Probabilistic smallest enclosing ball in high dimensions via subgradient sampling*

Amer Krivošija[1] and Alexander Munteanu[2]

1 **Department of Computer Science, TU Dortmund, Germany**
amer.krivosija@tu-dortmund.de
2 **Department of Computer Science, TU Dortmund, Germany**
alexander.munteanu@tu-dortmund.de

Abstract

We study a variant of the median problem for a collection of point sets in high dimensions. This generalizes the geometric median as well as the (probabilistic) smallest enclosing ball (pSEB) problems. Our main objective and motivation is to improve the previously best algorithm for the pSEB problem by reducing its exponential dependence on the dimension to linear. This is achieved via a novel combination of sampling techniques for clustering problems in metric spaces with the framework of stochastic subgradient descent. As a result, the algorithm becomes applicable to shape fitting problems in Hilbert spaces of unbounded dimension via kernel functions. We present an exemplary application by extending the support vector data description (SVDD) shape fitting method to the probabilistic case. This is done by simulating the pSEB algorithm implicitly in the feature space induced by the kernel function.

1 Introduction

The (probabilistic) smallest enclosing ball (pSEB) problem in $\mathbb{R}^d$ is to find a center that minimizes the (expected) maximum distance to the input points (see Definition 3.1). It occurs often as a building block for complex data analysis and machine learning tasks like estimating the support of high dimensional distributions, outlier detection, novelty detection, classification and robot gathering [5, 15, 16, 18]. It is thus very important to develop efficient algorithms for the base problem. This involves reducing the number of points but also keeping the dependence on the dimension as low as possible. We study both objectives and focus on a small dependence on the dimension. This is motivated as follows. Kernel methods are a common technique in machine learning. These methods implicitly project the d-dimensional input data into much larger dimension D where simple linear classifiers or spherical data fitting methods can be applied to obtain a non-linear separation or non-convex shapes in the original d-dimensional space. The efficiency of kernel methods is usually not harmed since inner products and thus distances in the D-dimensional space can be evaluated in $O(d)$ time.

In some cases, however, a proper approximation relying on sampling and discretizing the ambient solution space may require a polynomial or even exponential dependence on D. The algorithm of Munteanu et al. [11] is the only fully polynomial time approximation scheme (FPTAS) and the fastest algorithm to date for the pSEB problem in fixed dimension. However, it suffers from the stated problems. In particular, the number of realizations sampled by their algorithm had a linear dependence on D stemming from a ball-cover decomposition of the solution space. The actual algorithm made a brute force evaluation (on the sample)

* The full version of this paper will appear at SoCG 2019. This work was supported by the German Science Foundation (DFG) Collaborative Research Center SFB 876 "Providing Information by Resource-Constrained Analysis", projects A2 and C4.

35th European Workshop on Computational Geometry, Utrecht, The Netherlands, March 18–20, 2019.
This is an extended abstract of a presentation given at EuroCG'19. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

of all centers in a grid of exponential size in D. This is prohibitive in the setting of kernel methods since the implicit feature space may have infinite dimension. Even if it is possible to exploit the up to n-dimensional subspace spanned by n points in infinite dimensions, we would still have $D = n \gg d$ leading to exponential time algorithms. To make the pSEB algorithm viable in the context of kernel methods and generally in high dimensions, it is desirable to reduce the dependence on the dimension to a small polynomial occurring only in evaluations of inner products and distances of low dimensional vectors.

