

Probabilistic Smallest Enclosing Ball in High Dimensions via Subgradient Sampling

Amer Krivošija

Department of Computer Science, TU Dortmund, Germany
amer.krivosija@tu-dortmund.de

Alexander Munteanu

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.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation → Streaming, sublinear and near linear time algorithms; Theory of computation → Computational geometry

Keywords and phrases geometric median, convex optimization, smallest enclosing ball, probabilistic data, support vector data description, kernel methods

Digital Object Identifier 10.4230/LIPIcs.SoCG.2019.47

Related Version A full version of the paper is available at <https://arxiv.org/abs/1902.10966>.

Funding 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.

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 13). 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 [9, 27, 28, 30]. It is thus very important to develop highly efficient approximation algorithms for the base problem. This involves reducing the number of points but also keeping the dependence on the dimension as low as possible. Both objectives will be studied in this paper. We will 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. Despite the large dimension $D \gg d$, most important kernels, and thus inner products and distances in the D -dimensional space, can be evaluated in $O(d)$ time [25].



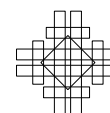
© Amer Krivošija and Alexander Munteanu;
licensed under Creative Commons License CC-BY
35th International Symposium on Computational Geometry (SoCG 2019).

Editors: Gill Barequet and Yusu Wang; Article No. 47; pp. 47:1–47:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



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. [22] 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) 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 probabilistic smallest enclosing ball algorithm viable in the context of kernel methods and generally in high dimensions, it is highly desirable to reduce the dependence on the dimension to a small polynomial occurring only in evaluations of inner products and distances between two (low dimensional) vectors.

1.1 Related work

Probabilistic smallest enclosing ball. The study of *probabilistic clustering problems* was initiated by Cormode and McGregor [11]. They developed approximation algorithms for the probabilistic settings of k -means, k -median as well as k -center clustering. For the metric 1-center clustering problem their results are bi-criteria $O(1)$ -approximation algorithms with a blow-up on the number of centers. Guha and Munagala [14] improved the previous work by giving $O(1)$ -approximations that preserve the number of centers. Munteanu et al. [22] gave the first fully polynomial time $(1 + \varepsilon)$ -approximation scheme (FPTAS) for the probabilistic Euclidean 1-center, i.e., the probabilistic smallest enclosing ball problem, in fixed dimensions. The algorithm runs in linear time for sampling a constant number of realizations. Solving the subsampled problem takes only constant time, though exponential in the dimension. This yields a total running time of roughly $O(nd/\varepsilon^{O(1)} + 1/\varepsilon^{O(d)})$. Based on ε -kernels for probabilistic data [16], Huang and Li [15] generalized the $(1 + \varepsilon)$ -approximation of [22] to a polynomial time approximation scheme (PTAS) for Euclidean k -center in \mathbb{R}^d for fixed constants k and d . We note that the running time of the algorithm in [15] grows as a double exponential function of the dimension and it is unclear how to reduce this. We thus base our work on the FPTAS of [22] and reduce its exponential dependence to linear.

Sampling techniques for 1-median. The work of [22] showed a reduction from the probabilistic smallest enclosing ball to 1-median problems on (*near*)-*metric* spaces. We thus review relevant results on sampling techniques for metric 1-median. Bădoiu et al. [7] showed that with constant probability, the span of a uniform sample of a constant number of input points contains a $(1 + \varepsilon)$ -approximation for the 1-median. This was used to construct a set of candidate solutions of size $O(2^{1/\varepsilon^{O(1)}} \log n)$. A more refined estimation procedure based again on a small uniform sample was used by Kumar et al. [19] to reduce this to $O(2^{1/\varepsilon^{O(1)}})$. Indyk and Thorup [17, 29] showed that a uniform sample of size $O(\log n/\varepsilon^2)$ is sufficient to approximate the discrete metric 1-median on n points within a factor of $(1 + \varepsilon)$. Ackermann et al. [1] showed how this argument can be adapted to doubling spaces which include the continuous Euclidean space. We adapt these ideas to find a $(1 + \varepsilon)$ -approximation to the best center in our setting.

Stochastic subgradient descent. One quite popular and often only implicitly used technique in the coresets literature is derived from convex optimization, see [21]. One of the first results of that kind is given in the uniform sampling algorithm of Bădoiu et al. [7] for 1-median stated above. In each iteration a single point is sampled uniformly. With high

probability moving the current center towards that point for a carefully chosen step size improves the solution. Each step can be seen as taking a descent towards a uniformly random direction from the subgradient which equals roughly the sum of directions to all points. Another example is the coresets construction for the smallest enclosing ball problem by Bădoiu and Clarkson [5], where the next point included in the coresets is the one maximizing the distance to the current best center. The direction taken towards that point is again related to the subgradient of the objective function at the current center position. The authors also gave a more explicit application of subgradient descent to the problem with a slightly larger number of iterations. More recently, Cohen et al. [10] developed one of the fastest $(1 + \varepsilon)$ -approximation algorithms to date for the geometric median problem via stochastic subgradient methods.

