On k-Median Clustering in High Dimensions

Ke Chen*

Abstract

We study approximation algorithms for k-median clustering. We obtain small coresets for k-median clustering in metric spaces as well as in Euclidean spaces. Specifically, in $\mathbb{R}^d$, those coresets are of size with only *polynomial* dependency on d. This leads to a $(1+\varepsilon)$-approximation algorithm for k-median clustering in $\mathbb{R}^d$, with running time $O(ndk + 2^{(k/\varepsilon)^{O(1)}} d^2 n^\sigma)$, for any $\sigma > 0$. This is an improvement over previous results [5, 20, 21]. We also provide fast constant factor approximation algorithms for k-median clustering in finite metric spaces.

We use those coresets to compute $(1+\epsilon)$-approximation k-median clustering in the streaming model of computation, using only $O(k^2 d\epsilon^{-2}\log^8 n)$ space, where the points are taken from $\mathbb{R}^d$. This is the first streaming algorithm, for this problem, that has space complexity with only *polynomial* dependency on the dimension.

1 Introduction

Clustering is an important problem in computer science with applications in many problem domains. One of the widely studied clustering variants is the *k-median clustering*, which requires finding k centers in a set of n points that minimize the sum of distances from the data points to their nearest centers.

The first constant factor approximation algorithm for k-median clustering, in the metric space settings, was given by Charikar *et al.* [8]. There have been several improvements to the approximation ratio and the running time. We focus on the most related results here, for further information see [9] and references therein. Indyk [16] gave a randomized constant factor approximation algorithm to produce $O(k)$ centers running in $O(nk \cdot \text{polylog}(nk))$ time. Based on Indyk's construction, Guha *et al.* [12] presented a $(300 + o(1))$-approximation algorithm running in $O(nk \cdot \text{polylog}(nk))$ time. Mettu and Plaxton [22] provided a constant factor approximation algorithm that runs in $O(nk + n\log n + k^2 \log^2 n)$ time. In addition, they showed that a running time of $\Omega(nk)$ is necessary even for randomized algorithms.

In the geometric settings, one is interested in $(1+\varepsilon)$-approximation algorithms. Arora *et al.* [3] presented a $(1+\varepsilon)$-approximation algorithm for k-median clustering for points in the plane, with running time $O\big(n^{O(1/\varepsilon)+1}\big)$. Kolliopoulos and Rao [19] further improved the running time to $O(\rho n \log n \log k)$, for the discrete k-median problem, where the clustering centers can only be selected from among the input point set, and $\rho = \exp[(O(1+\log(1/\varepsilon))/\varepsilon)^{d-1}]$. Further improvement can be achieved by using coresets [1]. Informally speaking, a coreset for a clustering problem is a small (weighted) subset from the input set, such that for any set of clustering centers, the cost of clustering the coreset by the set of centers is close to the true cost (i.e., the cost of clustering the original input set by the same set of centers). In particular, Har-Peled and Mazumdar [14] used coresets to improve the running time to $O(n + \rho k^{O(1)} \log^{O(1)} n)$, using a coreset of size $O(k\varepsilon^{-d}\log n)$, for k-median clustering. Har-Peled and Kushal [13] recently showed that one can construct coresets for these problems with size independent of n. Kumar *et al.* [20, 21] showed a $(1+\varepsilon)$-approximation algorithm for k-median clustering running in $O\Big(2^{(k/\varepsilon)^{O(1)}} dn\Big)$ time.

Recently, there was growing interest in performing clustering in the streaming model of computation [12, 9, 17, 14, 10]. Here, points arrive one by one in a stream and one is interested in maintaining a clustering of the points seen so far. Typically, the input point set is too large to fit in the memory. Therefore, it is necessary to maintain a data structure to sketch the data seen so far. In this model, the complexity measure includes the overall space used and the time required to update the data structure.

In the streaming model of computation, Guha *et al.* [12] proposed an algorithm that uses $O(n^\varepsilon)$ space to compute a $2^{O(1/\varepsilon)}$-approximation to k-median clustering of points taken from a metric space. Charikar *et al.* [9] improved the result significantly by proposing a constant factor approximation algorithm using $O(k\log^2 n)$ space. In the geometric settings, Har-Peled and Mazumdar used coresets to com-

*Department of Computer Science, University of Illinois; 201 N. Goodwin, Springfield Ave.; Urbana, IL 61801; USA; kechen@uiuc.edu.

SODA '06, January 22-26, Miami, FL
 ISBN 0-89871-605-5/06/01

pute an $(1+\varepsilon)$-approximation k-median clustering using $O(k\varepsilon^{-d}\log^{2d+2} n)$ space. The algorithms handle streams with insertions only. Indyk [17] showed how to handle both insertions and deletions, under the restriction that the points are taken from a grid of finite resolution (however, the resulting clustering algorithm was prohibitively slow). Frahling and Sohler [10] extended the work of Indyk, by showing how to extract the coreset quickly and cluster it in the insertion/deletion streaming model.

Our results. We propose fast approximation algorithms for k-median clustering using coresets. We use a fast bi-criteria approximation for k-median clustering to guide a random sampling from the original input set. The sampling allows us to extract a (k,ε)-coreset (see Section 2 for formal definition) of size $O(k\varepsilon^{-2}\log n(k\log n + \log(1/\lambda)))$ in a general metric space, and a coreset of size $O(k\varepsilon^{-2}\log n(kd\log(1/\varepsilon) + k\log k + k\log\log n + \log(1/\lambda)))$ in $\mathbb{R}^d$, where the probability of success is $\geq 1-\lambda$, and λ is a prespecified parameter. The construction of this coreset is the main result of this paper.

For geometric k-median clustering in $\mathbb{R}^d$, we obtain an algorithm to find a $(1+\varepsilon)$-approximation k-median clustering in time $O(ndk + 2^{(k/\varepsilon)^{O(1)}} d^2 n^\sigma)$ that succeeds with constant probability, for any $\sigma > 0$. This result improves over the algorithm of Kumar *et al.* [20, 21], which had running time $O(2^{(k/\varepsilon)^{O(1)}} dn)$.

Our main result implies an algorithm that uses $O(k^2 d\varepsilon^{-2}\log^8 n)$ space, for $(1+\varepsilon)$-approximation k-median clustering in the streaming model, where the points are taken from $\mathbb{R}^d$. The algorithm assumes that the points arrive one by one, and removal of points is not allowed. Upon the arrival of a new point, the amortized time to update the data structure is $O(kd\cdot \mathrm{polylog}(ndk/\varepsilon))$. In comparison, previous algorithms required space and time exponential in the dimension.