Related work: The study of *probabilistic clustering problems* was initiated by Cormode and McGregor [7]. They developed approximation algorithms for the probabilistic settings of k-means, k-median as well as k-center clustering. Munteanu et al. [11] gave the first fully polynomial time $(1+\varepsilon)$-approximation scheme (FPTAS) for the pSEB problem, in fixed dimensions, in time $O(nd/\varepsilon^{O(1)} + 1/\varepsilon^{O(d)})$. We reduce its exponential dependence on d to linear, using sampling techniques. The stochastic subgradient descent from convex optimization [6, 12] is a quite popular and often only implicitly used technique in the core-set literature [2, 4, 10]. Indyk and Thorup [8, 17] showed that a uniform sample of size $O(\log n/\varepsilon^2)$ is sufficient to approximate the discrete metric 1-median within a factor of $(1+\varepsilon)$. We adapt these ideas to find a $(1+\varepsilon)$-approximation to the best center in our setting.

Kernel functions simulate an inner product space in large or even unbounded dimensions but can be evaluated via simple low dimensional vector operations in the original dimension of input points [14]. This enables simple spherical shape fitting via a smallest enclosing ball algorithm in the high dimensional feature space, which implicitly defines a more complex and even non-convex shape in the original space. The smallest enclosing ball problem in kernel spaces is well-known as the support vector data description (SVDD) problem [16, 18].

1.1 Contributions and outline

We extend the geometric median in Euclidean space to the more general problem of finding a center $c \in \mathbb{R}^d$ that minimizes the sum of maximum distances to sets of points in a given collection of N point sets. We show how to solve this problem via estimation and sampling techniques combined with a stochastic subgradient descent algorithm, see Theorem 2.2.

The elements in the collection are sets of n points in $\mathbb{R}^d$. In [11] they were summarized via strong coresets of size $1/\varepsilon^{\Theta(d)}$. This is not an option in high dimensions. Reviewing the techniques of [1] we show that no reduction below $\min\{n, \exp(d^{1/3})\}$ is possible unless sacrificing an additional approximation factor of roughly $\sqrt{2}$, see Theorem 2.3. However, it is possible to achieve roughly a $(\sqrt{2}+\varepsilon)$-approximation in streaming via the *blurred-ball-cover* [1] of size $O(1/\varepsilon^3 \cdot \log 1/\varepsilon)$, and in an off-line setting via weak coresets [2, 3] of size $O(1/\varepsilon)$.

We show in Theorem 3.2 how Theorem 2.2 improves the previously best FPTAS for the pSEB problem from $O(dn/\varepsilon^3 \cdot \log 1/\varepsilon + 1/\varepsilon^{O(d)})$ to $O(dn/\varepsilon^4 \cdot \log^2 1/\varepsilon)$. In particular the dependence on the dimension d is reduced from exponential to linear and more precisely occurs only in distance evaluations between points in d-dimensional Euclidean space, but not in the number of sampled points nor in the number of candidate centers to evaluate.

This enables working in very high D-dimensional Hilbert spaces whose inner products and distances are given implicitly via positive semidefinite kernel functions. These functions can be evaluated in $O(d)$ time although D is large or even unbounded. We extend the well-known SVDD method to the probabilistic case, see Theorem 3.3.

1.2 General notation

We denote the set of positive integers up to $n \in \mathbb{N}$ by $[n] = \{1, \ldots, n\}$. For any convex function $f\colon \mathbb{R}^d \to \mathbb{R}$ we denote by $\partial f(x) = \left\{g \in \mathbb{R}^d \mid \forall y \in \mathbb{R}^d\colon f(x) - f(y) \leq \langle g, x - y\rangle\right\}$ the set of subgradients of f at x. We assume the error parameter satisfies $0 < \varepsilon < 1/9$.

2 A generalized median problem

The pSEB problem can be reduced to two different types of 1-median problems [11]. One of them is defined on the set of all non-empty locations in $\mathbb{R}^d$ where probabilistic points may appear, equipped with the Euclidean distance. The other is defined on the collection of all possible realizations of probabilistic point sets, and the distance measure between a center $c \in \mathbb{R}^d$ and a realization $P_i \subset \mathbb{R}^d$ is $m(c, P_i) = \max_{p \in P_i} \|c - p\|$. We state a generalized median problem that we call the *set median problem* and covers both of these cases.