Kernel methods in machine learning. 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 [26]. 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) by Tax and Duin [28]. A more subtle connection between (the dual formulations of) several kernel based methods in machine learning and the smallest enclosing ball problem was established by Tsang et al. [30].

1.2 Our contributions and outline

- We extend in Section 2 the geometric median in Euclidean space to the more general set median problem. It consists 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.
- The elements in the collection are sets of up to n points in \mathbb{R}^d . We discuss in Section 2.1 the possibility of further reducing their size. In the previous work [22] they were summarized via strong coresets of size $1/\varepsilon^{\Theta(d)}$ for constant dimension d . This is not an option in high dimensions where, e.g. $d \approx n$. Reviewing the techniques of Agarwal and Sharathkumar [2] we can show that no reduction below $\min\{n, \exp(d^{1/3})\}$ is possible unless one is willing to sacrifice an additional approximation factor of roughly $\sqrt{2}$. However, we discuss the possibility to achieve roughly a factor $(\sqrt{2} + \varepsilon)$ -approximation in streaming via the *blurred-ball-cover* [2] of size $O(1/\varepsilon^3 \cdot \log 1/\varepsilon)$, and in an off-line setting via weak coresets [5, 6] of size $O(1/\varepsilon)$.
- We show in Section 3.1 how this improves the previously best FPTAS for the probabilistic smallest enclosing ball 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 notably 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 in Section 3.2 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 depending on the kernel function. As an example we extend the well-known support vector data description (SVDD) method to the probabilistic case. SVDD is equivalent to the smallest enclosing ball problem in the implicit high-dimensional feature space.

Please find the missing proofs in the full version of this paper.

1.3 General notation

We denote the set of positive integers up to $n \in \mathbb{N}$ by $[n] = \{1, \dots, n\}$. For any vectors $x, y \in \mathbb{R}^d$ we denote their inner product by $\langle x, y \rangle = x^T y = \sum_{i=1}^d x_i y_i$ and the Euclidean norm by $\|x\| = \sqrt{\langle x, x \rangle} = (\sum_{i=1}^d x_i^2)^{1/2}$. The Cauchy-Schwarz inequality (CSI) states that $\langle x, y \rangle \leq \|x\| \|y\|$. For any convex function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ we denote by $\partial f(x) = \{g \in \mathbb{R}^d \mid \forall y \in \mathbb{R}^d: f(x) - f(y) \leq \langle g, x - y \rangle\}$ the subdifferential, i.e., the set of subgradients of f at x . For any event \mathcal{E} let the indicator function be $\mathbb{1}_{\mathcal{E}} = 1$ if \mathcal{E} happens and 0 otherwise. We denote by $B(c, r) = \{x \mid \|c - x\| \leq r\}$ the Euclidean ball centered at $c \in \mathbb{R}^d$ with radius $r \geq 0$. We assume the error parameter satisfies $0 < \varepsilon < 1/9$. All our results hold with constant probability, say $1/8$, which can be amplified to arbitrary $1 - \eta$, $0 < \eta < 1$, by running $O(\log 1/\eta)$ independent repetitions and returning the minimum found.

2 A generalized median problem

The probabilistic smallest enclosing ball (pSEB) problem is to find a center that minimizes the expected maximum distance to points drawn from the input distributions (see Section 3.1 and Definition 13), and it can be reduced to two different types of 1-median problems [22]. 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 the maximum distance between the center and any of the realized points, i.e., $\max_{p \in P_i} \|c - p\|$.

We begin our studies with a generalized median problem that we call the *set median problem* and covers both of these cases.

► **Definition 1** (set median problem). *Let $\mathcal{P} = \{P_1, \dots, P_N\}$ be a family of finite non-empty sets where $\forall i \in [N]: P_i \subset \mathbb{R}^d$ 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*

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

where $m(c, P_i) = \max_{p \in P_i} \|c - p\|$.

It was noted in [22] that the distance measure m between any two sets $A, B \subset \mathbb{R}^d$ defined by the maximum distance $m(A, B) = \max_{a \in A, b \in B} \|a - b\|$ is not a metric since for any non-singleton set $C \subset \mathbb{R}^d$ it holds that $m(C, C) > 0$. We stress here that we consider only cases where, as in Definition 1, one of the sets $A = \{c\}$ is a singleton, and $B = P$ is an arbitrary non-empty set of points from \mathbb{R}^d . In order to directly apply results from the theory of metric spaces we thus simply define $m(A, B) = 0$ whenever $A = B$.

► **Lemma 2.** *Let \mathcal{X} be the set of all finite non-empty subsets of \mathbb{R}^d . We define*

$$m(A, B) = \begin{cases} \max_{a \in A, b \in B} \|a - b\| & \text{if } A \neq B \\ 0 & \text{if } A = B \end{cases}$$

for any $A, B \in \mathcal{X}$. Then (\mathcal{X}, m) is a metric space.

