Core-Preserving Algorithms

Hamid Zarrabi-Zadeh*

Abstract

We define a class of algorithms for constructing coresets of (geometric) data sets, and show that algorithms in this class can be dynamized efficiently in the insertion-only (data stream) model. As a result, we show that for a set of points in fixed dimensions, additive and multiplicative ε-coresets for the k-center problem can be maintained in $O(1)$ and $O(k)$ time respectively, using a data structure whose size is independent of the size of the input. We also provide a faster streaming algorithm for maintaining ε-coresets for fat extent-related problems such as diameter and minimum enclosing ball.

1 Introduction

The data stream model of computation has recently attracted considerable interest due to growing applications involving massive data sets. In this model, data is presented to the algorithm one by one as a stream over time, and the algorithm must compute a function over the stream in only one pass, using a limited amount of storage.

The coreset framework is a fundamental tool for designing algorithms in the data stream model as it allows to compute a function approximately over the data stream by keeping only a small-size "sketch" of the input, called a coreset. Roughly speaking, a subset Q of the input set P is called an *ε-coreset* of P with respect to an optimization problem, if solving the optimization problem on Q gives an ε-approximate solution to the problem on the whole input set, P.

Several streaming algorithms have been developed over the past few years for various geometric problems using the notion of coresets [1, 6, 9, 14]. For all these problems, coresets defined satisfy the following two properties:

a) If Q is an ε-coreset of P and Q' is an ε-coreset of P', then $Q \cup Q'$ is an ε-coreset of $P \cup P'$;

b) If Q is an ε-coreset of S and S is an δ-coreset of P, then Q is an $(\varepsilon + \delta)$-coreset of P.

Using the above two properties and based on the general dynamization technique of Bentley and Saxe [5], Agarwal *et al.* [2] obtained the following result in the data stream model: If there is an ε-coreset of size $f(\varepsilon)$ for a problem, then one can solve the problem in the data stream model using $O(f(\varepsilon/\log^2 n)\log n)$ overall space, where n is the number of elements received so far in the stream.

In this paper, we show that for a special class of algorithms which we call core-preserving, the space complexity of the corresponding streaming algorithms can be reduced to $f(\varepsilon)$, using a simple bucketing scheme. The importance of this result is that the dependency of the space complexity to the input size, n, is removed. (Such a result was previously known only for the ε-coresets with respect to the extent measure [6, 4].) This independency to the input size is very important as the input size in the data streams is usually huge.

Our framework leads to improved algorithms for a number of problems in the data stream model, some of which are listed below. In the following, the input is assumed to be a stream of points in $\mathbb{R}^d$, where d is a constant.

- **(Additive) coreset for k-center**: We show that an additive ε-coreset for the k-center problem can be maintained in $O(k/\varepsilon^d)$ space and $O(1)$ amortized update time, improving the previous algorithm attributed to Har-peled [12] which requires $O(\text{poly}(k, 1/\varepsilon, \log n))$ space and similar time. This is indeed the first streaming algorithm maintaining an ε-coreset for this problem using a total space independent of n.

- **Multiplicative coreset for k-center**: For the k-center problem, we show that a multiplicative ε-coreset (as defined in Section 2) can be maintained in $O(k!/\varepsilon^{kd})$ space and $O(k)$ amortized update time. This is again the first streaming algorithm for this problem whose space is independent of the input size. This result immediately extends to a variant of the k-clustering problem in which the objective is to minimize the sum of the clusters radii [7, 10].