For the general metric case, the coreset can also be used to stream k-median clustering using small space, such that one can compute $(1+\varepsilon)$-approximation k-median clustering using this data structure. To our knowledge, this is the first algorithm, for general metric spaces, that can use small space and can provide $(1+\varepsilon)$-approximation for clustering. Of course, since the $(1+\varepsilon)$-approximation itself can not be computed efficiently (i.e., in polynomial time), this is of limited interest. Nevertheless, it leads to a $(10+\varepsilon)$-approximation algorithm for metric k-median clustering running in $O\big(nk + k^7\varepsilon^{-4}\log^5 n\big)$ time using known techniques [4]. This result provides better trade-offs between overall running time and approximation quality over previous results when k is small. In particular, all previous algorithms with $O(nk\cdot\mathrm{polylog}(nk))$ running time [16, 12, 22] provided constant approximation, where the constant is considerably larger than the new algorithm.

The main tool we employ to extract small coresets is random sampling. We note that Mishra *et al.* [23] used similar approach to obtain fast k-median clustering algorithms. Their algorithm requires $O((M/\varepsilon)^2\log n)$ samples to approximately represent the original input point set, where n is the input size, M is the diameter of the input, and ε is the difference between the average clustering cost on the samples and the average clustering cost on the original input. Depending on the size of diameter M, their algorithm may yield running time as high as $O(n^2)$. Our approach can be interpreted as combining the approach of Mishra *et al.* with the usage of coresets and exponential grids, such that we can obtain "good" samples with size independent of M and with low dependency on the dimension.

The paper is organized as follows. In Section 3 we show an algorithm to compute small (k,ε)-coresets for metric k-median clustering. This algorithm helps illustrate our main idea. In Section 4, we prove the existence of a small (k,ε)-coreset for geometric k-median clustering. In Section 5, we propose fast approximation algorithms using those (k,ε)-coresets. We conclude in Section 6.

2 Problem definition

Let $(X,\mathbf{d})$ be a *metric space*, where $\mathbf{d}$ is the distance function defined over the points of X. Let $\mathbf{d}(v,Q) = \min_{q\in Q}\mathbf{d}(v,q)$ denote the distance between a point $v\in X$ and a set $Q\subseteq X$. Let $\mathbf{d}(V,Q) = \min_{v\in V}\mathbf{d}(v,Q)$ denote the distance between the two sets $V,Q\subseteq X$. Let $\mathrm{diam}(Q) = \max_{s,t\in Q}\mathbf{d}(s,t)$ denote the diameter of a set $Q\subseteq X$.

Let $P\subseteq X$ be a set of n points. In the following, we assume that each point $p\in P$ is associated with a positive integer weight w_p. An unweighted set P can be considered as weighted, with $w_p = 1$ for each point $p\in P$. Let $w(P) = \sum_{p\in P} w_p$ denote the total weight of P.

Definition 2.1. (Metric k-median clustering.) *Let P be a set of n points in a metric space $(X,\mathbf{d})$. A* center set $C = \{c_1,\ldots,c_k\}\subseteq X$ *induces a k clustering; that is, each point of P is assigned to its nearest center in C. The point $p\in P$ is* served by c_i *if the nearest neighbor to p in C is c_i. Let $\nu_C(p) = \mathbf{d}(p,C)\cdot w_p$ denote the cost of clustering p using C. The cost of* k-median clustering *of P by C*

is $\nu_C(P) = \sum_{p \in P} \nu_C(p)$.

The metric k-median *problem is to find a set of* k *centers* $C \subseteq P$ *that minimizes the cost* $\nu_C(P)$. *Let* $\nu_{opt}(P, k)$ *denote the cost of optimal* k*-median clustering of* P.

The average radius *of the* k*-median clustering is denoted by* $\mathbf{r}_C(P) = \nu_C(P)/w(P)$.

DEFINITION 2.2. (GEOMETRIC k-MEDIAN CLUSTERING.) *Let* P *be a set of* n *points in* $\mathbb{R}^d$, *the* geometric k-median *problem is to find a set of* k *centers* $C \subseteq \mathbb{R}^d$ *that minimizes the cost* $\nu_C(P)$, *where the distance is the usual Euclidean* L_2 *distance.*

DEFINITION 2.3. ((k, ε)-CORESET.) *Given a (weighted) point set* P *in a finite metric space, a (weighted) subset* $\mathcal{S} \subseteq P$ *is a* (k, ε)-coreset *of* P *for metric* k*-median clustering, if*

$$|\nu_C(\mathcal{S}) - \nu_C(P)| \leq \varepsilon \cdot \nu_C(P)$$

for any set $C \subseteq P$ *such that* $|C| = k$.

In the geometric settings, given a (weighted) point set P *in* $\mathbb{R}^d$, *a (weighted) set* $\mathcal{S} \subseteq P$ *is a* (k, ε)-coreset *of* P *for geometric* k*-median clustering, if*

$$|\nu_C(\mathcal{S}) - \nu_C(P)| \leq \varepsilon \cdot \nu_C(P)$$

for any set C *of* k *centers in* $\mathbb{R}^d$.

3 Metric k-median clustering

In this section, we present an algorithm to compute a (k, ε)-coreset of P for metric k-median clustering. The input are a set P of n points in a finite metric space $(X, \mathbf{d})$, and a confidence parameter λ. Our algorithm consists of two steps: (i) partition the input point set P into several disjoint subsets, and (ii) take a random sample from each subset, and return the union of the samples as the desired coreset.

3.1 The algorithm

3.1.1 Step 1: partitioning P For the sake of simplicity of exposition, we assume that the input point set P is unweighted in the remainder of the paper unless explicitly stated otherwise. Note that the results also hold when P is weighted, with slightly worse bounds on the running time and coreset size.

Assume that $\mathcal{A} \subseteq P$ is the center set of a *bi-criteria* (α, β)*-approximation* to the optimal k-median clustering of P. Namely, $\mathcal{A} = \{a_1, \ldots a_m\}$ satisfies

$$\nu_{\mathcal{A}}(P) \leq \beta \cdot \nu_{opt}(P, k),$$

where $m \leq \alpha k$. Let $R = \nu_{\mathcal{A}}(P)/(\beta n)$ be a lower bound estimate of the average radius of optimal k-median clustering.

Let P_i be the set of points of P that are served by the center a_i, for $i = 1, \ldots, m$. Let $\mathbf{b}(p, r)$ denote the close ball of radius r centered at p.

For $i = 1, \ldots, m$, let

$$P_{i,0} = P_i \cap \mathbf{b}(a_i, R)$$

and

$$P_{i,j} = P_i \cap \left(\mathbf{b}(a_i, 2^j R) \setminus \mathbf{b}(a_i, 2^{j-1} R)\right),$$

for $j = 1, \ldots, \phi$, where $\phi = \lceil \log_2(\beta n) \rceil$. We call each set $P_{i,j}$ a *ring set*, for $i = 1, \ldots, m$ and $j = 0, \ldots, \phi$.

Note that for any point in P, it must be in exactly one ring set defined above, since no point of P can be in distance $\geq \beta n R$ from all the centers of $\mathcal{A}$. Therefore, P is partitioned into disjoint ring sets $P_{i,j}$, for $i = 1, \ldots, m$ and $j = 0, \ldots, \phi$.