▶ **Definition 2.1** (set median problem)**.** Let $\mathcal{P} = \{P_1, \ldots, P_N\}$ be a family of finite non-empty sets where $\forall i \in [N]\colon P_i \subset \mathbb{R}^d$ and $n = \max\{|P_i| \mid i \in [N]\}$. The set median problem on $\mathcal{P}$ consists in finding a center $c \in \mathbb{R}^d$ that minimizes the cost function

$$f(c) = \sum_{i=1}^{N} m(c, P_i).$$

Note that in case of singleton sets, the set median problem is equivalent to the well-known Fermat-Weber problem (a.k.a. 1-median or geometric median). Also, for $N = 1$ it coincides with the smallest enclosing ball or 1-center problem. For both of these problems there are known algorithms based on the subgradient method from convex optimization [2, 6].

The Lipschitz constant of the function f can be bounded by N. We want to minimize f via the subgradient method, see [12]. For that sake we need to compute a subgradient $g(c_i) \in \partial f(c_i)$ at the current center c_i. The subgradient computation takes $O(dnN)$ time to calculate, since in each of the N terms of the sum we maximize over $|P_i| \leq n$ distances in d dimensions to find a point in P_i that is furthest away from c. To remove the dependence on N we replace the exact subgradient $g(c_i)$ by a uniform sample of only one nonzero term which points into the right direction in expectation. Then we can adapt the deterministic subgradient method from [12] using the random unbiased subgradient in such a way that the result is in expectation a $(1 + \varepsilon)$-approximation to the optimal solution. Given an initial center c_0, a fixed step size s, and a number of iterations ℓ, our algorithm iteratively picks a set $P_j \in \mathcal{P}$ uniformly at random and chooses a point $p_j \in P_j$ that attains the maximum distance to the current center. This point is used to compute an approximate subgradient. The algorithm finally outputs the best center found in all iterations.

To bound in expectation the quality of the output of our algorithm to be a $(1 + \varepsilon)$-approximation, we choose the values of parameters c_0, s, and ℓ in an appropriate way. It suffices to run our algorithm for $\ell \in O(1/\varepsilon^2)$ iterations, and to choose c_0 to be an arbitrary point in a randomly chosen input set from $\mathcal{P}$. We estimate the average cost on a sample of size $1/\varepsilon$ [9], which bounds the value of s. It remains to describe how to find the best center out of all iterations of our algorithm efficiently. We cannot do this exactly since evaluating the cost even for one single center takes time $O(dnN)$. However, we use a sampling technique [8, 17] (cf. the related work above), adapted here to work in our setting, to find a point that is a $(1 + \varepsilon)$-approximation of the best center in a finite set of candidate centers. The main difference is that in the original work the set of input points and the set of candidate solutions are identical. In our setting, however, we have that the collection of input sets and

the set of candidate solutions may be completely distinct. Putting all pieces together we have the following Theorem.

▶ **Theorem 2.2.** *Consider an input $\mathcal{P} = \{P_1, \ldots, P_N\}$, where for every $i \in [N]$ we have $P_i \subset \mathbb{R}^d$ and $n = \max\{|P_i| \mid i \in [N]\}$. There exists an algorithm that computes a center $\tilde{c}$ that is with constant probability a $(1+\varepsilon)$-approximation to the optimal solution c^* of the set median problem (see Definition 2.1). Its running time is $O(dn/\varepsilon^4 \cdot \log^2 1/\varepsilon)$.*

The removal of the linear dependence on n for the maximum distance computations was achieved in [11] via a grid based strong coreset of size $1/\varepsilon^{\Theta(d)}$. However, here we focus on reducing the dependence on d, and exponential is not an option if we want to work in high dimensions. It turns out that without introducing an exponential dependence on d, we would have to lose a constant approximation factor. We adapted the techniques of [1] to show that no small data structure can exist for answering maximum distance queries to within a factor of less than roughly $\sqrt{2}$. In comparison to the previous results, it is not limited to the streaming setting, as in [1], and it is not restricted to subsets of the input, as in [13].

