

Streaming algorithms for proximity problems in high dimensions

R. Sharathkumar
Department of Computer Science
Duke University, Durham NC 27708

December 9, 2008

Abstract

In this project, we study proximity problems in high dimensional space. We give efficient algorithms in the data stream model that compute an approximation of the Minimum Enclosing Ball and diameter of a point set. We also give a simple insertion only data structure that answers approximate farthest point queries.

1 Introduction

At a high level, proximity problems can be seen as maximizing or minimizing distances between pairs of points subject to a set of constraints. Well known single-shot proximity problems include computing the closest pair, diameter, minimum enclosing ball, k -center clustering of a given point set. Furthermore there are data structures that support proximity queries such as nearest-neighbor queries, farthest-point queries and testing of clustering. Most proximity problems have efficient solutions provided the input is from a low dimensional space [2]. Unfortunately the space complexity and/or running times of many of these algorithms grow exponentially in the dimension d of the problem. Thus, these solutions are inefficient when d is large. Research has thus focussed on approximation algorithms for proximity problems in high-dimensions [1] [3]. The running time and the working space of these algorithms are polynomial in d .

Many practical applications have massive input data sets with each element of the data arriving one at a time. Storing the entire data is expensive. This motivates the need to design data structures and algorithms in the *data stream model*. Algorithms in the data stream model allow only one pass over the data. Further, these algorithms are allowed limited working storage space. See [4]. For several proximity problems, an exact solution would require $\Omega(n)$ working space. Also, any approximation algorithms to several proximity problems (like the closest pair) require $\Omega(n)$ working space indicating the difficulty in designing data streaming algorithms for proximity problems.

In this project, we consider the following proximity problems.

Definition 1 *Minimum Enclosing Ball (MEB) problem:* Given a set S of points in \mathbb{R}^d , compute the ball \mathbb{B} with minimum radius that encloses all the points in S .

Definition 2 *Diameter Problem ($\text{diam}(S)$):* Given a set S of points in \mathbb{R}^d , compute the distance between pairs of points $p, q \in S$ such that $\|pq\| \geq \|rs\|$ for any $r, s \in S$.

Definition 3 *k -center:* Given a set S of points in \mathbb{R}^d and an integer k , compute a set $\Sigma = \{\sigma_1, \dots, \sigma_k\}$ of k points in \mathbb{R}^d such that

$$\max_{p \in P} \min_{\sigma \in \Sigma} \|p - \sigma\|$$

is minimized.

Definition 4 *c -farthest point query problem (CFP):* Given a point set $S \subset \mathbb{R}^d$, construct a data structure D , which given any query point $p \in \mathbb{R}^d$ outputs a point t such that $c\|pt\| \geq \|pq\|$ for any $q \in S$.

For each of these problems, we design data-streaming algorithms with working space polynomial in d and independent of n and which return an approximate solution. We design an insertion-only data structure for the CFP problem. The size of the data structure is polynomial in d and independent of n .

Throughout this paper, we use the following notations. $B(c, r)$ is used to refer to a ball B with center c and radius r . We use the set S to denote a set of input points (or a stream of points). The basis K of a ball B is a set of at most $d + 1$ points whose MEB is the ball B . An ε -expansion of a ball $B(c, r)$ is a ball $B(c, (1 + \varepsilon)r)$.

The next section will summarize the existing work on MEB, diameter, CFP, k -center and related problems.

2 Related Work

Minimum Enclosing Ball Minimum Enclosing Ball has been extensively studied in computational geometry and has several theoretical and practical applications. Minimum enclosing ball can be used to reduce between several proximity problems [5]. Thus efficient solutions to the minimum enclosing ball results in efficient solutions to various other proximity problems. Efficient computation of MEB has applications in machine learning as well [6]. Recent work, motivated by machine learning, shows the need to develop efficient streaming algorithms for computing MEB [8]. There are several solutions to compute the exact minimum enclosing ball. We can formulate the problem of computing minimum enclosing ball of a point set as a linear program. This linear program can be solved in time $O(d^{O(d)}n)$ using standard linear programming algorithms. Thus, for fixed d , we have a linear time exact algorithm. Unfortunately, running times of all known exact solutions to the MEB problem scale exponentially in d . This motivates the need of designing approximation algorithms which have running times polynomial in d . In fixed dimensions, ε -kernel proposed by Agarwal et al. [2] can be used to compute the approximate minimum enclosing ball in time $O(n + 1/\varepsilon^d)$. This can be extended to streaming algorithm with working space of $O((1/\varepsilon)^d)$ [7]. Again, these solutions are efficient when d is fixed.