To obtain the set $\mathcal{A}$, we use the algorithm of Indyk [16]. It uses a black-box algorithm BB which provides a constant factor approximation k-median clustering. Such a black-box algorithm runs in roughly quadratic time. We slightly modify the algorithm of Indyk as follows. We plug the algorithm of Indyk into itself as the black-box algorithm (Namely, we recursively use the algorithm of Indyk; in the bottom of this recurrence, we still use the algorithm BB. Note that the recursion depth is a constant). This reduces the overall running time to $O(nk)$, and the algorithm returns a (α, β)-approximation with constant probability, where $\alpha = O(1), \beta = O(1)$. We boost the success probability of the algorithm to $\geq 1 - \lambda/2$ by running it $O(\log(1/\lambda))$ times, and take the solution of the smallest cost. Here λ is the prespecified confidence parameter.

REMARK 3.1. *If* P *is weighted, with total weight* W, *we use standard grouping technique to group points with roughly equal weights together, and run the above algorithm on each group with confidence parameter set to* $O(1/\log W)$. *This yields a constant factor approximation* k*-median clustering of* P *using* $O(k \log W)$ *centers with constant probability. The overall running time is* $O(nk \log \log W)$ *time.*

REMARK 3.2. *The algorithm of Jain and Vazirani [18] can serve as the black-box algorithm* BB. *Its running time is* $O(n^2 \log n(L + \log n))$, *where* 2^L *is the ratio between the largest edge length and the smallest edge length of* P. L *can be reduced to* $O(\log n)$ *by using a standard trick: First find a rough estimate of the optimal* k*-median average radius* r' *[11]; then decrease the length of every edge longer than* $10n^2 r'$ *to* $10n^2 r'$, *and increase the length of every edge shorter than* $r'/(10n^2)$ *to* $r'/(10n^2)$. *It is easy to verify, in the new problem,* $L = O(\log n)$. *Note that, a constant*

factor approximation to the new problem will still be a constant factor approximation to the original problem.

3.1.2 Step 2: random sampling Let $s = \lceil 100 \cdot \beta^2 \varepsilon^{-2}(2k \ln n + \ln(4/\lambda)) \rceil$. For each ring set $P_{i,j}$, for $i = 1, \ldots m$ and $j = 0, \ldots, \phi$, we extract a subset $\mathcal{S}_{i,j}$ as follows. If $|P_{i,j}| \leq s$, let $\mathcal{S}_{i,j} = P_{i,j}$. Otherwise, draw a set of s points from $P_{i,j}$ independently and identically, assign each point the weight $|P_{i,j}|/s$, and let $\mathcal{S}_{i,j}$ be the resulting weighted sample. For the simplicity of exposition, we assume in the reminder of the paper that the weight $|P_{i,j}|/s$ is an integer number. This is a minor technical difficulty that can be easily resolved.

Let $\mathcal{S} = \cup_{i,j} \mathcal{S}_{i,j}$. We claim that $\mathcal{S}$ is the desired (k, ε)-coreset of P.

3.2 Proof of correctness The following lemma is straightforward and we omit the easy proof.

LEMMA 3.1. *(i)* $\operatorname{diam}(P_{i,j}) \leq \beta n R$ *and* $\operatorname{diam}(P_{i,j}) \leq 2^{j+1} R$, *for* $i = 1, \ldots, m$ *and* $j = 0, \ldots, \phi$. *The same holds for* $\operatorname{diam}(\mathcal{S}_{i,j})$.

(ii) For $p \in P_{i,j}$, $2^{j-1} R < \mathbf{d}(p, \mathcal{A}) \leq 2^j R$, *for* $j = 1, \ldots, \phi$; *the same holds for* $p \in \mathcal{S}_{i,j}$. *For* $p \in P_{i,0}$, $0 \leq \mathbf{d}(p, \mathcal{A}) \leq R$; *the same holds for* $p \in \mathcal{S}_{i,0}$.

(iii) $\nu_{\mathcal{A}}(\mathcal{S}) \leq 2\nu_{\mathcal{A}}(P) + nR$.

LEMMA 3.2. (HAUSSLER [15]) *Let* $h(\cdot)$ *be a function defined on a set* P, *such that for all* $p \in P$, *we have* $0 \leq h(p) \leq M$, *where* M *is a fixed constant. Let* $S = \{p_1, \ldots, p_s\}$ *be a multiset of* s *samples drawn independently and identically from* P, *and let* $\delta > 0$ *be a parameter. If* $s \geq (M^2/2\delta^2) \cdot \ln(2/\lambda)$, *then* $\mathbf{Pr}\left[\left|\frac{h(P)}{|P|} - \frac{h(S)}{|S|}\right| \geq \delta\right] \leq \lambda$, *where* $h(S) = \sum_{s \in S} h(s)$, *and* $h(P) = \sum_{p \in P} h(p)$.

The following lemma is an application of Lemma 3.2 to k-median clustering.

LEMMA 3.3. *Let* $\xi > 0$ *and* $\lambda > 0$ *be parameters, and let* S *be a set of* $s = \lceil \xi^{-2} \ln(2/\lambda) \rceil$ *samples drawn from a point set* V *independently and identically, where* V *lies in a metric space. For a fixed set* C *of centers, we have* $|\mathbf{r}_C(V) - \mathbf{r}_C(S)| \leq \xi(\mathbf{d}(V, C) + \operatorname{diam}(V))$, *with probability* $\geq 1 - \lambda$.

Proof. Consider a point $v \in V$. By the triangle inequality we have

$$0 \leq \mathbf{d}(v, C) \leq \mathbf{d}(V, C) + \operatorname{diam}(V).$$

Consider $h(v) = \mathbf{d}(v, C)$ as a function defined over the points of V. By Lemma 3.2, setting $M = \mathbf{d}(V, C) + \operatorname{diam}(V)$, $\delta = \xi(\mathbf{d}(V, C) + \operatorname{diam}(V))$, for a sample S of size $s = \lceil \xi^{-2} \ln(2/\lambda) \rceil \geq (M^2/2\delta^2) \cdot \ln(2/\lambda)$ from V, we have

$$\begin{aligned} &\mathbf{Pr}[|\mathbf{r}_C(V) - \mathbf{r}_C(S)| \geq \xi(\mathbf{d}(V, C) + \operatorname{diam}(V))] \\ &= \mathbf{Pr}\left[\left|\frac{h(V)}{|V|} - \frac{h(S)}{|S|}\right| \geq \delta\right] \\ &\leq \lambda, \end{aligned}$$

since $\mathbf{r}_C(V) = \frac{1}{|V|} \sum_{v \in V} \mathbf{d}(v, C)$ and $\mathbf{r}_C(S) = \frac{1}{|S|} \sum_{p \in S} \mathbf{d}(p, C)$. □

THEOREM 3.1. *Given a set* P *of* n *points in a finite metric space and a parameter* $\lambda > 0$, *one can compute a weighted set* $\mathcal{S}$ *of size* $O(k\varepsilon^{-2} \log n(k \log n + \log(1/\lambda)))$, *in* $O(nk \ln(1/\lambda))$ *time. The set* $\mathcal{S}$ *is a* (k, ε)*-coreset of* P *for* k*-median clustering with probability* $\geq 1 - \lambda$.