▶ **Theorem 2.3.** *Any data structure that, with probability at least $2/3$, α-approximates maximum distance queries on a set $S \subset \mathbb{R}^d$ of size $|S| = n$, for $\alpha < \sqrt{2}\left(1 - 2/d^{1/3}\right)$, requires $\Omega\left(\min\{n, \exp\left(d^{1/3}\right)\}\right)$ bits of storage.*

3 Applications

3.1 Probabilistic smallest enclosing ball

We apply our result to the pSEB problem, as given in [11]. In such a setting, the input is a set $\mathcal{D} = \{D_1, \ldots, D_n\}$ of n discrete and independent probability distributions. The i-th distribution D_i is defined over a set of z possible locations $q_{i,j} \in \mathbb{R}^d \cup \{\perp\}$, for $j \in [z]$, where $\perp$ indicates that the i-th point is not present in a sampled set, i.e., $q_{i,j} = \perp \Leftrightarrow \{q_{i,j}\} = \emptyset$. We call these points *probabilistic points*. Each location $q_{i,j}$ is associated with the probability $p_{i,j}$, such that $\sum_{j=1}^{z} p_{i,j} = 1$, for every $i \in [n]$. Thus the probabilistic points can be considered as independent random variables X_i. A probabilistic set X of probabilistic points is also a random variable.

▶ **Definition 3.1.** ([11]) Let $\mathcal{D}$ be a set of n discrete distributions, where each distribution is defined over z locations in $\mathbb{R}^d \cup \{\perp\}$. The pSEB problem is to find a center $c^* \in \mathbb{R}^d$ that minimizes the expected smallest enclosing ball cost: $c^* \in \operatorname{argmin}_{c \in \mathbb{R}^d} \mathbb{E}\left[m(c, X)\right]$, where the expectation is taken over the randomness of $X \sim \mathcal{D}$.

Our pSEB algorithm adapts the framework of [11], but plugging in Theorem 2.2 it differs mainly in three points. First, the number of samples had a dependence on d hidden in the O-notation. This is not the case any more. Second, the sampled realizations are not sketched via coresets of size $1/\varepsilon^{\Theta(d)}$ any more. Third, the running time of the actual optimization task is reduced instead of an exhaustive grid search.

▶ **Theorem 3.2.** *Let $\mathcal{D}$ be a set of n discrete distributions, where each distribution is defined over z locations in $\mathbb{R}^d \cup \{\perp\}$. Let $\tilde{c} \in \mathbb{R}^d$ be the output of our pSEB algorithm on input $\mathcal{D}$. Let $\varepsilon < 1/9$. With constant probability $\tilde{c}$ is a $(1+\varepsilon)$-approximation for the pSEB problem,*

$$\mathbb{E}_X\left[m(\tilde{c}, X)\right] \leq (1+\varepsilon) \min_{c \in \mathbb{R}^d} \mathbb{E}_X\left[m(c, X)\right].$$

The running time of our pSEB algorithm is $O(dn \cdot (z/\varepsilon^3 \cdot \log 1/\varepsilon + 1/\varepsilon^4 \cdot \log^2 1/\varepsilon))$.

Comparing to the result of [11], the running time is reduced from $O(dnz/\varepsilon^{O(1)} + 1/\varepsilon^{O(d)})$ to $O(dnz/\varepsilon^{O(1)})$. The factor of d plays a role only in computations of distances between two points in $\mathbb{R}^d$. Further the sample size and the number of centers that need to be evaluated do not depend on the dimension d any more. This is crucial in the following.

3.2 Probabilistic support vector data description (pSVDD)

We consider the SVDD problem, i.e., the SEB problem in kernel spaces, and show how to extend it to its probabilistic version. Let $K\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a positive semidefinite kernel function with feature map $\varphi\colon \mathbb{R}^d \to \mathcal{H}$, where $\mathcal{H}$ is a high dimensional Hilbert space, say $\mathbb{R}^D$, where $D \gg d$ [14].