- **Coreset for fat measures**: For "fat" measures such as diameter and radius of the minimum enclosing ball, one can easily maintain an ε-coreset by just keeping the extreme points along $O(1/\varepsilon^{(d-1)/2})$ directions. The time and space complexity of this naïve algorithm is $O(1/\varepsilon^{(d-1)/2})$. In two-dimensions, using the recent algorithm of Agarwal

*School of Computer Science, University of Waterloo, Waterloo, Ont. N2L 3G1, Canada; hzarrabi@uwaterloo.ca

and Yu [4], one can improve the update time from $O(\sqrt{1/\varepsilon})$ to $O(\log(1/\varepsilon))$. We show that the update time in 2D can be further reduced to $O(1)$ using our framework. Moreover, the update time in three dimensions is reduced from $O(1/\varepsilon)$ to $O(\log(1/\varepsilon))$ using our algorithm. A slight improvement in higher dimensions is implied as well.

2 Preliminaries

Let P be a set of points in $\mathbb{R}^d$. A *k-clustering* of P is a set $\mathcal{B}$ of k balls that completely cover P. We denote by $\text{rad}(b)$ the radius of a ball b, and define $\text{rad}(\mathcal{B}) = \max_{b \in \mathcal{B}} \text{rad}(b)$. A *$\delta$-expansion* of $\mathcal{B}$ is obtained by increasing the radius of each ball of $\mathcal{B}$ by an additive factor of δ.

Definition 1 A set $Q \subseteq P$ is called an *additive ε-coreset* of P for the k-center problem, if for every k-clustering $\mathcal{B}$ of Q, P is covered by an $(\varepsilon \cdot \text{rad}(\mathcal{B}))$-expansion of $\mathcal{B}$.

We denote by $(1+\varepsilon)\mathcal{B}$ a clustering obtained from $\mathcal{B}$ by expanding each ball $b \in \mathcal{B}$ by a factor of $\varepsilon \cdot \text{rad}(b)$.

Definition 2 A set $Q \subseteq P$ is called a *multiplicative ε-coreset* of P for the k-center problem, if for every k-clustering $\mathcal{B}$ of Q, P is covered by $(1+\varepsilon)\mathcal{B}$.

Given two points $p, q \in \mathbb{R}^d$, we say that p is *smaller* than q, if p lies before q in the lexicographical order of their coordinates. Throughout this paper, we denote by $\lfloor x \rfloor_2$ the largest (integer) power of 2 which is less than or equal to x.

3 Core-Preserving Algorithms

In this section, we formally define the notion of core-preserving algorithms, and show how it can be used to efficiently maintain coresets in data streams.

Definition 3 *Let $\mathcal{A}$ be an (offline) algorithm that for every input set P, computes an ε-coreset $\mathcal{A}(P)$ of P. We call $\mathcal{A}$* core-preserving, *if for every two sets R and S, $\mathcal{A}(R \cup \mathcal{A}(S))$ is an ε-coreset of $R \cup S$.*

For $R = \emptyset$, the above property implies that $\mathcal{A}(\mathcal{A}(S))$ is an ε-coreset of S. It means that repeated calls to a core-preserving algorithm on a set S always returns an ε-coreset of S. This is why the algorithm is called "core-preserving".

Theorem 1 *Let $\mathcal{A}$ be a core-preserving algorithm that for any set S, computes an ε-kernel of S of size $O(\mathcal{S}_\mathcal{A}(\varepsilon))$ in time $O(\alpha|S| + \mathcal{T}_\mathcal{A}(\varepsilon))$. Then for every stream P, we can maintain an ε-coreset of P of size $O(\mathcal{S}_\mathcal{A}(\varepsilon))$ using $O(\mathcal{S}_\mathcal{A}(\varepsilon))$ total space and $O(\alpha + \mathcal{T}_\mathcal{A}(\varepsilon)/\mathcal{S}_\mathcal{A}(\varepsilon))$ amortized time per update.*

Proof. The function Insert described below inserts a date item p into the stream P and returns an ε-kernel of P. Initially, Q and R are empty sets.

```
Insert($p$):
$R \leftarrow R \cup \{p\}$
if $|R| > \mathcal{S}_\mathcal{A}(\varepsilon)$ then
    $Q \leftarrow \mathcal{A}(R \cup Q)$
    $R \leftarrow \emptyset$
return $Q \cup R$
```