In high dimensions, there is a trivial 2-approximation algorithm.

Algorithm 1 *Approx – MEB(S)*

- 1: Pick an arbitrary point $p \in S$
 - 2: $q = \operatorname{argmax}_{p' \in P} \|p - p'\|$
 - 3: **return** Ball centered at p of radius $\|p - q\|$
-

Badoiu and Clarkson [3] gave a *coreset* based algorithm which computes the MEB in time $O(nd/\varepsilon + 1/\varepsilon^5)$. The efficiency of their algorithm hinges on the fact that for constructing the minimum enclosing ball it is sufficient to look at a set $C \subseteq S$ of size $O(1/\varepsilon)$ such that the MEB(S) has a radius $(1 + \varepsilon)$ times the radius of MEB(C). The algorithm for constructing such an C is described below.

Algorithm 2 *MEB – Coreset(S)*

- 1: Pick an arbitrary point $p_i \in S$.
 - 2: Let C be initialized to $\{p_i\}$.
 - 3: **for** $i = 1$ to $2/\varepsilon$ **do**
 - 4: Construct MEB(C) and let c_i be the center of MEB(C).
 - 5: Find p_{i+1} such that p_{i+1} is the farthest neighbor of c_i in S .
 - 6: $C = C \cup \{p_i\}$
 - 7: **end for**
 - 8: **return** C
-

While the working space of this algorithm is $O(d/\varepsilon)$, it requires $O(1/\varepsilon)$ passes of the data. Solutions with similar bounds were also given by Kumar et al [9]. A crucial lemma in their lemma is the following

Lemma 1 *Let $B(c, r)$ be the MEB of a point set $S \subset \mathbb{R}^d$, then any closed halfspace that contains the c must also contain at least one point from S that is at a distance r from c .*

The trivial 2-approximation algorithm requires $O(d)$ working space and can be easily modified to work in the data stream model. A simple 3/2-approximation algorithm in the data stream model was suggested by Chan and Zarrabi-Zadeh [10]. They also showed that there is a no $(1 + \sqrt{2}/2)$ approximation algorithm in the data stream model which uses no more than d space.

Algorithm 3 *MEB – streaming(S)*

```
1:  $B = \phi$ 
2: for each point  $p$  in the input stream  $S$  do
3:   if  $p$  is outside  $B$  then
4:      $B = \text{MEB}(B \cup p)$ 
5:   end if
6: end for
7: return  $B$ 
```

Diameter Computing the farthest pair of points is another well studied problem in computational geometry. All non-trivial exact solutions have almost quadratic running time. ε -kernel [2] can be used to compute the approximate diameter in fixed dimensions. In high dimensions, there is a trivial factor 2 approximation which requires $O(d)$ working space and works in the data stream model. For this problem, Indyk [11] has improved the approximation ratio to $c \geq \sqrt{2}$ using $O(dn^{1/(c^2-1)} \log n)$ working space. A simple two pass algorithm with $O(d)$ working space gives a $\sqrt{3}$ approximation to the diameter problem.

c -Farthest point query : Farthest point queries are important proximity data structure problem. This data structure can be trivially used to obtain a c -approximation of diameter of the point set. Goel et al. [5] showed how MEB of a given point set can be used to compute the c -farthest point queries for $c = \sqrt{2}$. The static data structure thus obtained requires $O(d^2)$ space. An insertion-only version of this data structure for $c > \sqrt{2}$ would automatically give a streaming algorithm to the diameter problem.

k -center problem k -center is an important clustering problem. The problem is known to be *NP*-Hard. In higher dimensions, the problem is known to be *NP*-Hard to approximate within a factor of 2 [14] [12]. Gonzalez [16] introduced the farthest point based greedy heuristic which gives a factor 2 approximation in time $O(kn)$. Later, Feder and Greene [14] improved the running time to $O(n \log k)$ and showed this to be optimal.