If P *is weighted, with total weight* W, *then the running time is* $O(nk \log(1/\lambda) \log \log W)$, *and the coreset is of size* $O(k\varepsilon^{-2} \log^2 W(k \log n + \log(1/\lambda)))$.

Proof. The algorithm for computing $\mathcal{S}$ is described in Section 3.1. First observe

$$|\mathcal{S}| = O(m \cdot \phi \cdot s) = O\big(k\varepsilon^{-2} \log n(k \log n + \log(1/\lambda))\big).$$

Furthermore, the overall running time of the algorithm is dominated by the computation of the set $\mathcal{A}$, which takes $O(nk \log(1/\lambda))$ time.

Next, we show that $\mathcal{S}$ is indeed the desired coreset with probability $1 - \lambda$.

CLAIM 3.1. *Under the assumption* $\nu_{\mathcal{A}}(P) \leq \beta \cdot \nu_{\mathrm{opt}}(P, k)$, $\mathcal{S}$ *is a* (k, ε)*-coreset of* P *for* k*-median clustering, with probability* $\geq 1 - \lambda/2$.

Proof. Fix a set C of k centers. Recall that $s = \lceil 100 \cdot \beta^2 \varepsilon^{-2}(2k \ln n + \ln(4/\lambda)) \rceil$. By Lemma 3.3, we have

$$|\mathbf{r}_C(P_{i,j}) - \mathbf{r}_C(\mathcal{S}_{i,j})| \leq \frac{\varepsilon}{10\beta}(\mathbf{d}(P_{i,j}, C) + \operatorname{diam}(P_{i,j})),$$

with probability $\geq 1 - n^{-2k}\lambda/2$, for $i = 1, \ldots, m$ and $j = 0, \ldots, \phi$.

It follows

$$\begin{aligned} &|\nu_C(P_{i,j}) - \nu_C(\mathcal{S}_{i,j})| \\ &= |P_{i,j}| \cdot |\mathbf{r}_C(P_{i,j}) - \mathbf{r}_C(\mathcal{S}_{i,j})| \\ &\leq \frac{\varepsilon}{10\beta} |P_{i,j}|(\mathbf{d}(P_{i,j}, C) + \operatorname{diam}(P_{i,j})). \end{aligned}$$

In addition, we have

$$|P_{i,j}| \cdot \mathbf{d}(P_{i,j}, C) \leq \nu_C(P_{i,j}).$$

Lemma 3.1 implies

$$\begin{aligned}|P_{i,j}| \cdot \operatorname{diam}(P_{i,j}) &\leq |P_{i,j}| \cdot 2^{j+1} R \\ &\leq 4\nu_{\mathcal{A}}(P_{i,j}) + |P_{i,j}| \cdot 2R.\end{aligned}$$

Combining the above inequalities together, we have

$$\begin{aligned}&|\nu_C(P_{i,j}) - \nu_C(\mathcal{S}_{i,j})| \\ &\quad\leq \frac{\varepsilon}{10\beta}(\nu_C(P_{i,j}) + 4\nu_{\mathcal{A}}(P_{i,j}) + |P_{i,j}| \cdot 2R)\,.\end{aligned}$$

Summing this up for all sets $P_{i,j}$, for $i = 1, \ldots, m$ and $j = 0, \ldots, \phi$, we have

$$\begin{aligned}&|\nu_C(P) - \nu_C(\mathcal{S})| \\ &\quad\leq \frac{\varepsilon}{10\beta}(\nu_C(P) + 4\nu_{\mathcal{A}}(P) + 2nR) \\ &\quad\leq \frac{\varepsilon}{10\beta}(\nu_C(P) + 4\beta\nu_{opt}(P,k) + 2\nu_{opt}(P,k)) \\ &\quad\leq \varepsilon \cdot \nu_C(P),\end{aligned}$$

and this holds with probability $\geq 1 - m\phi \cdot n^{-2k}\lambda/2$. Since there are at most $\binom{n}{k}$ different ways to select a set C of k centers out of P, the set $\mathcal{S}$ is a (k,ε)-coreset of P with probability $\geq 1 - \binom{n}{k} \cdot m\phi \cdot n^{-2k}\lambda/2 \geq 1 - \lambda/2$. □

By the algorithm in Section 3.1, $\nu_{\mathcal{A}}(P) \leq \beta \cdot \nu_{\text{opt}}(P,k)$ holds with probability $\geq 1 - \lambda/2$. Therefore, by the above claim, $\mathcal{S}$ is a (k,ε)-coreset of P for k-median clustering, with probability $\geq 1 - \lambda$. □

4 Geometric k-median clustering

In this section, we compute a coreset for geometric k-median clustering. The input are a set P of n points in $\mathbb{R}^d$, and a confidence parameter λ. The algorithm follows similar steps as its metric variant. We point out the differences below.

In the partitioning step, we use the same algorithms as described in Section 3.1. It takes $O(ndk\log(1/\lambda))$ time to compute a set $\mathcal{A} = \{a_1, \ldots, a_m\}$ in $\mathbb{R}^d$ such that $\nu_{\mathcal{A}}(P) \leq \beta \cdot \nu_{opt}(P,k)$ with probability $1 - \lambda/4$, where $m = \alpha k$, $\alpha = O(1)$, and $\beta = O(1)$.

In the sampling step, let $s = \lceil c\varepsilon^{-2}(kd\ln(1/\varepsilon) + k\ln k + k\ln\ln n + \ln(1/\lambda))\rceil$, where c is an appropriate constant. As in the metric space (Section 3), we partition P into ring sets $P_{i,j}$, and extract the set $\mathcal{S}_{i,j}$ from each set $P_{i,j}$ separately, for $i = 1, \ldots, m$ and $j = 0, \ldots, \phi$, where $\phi = \lceil \log_2(\beta n(1 + 4/\varepsilon))\rceil$.

4.1 Proof of correctness The geometric k-median problem is different from the metric variant: There are infinite number of ways to select a set of k centers to serve P in $\mathbb{R}^d$, while there are only a finite number of ways to choose a set of k centers in the metric k-median problem.

To prove the correctness of the algorithm, we define an exponential grid structure. Note that the grid is used only in the analysis.

DEFINITION 4.1. *Let $\mathcal{A} = \{a_1, \ldots, a_m\}$ be a set of centers in $\mathbb{R}^d$ such that $\nu_{\mathcal{A}}(P) \leq \beta \cdot \nu_{opt}(P,k)$, where $m = \alpha k, \alpha = O(1)$, and $\beta = O(1)$.*

Let $\mathcal{U}$ denote the union of "huge" balls around the points of $\mathcal{A}$, formally $\mathcal{U} = \bigcup_{i=1}^{m} \mathbf{b}(a_i, R \cdot 2^{\phi})$, where $\mathbf{b}(p,r)$ denotes the close ball of radius r centered at p, $\phi = \lceil \log_2(\beta n(1 + 4/\varepsilon))\rceil$, and $R = \nu_{\mathcal{A}}(P)/(\beta n)$.