The algorithm divides the input stream P into buckets of size $\lceil \mathcal{S}_\mathcal{A}(\varepsilon) \rceil$. At any time, only the last bucket is active which is maintained in the set R. Let $S = P \backslash R$. The algorithm maintains an ε-coreset of S in Q. Upon arrival of a new item p, it is first added to the active bucket R, and if R is full, algorithm $\mathcal{A}$ is invoked to compute an ε-coreset of $R \cup Q$. The correctness of the algorithm immediately follows from the facts that $\mathcal{A}$ is core-preserving and Q is an ε-coreset of S; thus, $\mathcal{A}(R \cup Q)$ is an ε-coreset of $R \cup S = P$.

The total space used by the algorithm is bounded by $|Q| + |R| = O(\mathcal{S}_\mathcal{A}(\varepsilon))$. Algorithm $\mathcal{A}$ is invoked once per $\lceil \mathcal{S}_\mathcal{A}(\varepsilon) \rceil$ inserts. Since each call to $\mathcal{A}$ requires $O(\alpha|S| + \mathcal{T}_\mathcal{A}(\varepsilon))$ time, the amortized update time per input is $O(\alpha + \mathcal{T}_\mathcal{A}(\varepsilon)/\mathcal{S}_\mathcal{A}(\varepsilon))$. □

Theorem 1 yields two major improvements over the general Bentley-Saxe method used in [2]: First of all, the total space required is reduced from $O(\mathcal{S}_\mathcal{A}(\varepsilon/\log^2 n)\log n)$ to $O(\mathcal{S}_\mathcal{A}(\varepsilon))$, which is independent of n. Secondly, the running time in the worst case is reduced from $O([\alpha\mathcal{S}_\mathcal{A}(\varepsilon/\log^2 n) + \mathcal{T}_\mathcal{A}(\varepsilon/\log^2 n)]\log n)$ to only $O(\alpha|P| + \mathcal{T}_\mathcal{A}(\varepsilon))$, again independent of n.

4 Additive Coreset for k-Center

In this section, we provide an efficient streaming algorithm for maintaining an additive ε-coreset for the k-center problem in fixed dimensions.

Lemma 2 *There is a core-preserving algorithm that for any given point set $P \subseteq \mathbb{R}^d$, computes an additive ε-coreset for the k-center problem of size $O(k/\varepsilon^d)$ in time $O(|P| + k/\varepsilon^d)$.*

Proof. Let $r^*(P)$ be the radius of the optimal k-clustering of P, and $\tilde{r}(P)$ be a 2-approximation of $r^*(P)$, i.e., $r^*(P) \leqslant \tilde{r}(P) \leqslant 2r^*(P)$.

We first define some notations: Let $\mathcal{G}_\alpha$ be a uniform grid of side length α, and $X_\alpha(P)$ be the set of all $p \in P$, such that p is the smallest point in a non-empty grid cell of $\mathcal{G}_\alpha$. Let $\delta(P) = \lfloor \varepsilon\tilde{r}(P)/(4d^{1/2}) \rfloor_2$. Our core-preserving algorithm is as follows: given a point set P, we first compute $\delta = \delta(P)$, and return $X_\delta(P)$ as the output. It is easy to observe that any k-clustering of $X_\delta(P)$, when expanded by a factor of $\varepsilon r^*(P)$, covers all the grid cells containing at least one point from P, and therefore, $X_\delta(P)$ is an ε-coreset of P [3, 13].

Let R and S be two arbitrary point sets in $\mathbb{R}^d$, and let Q be an ε-coreset of S computed by our algorithm. To show that our algorithm is core-preserving, we need to prove that for any input of the form $P = R \cup Q$, the algorithm returns an ε-coreset of $R \cup S$.