Several practical applications demand online clustering algorithms. Charikar et al. [15] proposed an incremental clustering model. Within this framework they suggested several constant factor approximations for k -center clustering. The best heuristic was a factor 6 approximation (but it required to solve an *NP*-Complete problem). A simpler factor 8 approximation was also suggested. Their algorithms work in the general metric space. In the euclidean space, the approximation ratio improves to 6.83.

3 Our Results

In this project, We give a simple insertion-only data structure of size $O(d^2/\varepsilon \log 1/\varepsilon)$ which computes a $\sqrt{2} + \varepsilon$ -approximation of MEB. This data structure also answers c -farthest point queries for $c = \sqrt{2} + \varepsilon$ and consequently helps in maintaining a $\sqrt{2} + \varepsilon$ -approximation of the diameter of point set.

We also state a few results which are not part of this report.

4 $(\sqrt{2} + \varepsilon)$ -approximation for MEB and CFP

In this section, we present a simple (insertion only) data structure which maintains a $(\sqrt{2} + \varepsilon)$ approximation of the MEB and also answers $(\sqrt{2} + \varepsilon)$ -farthest point queries. Given that this data structure answers farthest point queries, it can be trivially modified to maintain a $\sqrt{2} + \varepsilon$ -approximation of the diameter of the points in the stream.

The data structure stores a subset $C \subset S$ of the input set of points at any given point in time. The size of the subset is $O(d/\varepsilon^2 \log 1/\varepsilon)$. We define the notion of *blurred MEB* which can be seen as a set of $O(1/\varepsilon^2 \log 1/\varepsilon)$ balls where each ball is a minimum enclosing ball of some subset of S . The union of the basis of each of these balls is the coreset C .

ε -Blurred MEB Given a set of points S , a blurred MEB is a set of balls $B = \{B_1(c_1, r_1), \dots, B_u(c_u, r_u)\}$ and a set of basis $K = \{K_1, \dots, K_u\}$, $u \in O(1/\varepsilon \log 1/\varepsilon)$ such that $K_i \subseteq S$ is the basis of B_i and

- (i) $\forall j \leq i, K_j \subseteq B_i$
- (ii) For every point $p \in S$, there exists a ball $B_i \in B$ such that the ε -expansion of B_i contains p .

Data structure to maintain ε -blurred MEB We give an data structure to compute the blurred MEB of a point set. We initialize $B = \phi$ and $K = \phi$. The update algorithm is as follows.

Algorithm 4 *MEB – Update*(p)

- 1: **if** there is no ball $B_i \in B$ such that p is in ε -expansion of B_i **then**
 - 2: $B^* = \text{MEB}(\cup_{i=1}^u K_i \cup \{p\})$
 - 3: K^* : basis of B^* , r : radius of B^* .
 - 4: $B_{del} : \{B_i | B_i \in \text{Brad}(B_i) \leq \varepsilon r/4\}$
 - 5: $K_{del} : \{K_i | B_i \in B_{del}\}$
 - 6: $B = B \cup B^* \setminus B_{del}$
 - 7: $K = K \cup K^* \setminus K_{del}$
 - 8: **end if**
-

Correctness To show the correctness, we show that conditions (i) and (ii) are satisfied after every update. If the point p is in the ε -expansion of ball $B_i \in B$, then both conditions (i) and (ii) trivially hold. Otherwise, the algorithm computes a ball B^* . Condition (i) is satisfied if we show that all $K_i \subseteq B^*$. This is true by construction, since B^* is the MEB of the union of all K_i .

It remains to show that condition (ii) is satisfied at the end of each update. Condition (ii) can be violated only if there is a point $p \in S$ such that it is contained in the ε -expansion of a ball in B_{del} but is not contained in the ε -expansion of any other ball. The following lemmas prove that no such point exists implying condition (ii) holds.