For $i = 1, \ldots, m$, let

$$L_{i,0} = \mathbf{b}(a_i, R)$$

and $L_{i,j} = \mathbf{b}(a_i, 2^j R) \setminus \mathbf{b}(a_i, 2^{j-1}R)$, for $j = 1, \ldots, \phi$.

We use an axis-parallel grid with side length $\varepsilon 2^j R/(24\beta\sqrt{d})$ to partition $L_{i,j}$ into cells, for $i = 1, \ldots, m$ and $j = 0, \ldots, \phi$. In each grid cell, the center of the cell is called a representative point *for the points in the cell. Note that any point inside $\mathcal{U}$ has a representative point.*

Let $\mathcal{G}$ denote the set of all representative points.

LEMMA 4.1. *For any set C' of at most k centers chosen from $\mathcal{G}$, $|\nu_{C'}(P_{i,j}) - \nu_{C'}(\mathcal{S}_{i,j})| \leq (\varepsilon/10\beta)(\nu_{C'}(P_{i,j}) + 4\nu_{\mathcal{A}}(P_{i,j}) + 2|P_{i,j}| \cdot R)$, for $i = 1, \ldots, m$ and $j = 0, \ldots, \phi$. This holds with probability $\geq 1 - \lambda/2$, where λ is the prespecified confidence parameter.*

Proof. It is easy to verify $|\mathcal{G}| = O\Big(k(\varepsilon/100)^{-d}\log n\Big)$. The lemma follows from similar arguments as used in Theorem 3.1. □

LEMMA 4.2. *For any set $C = \{c_1, \ldots, c_h\}$ of at most k points lying inside $\mathcal{U}$, there exists a set $C' = \{c'_1, \ldots, c'_h\}$ such that $C' \subseteq \mathcal{G}$, $|\nu_C(P_{i,j}) - \nu_{C'}(P_{i,j})| \leq (\varepsilon/24\beta)(2\nu_C(P_{i,j}) + 2\nu_{\mathcal{A}}(P_{i,j}) + |P_{i,j}| \cdot R)$, and $|\nu_C(\mathcal{S}_{i,j}) - \nu_{C'}(\mathcal{S}_{i,j})| \leq (\varepsilon/24\beta)(2\nu_C(\mathcal{S}_{i,j}) + 2\nu_{\mathcal{A}}(\mathcal{S}_{i,j}) + |P_{i,j}| \cdot R)$, for $i = 1, \ldots, m$ and $j = 0, \ldots, \phi$.*

Proof. For each center c_t choose its nearest neighbor c'_t in $\mathcal{G}$, for $t = 1, \ldots, h$. Let $C' = \{c'_1, \ldots, c'_h\}$.

Fix a set $P_{i,j}$, and consider a point $q \in P_{i,j}$. Suppose that c_r is the closest center to q in C, c'_l is the closest center to q in C', where $1 \leq r, l \leq h$.

It is easy to verify $\|c_r c'_r\| \leq \varepsilon R/(24\beta)$ if $\mathbf{d}(c_r, \mathcal{A}) \leq R$, and $\|c_r c'_r\| \leq 2\varepsilon\mathbf{d}(c_r, \mathcal{A})/(24\beta)$ if $\mathbf{d}(c_r, \mathcal{A}) > R$. As such, we have $\|c_r c'_r\| \leq (\varepsilon/24\beta)(2\mathbf{d}(c_r, \mathcal{A}) + R)$.

On the other hand, due to the triangle inequality, $\mathbf{d}(c_r, \mathcal{A}) \leq \|qc_r\| + \mathbf{d}(q, \mathcal{A})$.

Therefore, if $\nu_C(q) \leq \nu_{C'}(q)$, we have

$$\begin{aligned}\nu_{C'}(q) - \nu_C(q) &\leq \|qc'_r\| - \|qc_r\| \leq \|c_r c'_r\| \\ &\leq \frac{\varepsilon}{24\beta}(2\nu_C(q) + 2\mathbf{d}(q, \mathcal{A}) + R),\end{aligned}$$

and similarly, if $\nu_{C'}(q) \leq \nu_C(q)$, we have

$$\begin{aligned}\nu_C(q) - \nu_{C'}(q) &\leq \|qc_l\| - \|qc'_l\| \leq \|c_l c'_l\| \\ &\leq \frac{\varepsilon}{24\beta}(2\nu_{C'}(q) + 2\mathbf{d}(q, \mathcal{A}) + R) \\ &\leq \frac{\varepsilon}{24\beta}(2\nu_C(q) + 2\mathbf{d}(q, \mathcal{A}) + R).\end{aligned}$$

Summing this up for all the points in $P_{i,j}$, we have

$$\begin{aligned}&|\nu_{C'}(P_{i,j}) - \nu_C(P_{i,j})| \\ &\leq \frac{\varepsilon}{24\beta}(2\nu_C(P_{i,j}) + 2\nu_{\mathcal{A}}(P_{i,j}) + |P_{i,j}| \cdot R).\end{aligned}$$

The claim about $\mathcal{S}_{i,j}$ can be proved similarly. □

LEMMA 4.3. *For any set C of at most k centers outside $\mathcal{U}$, $|\nu_C(P_{i,j}) - \nu_C(\mathcal{S}_{i,j})| \leq (\varepsilon/4)\nu_C(P_{i,j})$, for $i = 1, \ldots, m$ and $j = 0, \ldots, \phi$.*

Proof. Fix a set $P_{i,j}$. Note that $\mathcal{S}_{i,j}$ is a weighted subset from $P_{i,j}$, and $w(\mathcal{S}_{i,j}) = w(P_{i,j})$. By the triangle inequality,

$$\begin{aligned}|\nu_C(P_{i,j}) - \nu_C(\mathcal{S}_{i,j})| &\leq |P_{i,j}| \cdot \mathrm{diam}(P_{i,j}) \\ &\leq |P_{i,j}| \cdot \beta nR.\end{aligned}$$

Since C is outside $\mathcal{U}$, $\mathbf{d}(C, P_{i,j}) \geq (2^\phi - 1)\beta nR \geq (4/\varepsilon)\beta nR$. This implies $\beta nR \leq (\varepsilon/4)\mathbf{d}(C, P_{i,j}))$.