Let $\delta = \delta(P)$, $\sigma = \delta(S)$, and $\rho = \max\{\delta(P), \delta(S)\}$. Obviously, $X_\rho(R \cup S)$ is an ε-coreset of $R \cup S$, because both P and S are subsets of $R \cup S$, and hence, $\max\{\tilde{r}(P), \tilde{r}(S)\} \leqslant 2r^*(R \cup S)$. We claim that $X_\rho(R \cup S) \subseteq X_\delta(R \cup Q)$. Since ρ/δ (resp., ρ/σ) is a non-negative power of 2, every grid cell of $\mathcal{G}_\delta$ (resp., $\mathcal{G}_\sigma$) is completely contained in a grid cell of $\mathcal{G}_\rho$ (see Figure 1). Let p be the smallest point of $R \cup S$ in a grid cell c of $\mathcal{G}_\rho$. Two cases arise:

- $p \in R$: in this case, p is the smallest point of a cell $c' \in \mathcal{G}_\delta$ (otherwise, there is a point p' smaller than p in c', which is smaller than p in c as well, a contradiction). Therefore, $p \in X_\delta(R \cup Q)$.
- $p \in S$: here, p is simultaneously the smallest point of a cell $c' \in \mathcal{G}_\sigma$ and a cell $c'' \in \mathcal{G}_\delta$ (otherwise, if there is a smaller point p' in either c' or c'', it would be picked instead of p as the smallest point of c, a contradiction). Since p is the smallest point in c', we have $p \in Q$, and since p is the smallest point of c'', we conclude that $p \in X_\delta(R \cup Q)$.

Therefore, any $p \in X_\rho(R \cup S)$ is contained in $X_\delta(R \cup Q) = X_\delta(P)$, which completes the proof.

For the space complexity, note that every ball of an optimal k-clustering of P intersects $O(1/\varepsilon^d)$ grid cells of $\mathcal{G}_\delta$. Therefore, the size of the resulting ε-coreset is $O(k/\varepsilon^d)$. We can use a linear-time implementation of Gonzalez's algorithm [11, 12] to compute a 2-approximation of $r^*(P)$, and therefore, the total running time required is $O(|P| + k/\varepsilon^d)$. □

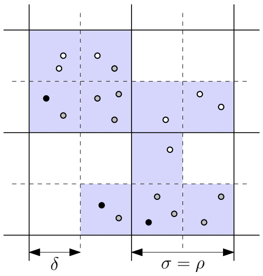

Figure 1: Additive coreset for k-center. The points of R, Q, and $S \backslash Q$ are shown in white, black, and gray, respectively.

Plugging Lemma 2 into the general framework provided in Theorem 1, we immediately get the following result.

Theorem 3 *Given a stream of points P in $\mathbb{R}^d$, an additive ε-coreset for the k-center problem of size $O(k/\varepsilon^d)$ can be maintained using $O(k/\varepsilon^d)$ total space and $O(1)$ amortized time per update.*

The above results also hold for any L_p metric: it just suffices to replace $d^{1/2}$ by $d^{1/p}$ in the definition of $\delta(P)$. The algorithm for multiplicative ε-coresets is omitted in this extended abstract.

5 Coresets for Fat Extent-Related Problems

Given a point set $P \subseteq \mathbb{R}^d$, let $\mathbb{B}(P)$ denote the minimum axis-parallel hyperbox enclosing P. We denote by $\ell(P)$ the length of the longest side of $\mathbb{B}(P)$. A subset $Q \subseteq P$ is called an *additive ε-kernel of* P, if for all $u \in \mathbb{S}^{d-1}$,

$$w(Q, u) \geqslant w(P, u) - \varepsilon\ell(P),$$

where $w(P, u) = \max_{p,q \in P} \langle p - q, u\rangle$.