Lemma 2 *Let S_1 and S_2 be the basis of two balls $B_1(c_1, r_1)$ and $B_2(c_2, r_2)$. If $S_1 \subset B_2$ then the center $c_1 \in B_2$.*

Proof: Assume that the center c_1 lies outside B_2 . Consider a halfspace H passing through c_1 and that doesn't contain B_2 . From Lemma 1 any halfspace that contains the center of a ball also contains at least one point in the basis. This implies that H contains a point p . This leads to a contradiction of the condition $S_1 \subset B_2$ since p is a point in S_1 and p is contained in H which has no intersection with B_2 . \square

Lemma 3 ε -expansion of B^* contains ε -expansion of B_i for all $B_i \in B_{del}$.

Proof: Consider any $B_i \in B_{del}$. Clearly the basis of B_i , $K_i \in B^*$. Thus, from Lemma 2, it follows that the center of B_i , c_i is in B^* . Consider any point q in the ε -expansion of B_i . Consider the triangle formed by c_i , c^* and q . From triangular inequality, it follows that $\|c^*q\| \leq \|cc^*\| + \|qc^*\| \leq r + (1 + \varepsilon)\varepsilon r/4 \leq (1 + \varepsilon)r$. Thus q is in the ε -expansion of $B^*(c^*, r^*)$. \square

Space complexity Space complexity of blurred MEB is the maximum size of $\cup K_i = O(d|B|)$. Thus, to get a bound on the working space of our algorithm, we need to bound the size of the set B . To bound $|B|$, note that the radius of each ball $B_i \in B$ lies in the interval $(r_1, 4r_1/\varepsilon)$ where r_1 is the radius of B_1 . Further, we show that for any two consecutive ball B_i and B_{i+1} , $r_{i+1} \leq (1 + \varepsilon^2)r_i$. This implies that if we break the interval $(r_1, 4r_1/\varepsilon)$ into a set I of $O(1/\varepsilon^2 \log 1/\varepsilon)$ intervals each of length $\varepsilon^2 r$, then for each $i \in I$, there can be at most one ball whose radius lies within the interval i . This implies that the total number of balls in B is at most $O(1/\varepsilon^2 \log 1/\varepsilon)$ implying that the working space of our algorithm is $O((d/\varepsilon)^2 \log 1/\varepsilon)$.

The following lemma remains to be shown.

Lemma 4 Let $B(c, r)$ be the MEB of a set S of points. Let p be a point such that p is not in the ε -expansion of B . Then the MEB of $S \cup \{p\}$, $B_1(c_1, r_1)$, has radius $r_1 \geq (1 + \varepsilon^2)r$.

Proof: Let H be a hyperplane whose normal has a direction parallel to cc_1 . Let H^+ be a halfspace bounded by H containing c and not containing c_1 . By Lemma 1, it follows that there is a point $q \in H^+$ with $|qc| = r$. Consider two cases

- (i) If distance of $cc_1 \leq \varepsilon r/2$, then consider the triangle Δpcc_1 . By triangular inequality, $\|pc_1\| + \|cc_1\| \geq \|pc\|$ or, $r_1 \geq \|pc_1\| \geq (1 + \varepsilon/2)r$.
- (ii) If the distance between $cc_1 \geq \varepsilon r/2$, then consider the triangle Δc_1cq . This triangle is obtuse-angled at c . Thus $\|c_1q\| \geq \sqrt{(cc_1)^2 + r^2} \geq \sqrt{1 + \varepsilon^2}r \geq (1 + \varepsilon^2/k)r$ for some constant k .

\square

Since B_i, B_{i+1} differ in only one point p such that $p \in B_{i+1}$ and p not in ε -expansion of B_i , we can apply the lemma to conclude that the $r_{i+1} \geq (1 + \varepsilon^2)r_i$.