It follows

$$\begin{aligned}&|\nu_C(P_{i,j}) - \nu_C(\mathcal{S}_{i,j})| \\ &\leq |P_{i,j}| \cdot (\varepsilon/4)\mathbf{d}(C, P_{i,j}) \leq (\varepsilon/4)\nu_C(P_{i,j}),\end{aligned}$$

by combining above inequalities. □

LEMMA 4.4. *For any set C of k centers in $\mathbb{R}^d$, $|\nu_C(P) - \nu_C(\mathcal{S})| \leq \varepsilon \cdot \nu_C(P)$. This holds with probability $\geq 1 - \lambda$, where λ is the prespecified confidence parameter.*

Proof. Suppose $C = C_{\mathrm{in}} \cup C_{\mathrm{out}}$, where $C_{\mathrm{in}} = \{c_1, \ldots, c_h\}$ is the set of centers inside $\mathcal{U}$ and $C_{\mathrm{out}} = \{c_{h+1}, \ldots, c_k\}$ is the set of centers outside $\mathcal{U}$.

Consider a ring set $P_{i,j}$ where there exists $p, q \in P_{i,j}$ such that p is served by C_{in} and q is served by C_{out}. By the triangle inequality, we have $|\mathbf{d}(q, C_{\mathrm{in}}) - \mathbf{d}(p, C_{\mathrm{in}})| \leq \|pq\|$ and $|\mathbf{d}(p, C_{\mathrm{in}}) - \mathbf{d}(q, C_{\mathrm{out}})| = |\mathbf{d}(p, C) - \mathbf{d}(q, C)| \leq \|pq\|$. It follows

$$\begin{aligned}&|\mathbf{d}(q, C_{\mathrm{in}}) - \mathbf{d}(q, C_{\mathrm{out}})| \\ &= |(\mathbf{d}(q, C_{\mathrm{in}}) - \mathbf{d}(p, C_{\mathrm{in}})) + (\mathbf{d}(p, C_{\mathrm{in}}) - \mathbf{d}(q, C_{\mathrm{out}}))| \\ &\leq 2\|pq\|.\end{aligned}$$

In addition, we have $\|pq\| \leq \beta nR$ since p and q are in the same ring set $P_{i,j}$, and $\mathbf{d}(q, C_{\mathrm{out}}) \geq (2^\phi - 1)\beta nR \geq (4/\varepsilon)\beta nR$ since C_{out} is outside $\mathcal{U}$.

Combining above inequalities, we have

$$\begin{aligned}&|\mathbf{d}(q, C_{\mathrm{in}}) - \mathbf{d}(q, C_{\mathrm{out}})| \\ &\leq 2\|pq\| \leq 2\beta nR \leq (\varepsilon/2)\mathbf{d}(q, C_{\mathrm{out}}).\end{aligned}$$

This implies that, if we reassign the points of $P_{i,j}$ that are served by C_{out} to C_{in}, the cost $\nu_C(P_{i,j})$ changes by a factor of at most $\varepsilon/2$, and the same is true for the corresponding set $\mathcal{S}_{i,j}$. Therefore, it suffices to prove $|\nu_C(P) - \nu_C(\mathcal{S})| \leq (\varepsilon/2)\nu_C(P)$ after reassignment.

Next, consider the clustering cost after the reassignment described above. In those settings, each ring set $P_{i,j}$ (along with $\mathcal{S}_{i,j}$) is either served by C_{in} or by C_{out}, for $i = 1, \ldots, m$ and $j = 0, \ldots, \phi$. Let P_{in} (resp. $\mathcal{S}_{\mathrm{in}}$) denote the points of P (resp. $\mathcal{S}$) that is served by C_{in}, and P_{out} (resp. $\mathcal{S}_{\mathrm{out}}$) denote the points of P (resp. $\mathcal{S}$) that are served by C_{out}.

By Lemma 4.3, we have

$$\begin{aligned}|\nu_{C_{\mathrm{out}}}(P_{\mathrm{out}}) - \nu_{C_{\mathrm{out}}}(\mathcal{S}_{\mathrm{out}})| &\leq (\varepsilon/4)\nu_{C_{\mathrm{out}}}(P_{\mathrm{out}}) \\ &\leq (\varepsilon/4)\nu_C(P).\end{aligned}$$

CLAIM 4.1. $|\nu_{C_{\mathrm{in}}}(P_{\mathrm{in}}) - \nu_{C_{\mathrm{in}}}(\mathcal{S}_{\mathrm{in}})| \leq (\varepsilon/4)\nu_C(P)$.

Proof. We sketch the proof below, and omit the tedious details.

By Lemma 4.2, there exists a $C'_{\mathrm{in}} \subseteq \mathcal{G}$ such that $|C'_{\mathrm{in}}| = |C_{\mathrm{in}}|$, $\nu_{C_{\mathrm{in}}}(P_{\mathrm{in}})$ is approximately equal to $\nu_{C'_{\mathrm{in}}}(P_{\mathrm{in}})$, and $\nu_{C_{\mathrm{in}}}(\mathcal{S}_{\mathrm{in}})$ is approximately equal to $\nu_{C'_{\mathrm{in}}}(\mathcal{S}_{\mathrm{in}})$.

On the other hand, by Lemma 4.1, $\nu_{C'_{\mathrm{in}}}(\mathcal{S}_{\mathrm{in}})$ is approximately equal to $\nu_{C'_{\mathrm{in}}}(P_{\mathrm{in}})$. As such, $\nu_{C_{\mathrm{in}}}(P_{\mathrm{in}})$ is approximately equal to $\nu_{C_{\mathrm{in}}}(\mathcal{S}_{\mathrm{in}})$. □

Therefore, we have

$$\begin{aligned}&|\nu_C(P) - \nu_C(\mathcal{S})| \\ &= |\nu_{C_{\mathrm{in}}}(P_{\mathrm{in}}) + \nu_{C_{\mathrm{out}}}(P_{\mathrm{out}}) - \nu_{C_{\mathrm{in}}}(\mathcal{S}_{\mathrm{in}}) - \nu_{C_{\mathrm{out}}}(\mathcal{S}_{\mathrm{out}})| \\ &\leq (\varepsilon/2)\nu_C(P).\end{aligned}$$

Note that Claim 4.1 assumes $\nu_{\mathcal{A}}(P) \leq \beta\nu_{\mathrm{opt}}(P, k)$, which holds with probability $\geq 1 - \lambda/4$. It also uses Lemma 4.1, which holds with probability $\geq 1 - \lambda/2$. As such, the assertion of the lemma holds with probability $\geq 1 - \lambda$. □

THEOREM 4.1. *Given a set P of n points in $\mathbb{R}^d$ and a parameter $\lambda > 0$, one can compute a set $\mathcal{S}$ of size*

$$O\left(\frac{k\log n}{\varepsilon^2}\left(kd\log\frac{1}{\varepsilon}+k\log k+k\log\log n+\log\frac{1}{\lambda}\right)\right),$$

in time $O(ndk\log(1/\lambda))$. The set $\mathcal{S}$ is a (k,ε)-coreset of P for k-median clustering with probability $\geq 1-\lambda$.

If P is weighted, with total weight W, then the the running time is $O(ndk\log(1/\lambda)\log\log W)$, and the coreset is of size

$$O\left(\frac{k\log^2 W}{\varepsilon^2}\left(kd\log\frac{1}{\varepsilon}+k\log k+k\log\log W+\log\frac{1}{\lambda}\right)\right).$$

5 Applications

In this section, we provide applications for the (k,ε)-coreset construction described in Section 3 and Section 4. We can plug the coreset into any k-median algorithm that works on a weighted point set.

