definition 2), and difference to ground truth. For ensemble clustering, we compute the percentage of misclustered items of the obtained prototype as the difference to ground truth. For Wasserstein barycenter, since it is difficult to determine a unique ground truth for each handwritten digit, we directly use the prototype obtained from the original input dataset as the ground truth; then we compute its matching cost to the prototype obtained from core-set, denoted by x, as well as the average matching cost over the input images to the ground truth, denoted by Ave; finally, we obtain the ratio x/Ave. In general, the lower the ratio x/Ave, the closer the obtained prototype to the ground truth (comparing with the input images).

Results. For each application, we run 50 trials and report the average results. Figure 2 shows the obtained normalized objective values over the base line (i.e., the objective value obtained on the original input dataset), which are all lower than 1.2; that means our core-sets are good approximations for the original data. More importantly, the running times are significantly reduced in Figure 3, e.g., for the core-set having 5% of the input data size, the algorithm (containing the core-sets construction) only runs within 10%-17% of the original time. In addition, our obtained prototypes are very close to the corresponding ground truths, even for the core-set at the level 5%. Figure 4 provides the percentages of misclustered items for ensemble clustering, which are around 8%-12%. Figure 5 shows the values of x/Ave, which are around 0.25. For Wasserstein barycenter, we can see the Network simplex algorithm and FastEMD algorithm achieve very similar qualities, but FastEMD only takes about 60% of the running time of the Network simplex algorithm.

5 Future Work

Following our work, several interesting problems for geometric prototype deserve to be explored. For example, is there any algorithm achieving a better approximation ratio than Theorem 4? In addition, we leave the hardness for the low dimensional case of geometric prototype as an open problem in future work.

References

1 Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4):671–687, 2003.

2 Pankaj K. Agarwal, Kyle Fox, Debmalya Panigrahi, Kasturi R. Varadarajan, and Allen Xiao. Faster algorithms for the geometric transportation problem. In *33rd International Symposium on Computational Geometry, SoCG 2017, July 4-7, 2017, Brisbane, Australia*, pages 7:1–7:16, 2017.

3 Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Geometric approximation via coresets. *Combinatorial and Computational Geometry*, 52:1–30, 2005.

4 Pankaj K. Agarwal and Kasturi R. Varadarajan. A near-linear constant-factor approximation for euclidean bipartite matching? In *Proceedings of the 20th ACM Symposium on Computational Geometry, Brooklyn, New York, USA, June 8-11, 2004*, pages 247–252, 2004.

5 Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, 1993.

6 Nir Ailon and Bernard Chazelle. The fast Johnson—Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on computing*, 39(1):302–322, 2009.

7 Alexandr Andoni, Aleksandar Nikolov, Krzysztof Onak, and Grigory Yaroslavtsev. Parallel algorithms for geometric graph problems. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 574–583, 2014.

8 Esther M Arkin, José Miguel Díaz-Báñez, Ferran Hurtado, Piyush Kumar, Joseph S.B. Mitchell, Belén Palop, Pablo Pérez-Lantero, Maria Saumell, and Rodrigo I Silveira. Bichromatic 2-center of pairs of points. *Computational Geometry*, 48(2):94–107, 2015.

9 Marcus Baum, Peter Willett, and Uwe D. Hanebeck. On wasserstein barycenters and MMOSPA estimation. *IEEE Signal Process. Lett.*, 22(10):1511–1515, 2015.

10 Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

11 Paola Bonizzoni, Gianluca Della Vedova, Riccardo Dondi, and Tao Jiang. On the approximation of correlation clustering and consensus clustering. *Journal of Computer and System Sciences*, 74(5):671–696, 2008.

12 Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

13 Sergio Cabello, Panos Giannopoulos, Christian Knauer, and Günter Rote. Matching point sets with respect to the earth mover's distance. *Computational Geometry*, 39(2):118–133, 2008.

14 Ke Chen. On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009.

15 Michael B Cohen, Yin Tat Lee, Gary Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 9–21. ACM, 2016.

16 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition.* The MIT Press, 3rd edition, 2009.

17 Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In *International Conference on Machine Learning*, pages 685–693, 2014.