Update time Every update requires us to compute MEB of a set of $O(d/\varepsilon \log 1/\varepsilon)$ points. This implies that the update time is $O(d^d/\varepsilon^2 \log 1/\varepsilon)$. One can improve the update time by computing approximate MEB using algorithm in

Applications of ε -Blurred MEB ε -Blurred MEB can be used to obtain a $(\sqrt{2} + \varepsilon)$ -approximation of the MEB. It is known that the basis of MEB of a S can be used to answer the $\sqrt{2}$ -farthest point queries. Thus, knowing an ε -approximation of the MEB would have given us a $\sqrt{2} + \varepsilon$ approximation to the farthest point queries. We show that an ε blurred MEB is indeed sufficient to obtain a $\sqrt{2} + \varepsilon$ approximation to the farthest point queries.

$\sqrt{2} + \varepsilon$ -approximation of the MEB Let r_{opt} be the radius of the MEB of S . Since $K^* \subseteq S$, $r^* \leq r_{opt}$. Now we show that for any point $p \in S$, $\|c^*p\|$ is at most $(\sqrt{2} + \varepsilon)r^* \leq (\sqrt{2} + \varepsilon)r_{opt}$. This would then imply that a ball $B'(c^*, (\sqrt{2} + \varepsilon)r^*)$ encloses all points in S and is a $\sqrt{2} + \varepsilon$ approximation of MEB of S .

Lemma 5 For any point $p \in S$, $\|c^*p\| \leq (\sqrt{2} + \varepsilon)r^*$

Proof: Let p be the farthest point from c^* . Let $B_i \in B$ be the ball whose ε -expansion contains p' . Clearly

$$\|c^*p\| \leq \|c^*c_i\| + \|c_i p\| \leq \|c^*c_i\| + (1 + \varepsilon)r_i.$$

Let H be a hyperplane passing through c^* with normal parallel to $c_i c^*$. Let H^+ be the halfspace that doesn't contain c^* . From Lemma 1, there is a point p which is at a distance r_i from c^* . Thus

$$r^* \geq \sqrt{\|c_i c^*\|^2 + r_i^2}.$$

We know that $(\|c_i c^*\| + r_i) / \sqrt{\|c_i c^*\|^2 + r_i^2}$ is at most $\sqrt{2}$. This implies that

$$\|c^*p\| \leq \|c^*c_i\| + (1 + \varepsilon)r_i \leq (1 + \varepsilon)(\|c^*c_i\| + r_i) \leq (1 + \varepsilon)\sqrt{2}r^*.$$

□

Theorem 1 Given a stream of points S , there is an algorithm in the data stream model which computes a $\sqrt{2} + \varepsilon$ -approximation of the MEB with working space $O(d^2/\varepsilon^2 \log 1/\varepsilon)$.

$(\sqrt{2} + \varepsilon)$ farthest point queries We show that given any query point q , the farthest point among the $O(d/\varepsilon \log 1/\varepsilon)$ points present in C is a $\sqrt{2} + \varepsilon$ approximation of the farthest point from query q in S .

Lemma 6 Given a query point q , there is a point $k \in C$ such that, $\|qk\| \leq \|qp\| \leq (\sqrt{2} + \varepsilon)\|qk\|$.

Proof:

Consider any query point q . Let $p \in S$ be the farthest point from q . By the definition of ε -blurred MEB, there exists a ball B_i such that p is in the ε -expansion of B_i . Consider the basis K_i of B_i . Let H be a hyperplane passing through c_i with normal parallel to $c_i q$. Let the halfspace defined by H and not containing q be H^+ . By Lemma 1, there is a point $k \in K_i$ such that $k \in H^+$. The triangle $\Delta q c_i k$ is obtuse angled at c_i (By construction). Clearly $\|qp\| \leq \|q c_i\| + (1 + \varepsilon)r_i$ and $\|qk\| \geq \sqrt{\|q c_i\|^2 + r_i^2}$.

Since, $(\|q c_i\| + r_i) / \sqrt{\|q c_i\|^2 + r_i^2} \leq \sqrt{2}$

$$\|qp\| \leq \|q c_i\| + r_i(1 + \varepsilon),$$

$$\|qp\| \leq (1 + \varepsilon)(\|q c_i\| + r_i),$$

$$\|qp\| \leq (1 + \varepsilon)\sqrt{2}\|qk\|.$$

□

Theorem 2 There is an insertion only data structure of size $O(d^2/\varepsilon^2 \log 1/\varepsilon)$ which given any query point q , outputs the $(\sqrt{2} + \varepsilon)$ -farthest point query in time $O(d^3/\varepsilon^2 \log 1/\varepsilon)$.