In the metric spaces, we apply the local search algorithm of Arya *et al.* [4].

THEOREM 5.1. *Given a set P of n points in a metric space, one can compute a $(10+\varepsilon)$-approximation k-median clustering of P in $O(nk+k^7\varepsilon^{-4}\log^5 n)$ time, with constant probability.*

In the Euclidean space $\mathbb{R}^d$, we would like to apply the $(1+\varepsilon)$-approximation algorithm of Kumar *et al.* [20, 21]. A simple extension of their algorithm to a coreset yields a $(1+\varepsilon)$-approximation algorithm for k-median clustering of P in $\mathbb{R}^d$. In particular, let $T(n,m)$ be the running time of their algorithm on the (k,ε)-coreset, the recurrence of $T(n,m)$ is

$$\begin{aligned} T(n,m) &= O(u(k,\varepsilon))T(n,m-1)+T(n/2,m) \\ &\quad +O((c(k,\varepsilon)+u(k,\varepsilon))d), \end{aligned}$$

where $u(k,\varepsilon)=O(2^{(k/\varepsilon)^{O(1)}})$, n is the total weight of the coreset, $c(k,\varepsilon)$ is the size of the (k,ε)-coreset, m is the number of centers. It is not difficult to show that $T(n,k)=O(d2^{(k/\varepsilon)^{O(1)}}c(k,\varepsilon)kn^\sigma)$, for any $\sigma>0$.

THEOREM 5.2. *Given a set P of n points in $\mathbb{R}^d$, one can compute a $(1+\varepsilon)$-approximation k-median clustering of P in time $O(ndk+2^{(k/\varepsilon)^{O(1)}}d^2n^\sigma)$, with constant probability, for any $\sigma>0$.*

Coresets were previously used to design approximation algorithms in the streaming model [2, 14]. In particular, Har-Peled and Mazumdar [14] used coresets to develop approximation algorithms for k-median clustering in the insertion-only streaming model. Randomized coreset construction described in Section 4 can also be used in the streaming model by the same techniques. See Appendix A for details. In particular, we have the following theorem.

THEOREM 5.3. *Given a stream P of n points in $\mathbb{R}^d$ and $\varepsilon>0$, one can maintain a (k,ε)-coreset for k-median clustering efficiently and use the coresets to compute $(1+\varepsilon)$-approximation to k-median clustering for the points seen so far. The coreset is correct with constant probability. The relevant complexities are:*

(i) Space to store the information: $O\big(k^2d\varepsilon^{-2}\log^8 n\big)$

(ii) Amortized update time: $O\big(kd\log^2 n\cdot\mathrm{polylog}(kd/\varepsilon)\big)$

6 Conclusions

In this paper, we used sampling techniques to extract a small (k,ε)-coreset for k-median clustering in both metric spaces and high dimensional Euclidean spaces. The coreset can be used to obtain fast approximation algorithms for k-median clustering. It is especially useful in the streaming model of computation, where the small storage space is desired. In particular, we provide the first streaming clustering algorithm that has space complexity with polynomial dependency on the dimension.

In addition, the small coreset leads to a $O(ndk+2^{(k/\varepsilon)^{O(1)}}d^2n^\sigma)$-time $(1+\varepsilon)$-approximation algorithm for k-median clustering in $\mathbb{R}^d$, which succeeds with constant probability, for any $\sigma>0$. This improves over the work of Kumar *et al.* [20, 21]. This result, together with the low dimensional result of Har-Peled and Mazumdar [14] indicates, surprisingly, that the expensive part in computing k-median clustering in $\mathbb{R}^d$, is answering nearest neighbor queries (this is the $O(ndk)$ term in the running time).

Furthermore, we believe that similar approximation algorithms exist for k-means clustering. (Essentially, all we need is a fast bi-criteria approximation algorithm for k-means clustering. It seems like adapting one of the k-median be-criteria algorithms for this purpose should be easy.) This would be included in the full version of the paper.

In light of the recent result of Har-Peled and Kushal [13], which constructed low dimensional coreset of size independent of n (but exponential in the dimension), it is natural to ask if one can construct a coreset of size with polynomial dependency on the dimension and with no dependency on n. We leave this as open problem for further research. A more intriguing possibility is that one can construct coresets of size independent of the dimension altogether, as was done in the min-enclosing ball case [7].

Acknowledgments

The author thanks Sariel Har-Peled for helpful discussions on the problems studied in the paper and

comments on the manuscript. The author also thanks referees for useful comments.

References

[1] P. K. Agarwal, S. Har-Peled, and K. Varadarajan. Geometric approximation via coresets. manuscript, 2004.

[2] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. *J. Assoc. Comput. Mach.*, 51(4):606–635, 2004.

[3] S. Arora, P. Raghavan, and S. Rao. Approximation schemes for Euclidean k-median and related problems. In *Proc. 30th Annu. ACM Sympos. Theory Comput.*, pages 106–113, 1998.

[4] V. Arya, N. Garg, R. Khandekar, K. Munagala, and V. Pandit. Local search heuristic for k-median and facility location problems. In *Proc. 33rd Annu. ACM Sympos. Theory Comput.*, pages 21–29, 2001.

[5] M. Badoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. In *Proc. 34th Annu. ACM Sympos. Theory Comput.*, pages 250–257, 2002.

[6] J. L. Bentley and J. B. Saxe. Decomposable searching problems i: Static-to-dynamic transformation. *J. Algorithms*, 1(4):301–358, 1980.

[7] M. Bădoiu and K. L. Clarkson. Optimal core-sets for balls. In *Proc. 14th ACM-SIAM Sympos. Discrete Algo.*, pages 801–802, 2003.

[8] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the k-median problem. In *Proc. 31st Annu. ACM Sympos. Theory Comput.*, pages 1–10, 1999.

[9] M. Charikar, L. O'Callaghan, and R. Panigrahy. Better streaming algorithms for clustering problems. In *Proc. 35th Annu. ACM Sympos. Theory Comput.*, pages 30–39, 2003.

[10] G. Frahling and C. Sohler. Coresets in dynamic geometric data streams. In *Proc. 37th Annu. ACM Sympos. Theory Comput.*, pages 209–217, 2005.

[11] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoret. Comput. Sci.*, 38:293–306, 1985.

[12] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams. In *Proc. 41th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 359–366, 2000.

[13] S. Har-Peled and A. Kushal. Smaller coresets for k-median and k-means clustering. In *Proc. 21th Annu. ACM Sympos. Comput. Geom.*, pages 126–134, 2005.

[14] S. Har-Peled and S. Mazumdar. Coresets for k-means and k-median clustering and their applications. In *Proc. 36th Annu. ACM Sympos. Theory Comput.*, pages 291–300, 2004.

[15] D. Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, 1992.