A function $\mu(\cdot)$ defined over subsets of $\mathbb{R}^d$ is called a *fat measure*, if there exists a constant $\alpha > 0$ such that for any additive ε-kernel Q of P, $\alpha\mu(P) \leqslant \mu(Q) \leqslant \mu(P)$. Examples of fat measures are diameter, radius of the minimum enclosing ball, and width of the smallest enclosing hypercube. Obviously, if Q is an additive ε-kernel of P and μ is a fat measure, then Q is an (ε/α)-coreset of P with respect to μ.

Given a point set $P \subseteq \mathbb{R}^d$, an additive ε-kernel of P can be computed efficiently using an adaptation of the simple grid-rounding method proposed in [6, 15] based on Dudley's construction [8]. The algorithm is described in the following lemma.

Lemma 4 *There is a core-preserving algorithm that for every point set $P \subseteq \mathbb{R}^d$, computes an additive ε-kernel of P of size $O(1/\varepsilon^{(d-1)/2})$ in $O(|P| + 1/\varepsilon^{d-(3/2)})$ time for $d \geqslant 2$, or in $O((|P| + 1/\varepsilon^{d-2})\log(1/\varepsilon))$ time for $d \geqslant 3$.*

Proof. We assume w.l.o.g. that conv(P) contains the origin. Let $\mathcal{B}(P)$ be the smallest hypercube centered at the origin containing P. If $\ell'(P)$ denotes the side length of $\mathcal{B}(P)$, then obviously $\ell(P) \leqslant \ell'(P) \leqslant 2\ell(P)$.

Let $B = \mathcal{B}(P)$. By a simple scaling, we may assume that $B = [-1, 1]^d$. Let $\mathcal{R}$ be the set of points of a $\sqrt{\varepsilon}$-grid over the boundary of the cube $[-2, 2]^d$, and let p_r denote the nearest neighbor of a point $r \in \mathcal{R}$ in the set P (see Figure 2). Let $\mathcal{Q} = \{p_r \mid r \in \mathcal{R}\}$. Obviously, $|\mathcal{Q}| \leqslant |\mathcal{R}| = O(1/\varepsilon^{(d-1)/2})$. Moreover, $\mathcal{Q}$ is an additive ε-kernel of P with the argument provided below. The running time follows immediately from the fast implementation of Chan using the discrete nearest neighbor queries [6].

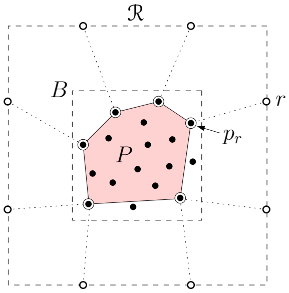

Figure 2: Construction of additive ε-kernel.

Consider two arbitrary point sets R and S in $\mathbb{R}^d$, and let Q be an additive ε-kernel of S computed by our algorithm. In order for our algorithm to be core-preserving, we need to show that for any input of the form $P = R \cup Q$, the algorithm returns an additive ε-kernel of $R \cup S$.

We adapt the proof from [6]. Fix a unit vector $u \in \mathbb{S}^{d-1}$ and a point $p \in R \cup S$. Obviously, there is a point $r \in \mathcal{R}$ such that $\angle(r-p, u) \leqslant \arccos(1-\varepsilon/8)$ (See [6], Observation 2.3). If $p_r \in S$, then by our construction there is a point $q \in Q$ such that $\|r-q\| \leqslant (1+c\varepsilon)\|r-p_r\|$ (details omitted). If $p_r \in R$, we simply set $q = p_r$. Therefore,

$$\begin{aligned} & \|r-q\| \leqslant (1+c\varepsilon)\|r-p\| \\ \Rightarrow\quad & (1-\varepsilon/8)\,\langle r-q, u\rangle \leqslant (1+c\varepsilon)\,\langle r-p, u\rangle \\ \Rightarrow\quad & \langle r-q, u\rangle - 3\sqrt{d}\varepsilon/8 \leqslant \langle r-p, u\rangle + 3c\sqrt{d}\varepsilon \\ & \quad (\text{since } \|r-p\| \leqslant 3\sqrt{d} \text{ and } \|r-q\| \leqslant 3\sqrt{d}) \\ \Rightarrow\quad & \langle p, u\rangle \leqslant \langle q, u\rangle + 3\sqrt{d}(c+1/8). \end{aligned}$$