18 Sanjoy Dasgupta. The hardness of k-means clustering. *Technical Report*, 2008.

19 Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

20 Hu Ding, Ronald Berezney, and Jinhui Xu. k-prototype learning for 3d rigid structures. In *Advances in Neural Information Processing Systems*, pages 2589–2597, 2013.

21 Hu Ding and Manni Liu. On geometric prototype and applications. *CoRR*, abs/1804.09655, 2018. arXiv:1804.09655.

22 Hu Ding, Lu Su, and Jinhui Xu. Towards distributed ensemble clustering for networked sensing systems: a novel geometric approach. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2016, Paderborn, Germany, July 4-8, 2016*, pages 1–10, 2016.

23 Hu Ding and Jinhui Xu. Solving the chromatic cone clustering problem via minimum spanning sphere. In *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 773–784, 2011.

24 Hu Ding and Jinhui Xu. Finding median point-set using earth mover's distance. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

25 Hu Ding and Jinhui Xu. A unified framework for clustering constrained data without locality property. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1471–1490, 2015.

26 Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 569–578, 2011.

27 Jing Gao, Feng Liang, Wei Fan, Yizhou Sun, and Jiawei Han. Graph-based consensus maximization among multiple supervised and unsupervised models. In *Advances in Neural Information Processing Systems*, pages 585–593, 2009.

28 Joydeep Ghosh and Ayan Acharya. Cluster ensembles: Theory and applications. In *Data Clustering: Algorithms and Applications*, pages 551–570. CRC, 2013.

29 Alexandre Gramfort, Gabriel Peyré, and Marco Cuturi. Fast optimal transport averaging of neuroimaging data. In *International Conference on Information Processing in Medical Imaging*, pages 261–272. Springer, 2015.

30 Piotr Indyk. A near linear time constant factor approximation for euclidean bichromatic matching (cost). In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 39–42. Society for Industrial and Applied Mathematics, 2007.

31 Piotr Indyk and Nitin Thaper. Fast color image retrieval via embeddings. In *Workshop on Statistical and Computational Theories of Vision (at ICCV)*, 2003.

32 Michael Langberg and Leonard J Schulman. Universal ε-approximators for integrals. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 598–607. SIAM, 2010.

33 Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

34 Edo Liberty and Steven W Zucker. The mailman algorithm: A note on matrix–vector multiplication. *Information Processing Letters*, 109(3):179–182, 2009.

35 Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. Multi-view clustering via joint nonnegative matrix factorization. In *Proc. of SDM*, volume 13, pages 252–260, 2013.

36 Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

37 Ofir Pele and Michael Werman. Fast and robust earth mover's distances. In *Computer vision, 2009 IEEE 12th international conference on*, pages 460–467. IEEE, 2009.

38 Jeff M. Phillips. Coresets and sketches. *Computing Research Repository*, 2016.

39 Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

40 R. Sharathkumar and Pankaj K. Agarwal. Algorithms for the transportation problem in geometric settings. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 306–317, 2012.

41 R. Sharathkumar and Pankaj K. Agarwal. A near-linear time ϵ-approximation algorithm for geometric bipartite matching. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 385–394, 2012.

42 Vikas Singh, Lopamudra Mukherjee, Jiming Peng, and Jinhui Xu. Ensemble clustering using semidefinite programming with applications. *Machine learning*, 79(1-2):177–200, 2010.

43 Matthew Staib, Sebastian Claici, Justin Solomon, and Stefanie Jegelka. Parallel streaming wasserstein barycenters. *arXiv preprint arXiv:1705.07443*, 2017.

44 Alexander Strehl and Joydeep Ghosh. Cluster ensembles-a knowledge reuse framework for combining partitionings. In *AAAI/IAAI*, pages 93–99, 2002.

45 Kasturi R. Varadarajan and Xin Xiao. A near-linear algorithm for projective clustering integer points. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1329–1342, 2012.

46 Jianbo Ye, Panruo Wu, James Z Wang, and Jia Li. Fast discrete distribution clustering using wasserstein barycenter with sparse support. *IEEE Transactions on Signal Processing*, 65(9):2317–2332, 2017.