[16] P. Indyk. Sublinear time algorithms for metric space problems. In *Proc. 31st Annu. ACM Sympos. Theory Comput.*, pages 154–159, 1999.

[17] P. Indyk. Algorithms for dynamic geometric problems over data streams. In *Proc. 36th Annu. ACM Sympos. Theory Comput.*, pages 373–380, 2004.

[18] K. Jain and V. V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *Proc. 40th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 2–13, 1999.

[19] S. G. Kolliopoulos and S. Rao. A nearly linear-time approximation scheme for the euclidean k-median problem. In *Proc. 7th Annu. Euro. Sympos. Algorithms*, pages 378–389, 1999.

[20] A. Kumar, Y. Sabharwal, and S. Sen. A simple linear time $(1+\varepsilon)$-approximation algorithm for k-means clustering in any dimensions. In *Proc. 45nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 454–462, 2004.

[21] A. Kumar, Y. Sabharwal, and S. Sen. Linear time algorithms for clustering problems in any dimensions. In *Proc. 32nd Int. Colloq. Automata Lang. Prog.*, pages 1374–1385, 2005.

[22] R. Mettu and C. G. Plaxton. Optimal time bounds for approximate clustering. In *Proc. 18th Conf. Uncertainty Artif. Intell.*, pages 344–351, 2002.

[23] N. Mishra, D. Oblinger, and L. Pitt. Sublinear time approximate clustering. In *Proc. 12th ACM-SIAM Sympos. Discrete Algo.*, pages 439–447, 2001.

A Streaming

The algorithm of Har-Peled and Mazumdar [14] was based on standard dynamization technique of Bentley and Saxe [6] and the following observation about coresets.

OBSERVATION A.1. *(i) If C_1 and C_2 are the (k,ε)-coresets for disjoint sets P_1 and P_2 respectively, then $C_1 \cup C_2$ is a (k,ε)-coreset for $P_1 \cup P_2$.*

(ii) If C_1 is (k,ε)-coreset for C_2, and C_2 is a (k,δ)-coreset for C_3, then C_1 is a $(k,(1+\varepsilon)(1+\delta)-1)$-coreset for C_3.

Next, we adapt the algorithm of Har-Peled and Mazumdar to our randomized coresets.

Suppose that a sequence of points $p_1, p_2, \ldots$ in $\mathbb{R}^d$ arrive one by one. We want to compute k-median clustering of the points arrived so far. And we want the result to be correct with probability $\geq 1-\lambda$, where λ is a prespecified confidence parameter.

Conceptually, we use buckets $B_0, B_1, \ldots$ to store points. The capacity of bucket B_0 is M, where M is to be specified shortly, and the capacity of bucket B_i is $2^{i-1}M$, for $i \geq 1$. We will keep an invariant: B_i is either full or empty, for $i \geq 1$. When p_m arrives, we insert p_m into B_0. If B_0 has less than M points, then we are done. Otherwise, we move all the points

of B_0 into a virtual bucket B'_1. If B_1 is empty, move points of B'_1 into B_1, and we are done; otherwise we merge the points of B'_1 and B_1 into a virtual bucket B'_2. Then we try to move points of B'_2 into B_2. We continue the process until we reach a stage r where B_r is empty; and then the points of virtual bucket B'_r are moved into B_r.

However, we can not afford to keep every point in the buckets as above in the streaming model. Instead, we maintain a coreset Q_i (resp. Q'_i) for each bucket B_i (resp. virtual bucket B'_i), for $i = 0, 1, \ldots$, as follows: Q_0 is B_0 itself; and whenever the points of B'_r and B_r are merged into B'_{r+1}, we compute a (k, ρ_r)-coreset Q'_{r+1} of $Q_r \cup Q'_r$ with confidence parameter $\lambda_m = \lambda/m^3$, where $\rho_r = \varepsilon/cr^2$, m is the number of points received so far, and c is a large positive constant. Let $Q = \bigcup_{i\geq 0} Q_i$.

Claim A.1. *The set Q is a (k, ε)-coreset of the points received so far, with probability $\geq 1 - \lambda$.*

Proof. Recall that $\rho_r = \varepsilon/cr^2$, where c is a large positive constant. It is easy to verify that $\prod_{l=0}^{r}(1 + \rho_l) \leq 1+\varepsilon$ when c is a large enough constant for $r \geq 1$. On the other hand, Q_r is a $(k, \prod_{l=0}^{r}(1+\rho_l)-1)$-coreset of B_r, by Observation A.1. Therefore,Q_r is a (k, ε)-coreset of the points in B_r, and $Q = \bigcup_{i\geq 0} Q_i$ is a (k, ε)-coreset of the points in $\bigcup_{i\geq 0} B_i$. That is, Q is a (k, ε)-coreset of the points received so far.

When we process the newly arrived point p_m, our computation may fail with probability $\leq \lambda_m = \lambda/m^3$ whenever we compute a coreset with confidence parameter λ_m. When p_m arrives, it may trigger at most $\lceil \log_2 m \rceil$ coreset computations. As such, overall the algorithm may fail with probability $\leq \sum_{i=1}^{m} \frac{\lambda \log_2 i}{i^3} \leq \lambda$. □

Set $M = k^2\varepsilon^{-2}d$ and assume that we have received n points so far. Note that $|Q_0| \leq M$ and $|Q_1| = M$ (if Q_1 is not empty). For $i = 2, \ldots, \lceil \log_2 n \rceil$, Q_i has a total weight $2^{i-1}M$ (if it is not empty) and it is generated as a $\left(k, \varepsilon/ci^2\right)$-coreset of Q'_{i-1} and Q_{i-1} with confidence parameter at least λ/n^3, where c is a constant. By Theorem 4.1,

$$\begin{aligned} |Q_i| \;=\; & O(\, k\varepsilon^{-2}i^4(i + \log M)^2 \,(\, kd \log(ci^2/\varepsilon) \\ & + k \log k + \log(i + \log M) + \log(n^3/\lambda)\,)\,). \end{aligned}$$

If λ is a positive constant, the total storage requirement is

$$2M + \sum_{i=2}^{\lceil \log_2 n \rceil} |Q_i| = O(k^2 d\varepsilon^{-2} \log^8 n)$$

To analyze the update time of the data structure, observe that the amortized time dealing with Q_0 and Q_1 is constant; and for $j = 2, \ldots, \lceil \log_2 n \rceil$, Q_j is constructed after every $2^{j-1}M$ insertions are made. Therefore by Theorem 4.1, the amortized time spent for an update is

$$\begin{aligned} & O\left(\sum_{i=2}^{\lceil \log_2 n \rceil} \frac{1}{2^{i-1}M} |Q_{i-1}| \cdot kd \cdot \operatorname{loglog}|B_{i-1}| \cdot \log\frac{n^3}{\lambda}\right) \\ & = O\left(kd \log^2 n \cdot \operatorname{polylog}\left(\frac{kd}{\varepsilon}\right)\right). \end{aligned}$$