It means that the projections of p and q in direction u differ by at most $O(\varepsilon)$. Since $\ell(P) \geqslant 1/2$, we conclude that $\langle p-q, u\rangle = O(\varepsilon)\ell(P)$ in every direction u, which completes the proof. □

Combining Lemma 4 with Theorem 1, we get the following result:

Theorem 5 *Given a stream of points P in $\mathbb{R}^d$ and a fat measure μ, an ε-coreset of P with respect to μ can be maintained using $O(1/\varepsilon^{(d-1)/2})$ total space and $\max\{O(1), O((1/\varepsilon^{(d-3)/2})\log(1/\varepsilon))\}$ amortized time per update.*

Remark. Using our framework to maintain ε-coresets of fat sets as a subroutine, we have recently succeeded to obtain a streaming algorithm for maintaining ε-coresets with respect to the general extent measure using near optimal space [16]. This leads to improved streaming algorithms for a wide variety of geometric optimization problems, including width, minimum enclosing cylinder, minimum-width enclosing annulus, minimum-width enclosing cylindrical shell, etc.

Acknowledgements The author would like to thank Timothy M. Chan for his helpful comments.

References

[1] P. K. Agarwal and S. Har-Peled. Maintaining approximate extent measures of moving points. In *Proc. 12th ACM-SIAM Sympos. Discrete Algorithms*, pages 148–157, 2001.

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

[3] P. K. Agarwal and C. M. Procopiuc. Exact and approximation algorithms for clustering. *Algorithmica*, 33(2):201–226, 2002.

[4] P. K. Agarwal and H. Yu. A space-optimal data-stream algorithm for coresets in the plane. In *Proc. 23rd Annu. ACM Sympos. Comput. Geom.*, pages 1–10, 2007.

[5] J. L. Bentley and J. B. Saxe. Decomposable searching problems I: Static-to-dynamic transformations. *J. Algorithms*, 1:301–358, 1980.

[6] T. M. Chan. Faster core-set constructions and data stream algorithms in fixed dimensions. *Comput. Geom. Theory Appl.*, 35(1–2):20–35, 2006.

[7] M. Charikar and R. Panigrahy. Clustering to minimize the sum of cluster diameters. *J. Comput. Systems Sci.*, 68:417–441, Mar. 2004.

[8] R. M. Dudley. Metric entropy of some classes of sets with differentiable boundaries. *J. Approx. Theory*, 10:227–236, 1974.

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

[10] M. Gibson, G. Kanade, E. Krohn, I. A. Pirwani, and K. Varadarajan. On clustering to minimize the sum of radii. In *Proc. 19th ACM-SIAM Sympos. Discrete Algorithms*, pages 819–825, 2008.

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

[12] S. Har-Peled. Clustering motion. *Discrete Comput. Geom.*, 31(4):545–565, 2004.

[13] S. Har-Peled. No Coreset, No Cry. In *Proc. 24th Conf. Found. Soft. Tech. and Theoret. Comput. Sci.*, pages 324–335, 2004.

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

[15] H. Yu, P. K. Agarwal, R. Poreddy, and K. R. Varadarajan. Practical methods for shape fitting and kinetic data structures using core sets. In *Proc. 20th Annu. ACM Sympos. Comput. Geom.*, pages 263–272, 2004.

[16] H. Zarrabi-Zadeh. An almost space-optimal streaming algorithm for coresets in fixed dimensions. In *Proc. 16th Annu. European Sympos. Algorithms*, 2008, to appear.