By answering farthest point queries, we can also maintain a $\sqrt{2} + \varepsilon$ approximation of the diameter of a point set.

5 Other Results

We can generalize the idea of Blurred MEB to the blurred k -center. The idea in the algorithm is to maintain a set of possible solution to the k -center problem such that for each point $p \in S$, p is contained in the ε -expansion of some candidate solution. At the end of the stream, if we expand the final solution by a constant factor, we should be able to cover all the points. This leads to an $O(1)$ approximation. If we use Gonzalez sequence [16] to compute all intermediate solutions, we get an $O(1)$ factor approximation to the k center problem in general metric spaces.

We can also show that any streaming algorithm which approximates the MEB by a factor $c < \sqrt{2}$, in $\Omega(\log n)$ dimensional space requires $\Omega(n)$ working space. This result shows that our algorithm is almost optimal.

6 Conclusion

We studied various proximity problems in this project. Interesting relations between these problems were obtained. We also obtained a single data structure which can be used to develop an approximation algorithm in the data stream model for the MEB, CFP and the diameter problem. We also came up with a simple notion of blurred MEB which seems to be generalizable to other minimum enclosing objects (Blurred Minimum enclosing cylinder, Blurred minimum enclosing box etc.) and might lead to interesting approximation algorithms for computing them as well.

References

- [1] Mihai Badoiu, Sariel Har-Peled, Piotr Indyk Approximate Clustering via Core-Sets. *Proceedings, Symposium on Theory of Computation*, 2002, 250-257.
- [2] P.K. Agarwal, Sariel Har-Peled, K.R. Varadarajan Geometric Approximation via Core-Sets - Survey.
- [3] Mihai Badoiu, Kenneth L. Clarkson Smaller core-sets for balls. *Proceedings of ACM-SIAM symposium on Discrete algorithms*, 2003, 801-802
- [4] S. Muthukrishnan Data Streams: Algorithms and Applications. *Foundations and Trends in Theoretical Computer Science*, 2005, 1:2
- [5] A. Goel and P. Indyk and K. Varadarajan Reductions among high dimensional proximity problems. *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, 2001, 769-778
- [6] I. W. Tsang, J. T. Kwok, J. M. Zurada Generalized Core Vector Machines *IEEE Transactions on Neural Networks*, 2006, 1126–1140
- [7] T. Chan Faster core-set constructions and data-stream algorithms in fixed dimensions *Proc 20th ACM Symp. on Computational Geometry*, 2004 152–159.
- [8] P. Rai, H. Daume III and S. Venkatasubramanian Single Pass SVM using Minimum Enclosing Ball of Streaming Data *Learning Workshop, Snowbird*, 2008
- [9] P. Kumar, J.S.B Mitchell, E.A. Yildirim Approximate minimum enclosing balls in high dimensions using core-sets *Journal of Experimental Algorithms*, 2003

- [10] H. Zarrabi-Zadeh, T. M. Chan A Simple Streaming Algorithm for Minimum Enclosing Balls. *Canadian Conference on Computational Geometry*, 2006
- [11] P. Indyk Better algorithms for high-dimensional proximity problems via asymmetric embeddings *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, 2003, 539–545
- [12] W. L. Hsu, G. L. Nemhauser. “Easy and hard bottleneck location problems”. *Discrete Applied Mathematics*, 1, 1979, 209-216.
- [13] Hanan Samet. “Fundamentals of Multidimensional and Metric data structure”, Aug, 2006
- [14] T. Feder, D. H. Greene. Optimal Algorithms for Approximate Clustering. *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, 1988, 434–444.
- [15] M. Charikar, C. Chekuri, T. Feder, R. Motwani Incremental Clustering and Dynamic Information Retrieval *Proceedings of Annual ACM Symposium on Theory of Computing* 1997, 626–635.
- [16] T. E. Gonzalez Clustering to minimize maximum intercluster distance. *Theoretical Computer Science*, 38, 1985, 293–306