▶ **Theorem 3.3.** *Let $\mathcal{D}$ be a set of n discrete distributions, where each distribution is defined over z locations in $\mathbb{R}^d \cup \{\perp\}$. There exists an algorithm that implicitly computes $\tilde{c} \in \mathcal{H}$ that with constant probability is a $(1+\varepsilon)$-approximation for the probabilistic SVDD problem. It is*

$$\mathbb{E}_X\left[m(\tilde{c}, \varphi(X))\right] \leq (1+\varepsilon)\min_{c\in\mathcal{H}} \mathbb{E}_X\left[m(c, \varphi(X))\right],$$

where the expectation is taken over the randomness of $X \sim \mathcal{D}$, and $\varphi(X) = \{\varphi(x_i) \mid x_i \in X\}$. The running time of the algorithm is $O(dn \cdot \left(z/\varepsilon^3 \cdot \log 1/\varepsilon + 1/\varepsilon^8 \cdot \log^2 1/\varepsilon\right))$.

References

1 P. K. Agarwal and R. Sharathkumar. Streaming algorithms for extent problems in high dimensions. *Algorithmica*, 72(1):83–98, 2015.

2 M. Bădoiu and K. L. Clarkson. Smaller core-sets for balls. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 801–802, 2003.

3 M. Bădoiu and K. L. Clarkson. Optimal core-sets for balls. *Comput. Geom*, 40(1):14–22, 2008.

4 M. Bădoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 250–257, 2002.

5 M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Distributed computing by mobile robots: Gathering. *SIAM J. Comput.*, 41(4):829–879, 2012.

6 M. B. Cohen, Y. T. Lee, G. L. Miller, J. Pachocki, and A. Sidford. Geometric median in nearly linear time. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 9–21, 2016.

7 G. Cormode and A. McGregor. Approximation algorithms for clustering uncertain data. In *Proceedings of the 27th Symposium on Principles of Database Systems (PODS)*, pages 191–200, 2008.

8 P. Indyk. *High-dimensional Computational Geometry.* PhD thesis, Stanford University, 2000.

9 A. Kumar, Y. Sabharwal, and S. Sen. Linear-time approximation schemes for clustering problems in any dimensions. *J. ACM*, 57(2):5:1–5:32, 2010.

10 A. Munteanu and C. Schwiegelshohn. Coresets-methods and history: A theoreticians design pattern for approximation and streaming algorithms. *KI*, 32(1):37–53, 2018.

11 A. Munteanu, C. Sohler, and D. Feldman. Smallest enclosing ball for probabilistic data. In *30th Annual Symposium on Computational Geometry (SoCG)*, pages 214–223, 2014.

12 Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course.* Applied Optimization. Springer, New York, 2004.

13 R. Pagh, F. Silvestri, J. Sivertsen, and M. Skala. Approximate furthest neighbor with application to annulus query. *Inf. Syst.*, 64:152–162, 2017.

14 B. Schölkopf and A. J. Smola. *Learning with Kernels: support vector machines, regularization, optimization, and beyond.* Adaptive computation and machine learning series. MIT Press, 2002.

15 M. Stolpe, K. Bhaduri, K. Das, and K. Morik. Anomaly detection in vertically partitioned data by distributed core vector machines. In *Proceedings of Machine Learning and Knowledge Discovery in Databases - European Conference, (ECML/PKDD) Part III*, pages 321–336, 2013.

16 D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54 (1):45–66, 2004.

17 M. Thorup. Quick k-median, k-center, and facility location for sparse graphs. *SIAM J. Comput.*, 34(2):405–432, 2005.

18 I. W. Tsang, J. T. Kwok, and P. Cheung. Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research*, 6:363–392, 2005.