Note that in case of singleton sets, the set median problem in Definition 1 is equivalent to the well-known Fermat-Weber problem also known as 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. Bádoiu and Clarkson [5] gave a simple algorithm for approximating the 1-center within $(1 + \varepsilon)$ -error. Starting from an initial center, it is iteratively moved a little towards the input point that is furthest away. Note that a suitable subgradient at the current center points exactly into the opposite direction. More precisely if $q \in P$ is a point that is furthest away from the current center c then $(c - q) / \|c - q\| \in \partial \max_{p \in P} \|c - p\|$. The algorithm can thus be interpreted as a subgradient descent minimizing $\max_{p \in P} \|c - p\|$. Weiszfeld's algorithm [31, 32] was the first that solved the 1-median problem within additive $O(\varepsilon)$ -error in $O(1/\varepsilon)$ subgradient iterations. And indeed one of the fastest algorithms to approximate the geometric median problem to date relies on a stochastic subgradient descent [10]. The crucial ingredients to turn the additive error to a relative error are finding a suitable starting point that achieves a constant approximation and estimating its initial distance to the optimal solution. Another important step is a bound on the Lipschitz constant of the cost function, see also [3]. We will generalize these approaches to minimizing the cost function f of the set median problem from Definition 1.

First note that f is a convex function which implies that we can apply the theory of convex analysis and optimization. In particular, it implies that the subdifferential $\partial f(c)$ is non-empty for any center $c \in \mathbb{R}^d$ and c is locally optimal if and only if $0 \in \partial f(c)$. Moreover, any local optimum is also globally optimal by convexity [4, 23]. This implies that if we find a $(1 + \varepsilon)$ -approximation to a local minimum of the convex function, the convexity implies that it is a $(1 + \varepsilon)$ -approximation to the global minimum as well.

To see that f is convex, note that the Euclidean norm is a convex function. Therefore the Euclidean distance to some fixed point is a convex function since every translation of a convex function is convex. The maximum of convex functions is a convex function and finally the sum of convex functions is again convex. We prove this claim for completeness.

► **Lemma 3.** *The objective function f of the set median problem (see Definition 1) is convex.*

Next we bound the Lipschitz constant of the function f by N . We may get a better bound if we limit the domain of f to a ball of small radius centered at the optimal solution, but the Lipschitz constant cannot be bounded by $o(N)$ in general. We will see later how we can remove the dependence on N .

► **Lemma 4.** *The objective function f of the set median problem (see Definition 1) is N -Lipschitz continuous, i.e., $|f(x) - f(y)| \leq N \cdot \|x - y\|$ for all $x, y \in \mathbb{R}^d$.*

We want to minimize f via the subgradient method, see [23]. For that sake we need to compute a subgradient $g(c_i) \in \partial f(c_i)$ at the current center c_i . To this end we prove the following lemma.

► **Lemma 5.** *Let $c_i \in \mathbb{R}^d$ be any center. For each set $P_j \in \mathcal{P}$, let $p_j \in P_j$ be a point with $\|c_i - p_j\| = m(c_i, P_j)$. We have*

$$g(c_i) = \sum_{j=1}^N \frac{c_i - p_j}{\|c_i - p_j\|} \cdot \mathbf{1}_{c_i \neq p_j} \in \partial f(c_i),$$

i.e., $g(c_i)$ is a valid subgradient of f at c_i .

For brevity of presentation we omit the indicator function in any use of the above Lemma in the remainder of this paper and simply define $(c_i - p_j) / \|c_i - p_j\| = 0$ whenever $c_i = p_j$.

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 . We are going to discuss the possibility of reducing n later. For

now we focus on removing the dependence on N . To this end we would like to replace the exact subgradient $g(c_i)$ by a uniform sample of only one nonzero term which points into the right direction in expectation. We formalize this in the following lemma.

► **Lemma 6.** *Let $c_i \in \mathbb{R}^d$ be any fixed center. For each set $P_j \in \mathcal{P}$, let $p_j \in P_j$ be a point with $\|c_i - p_j\| = m(c_i, P_j)$. Let $\tilde{g}(c_i)$ be a random vector that takes the value $\tilde{g}(c_i) = (c_i - p_j) / \|c_i - p_j\|$ for $j \in [N]$ with probability $1/N$ each. Then $\mathbb{E}[\|\tilde{g}(c_i)\|^2] \leq 1$ and $\mathbb{E}[\tilde{g}(c_i)] = g(c_i)/N$, where $g(c_i) \in \partial f(c_i)$ is the subgradient given in Lemma 5.*

We can now adapt the deterministic subgradient method from [23] using the random unbiased subgradient of Lemma 6 in such a way that the result is in expectation a $(1 + \varepsilon)$ -approximation to the optimal solution. This method is presented in Algorithm 1. Given an initial center c_0 , a fixed step size s , and a number of iterations ℓ , the 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 via Lemma 6. The Algorithm finally outputs the best center found in all iterations.

Algorithm 1: Stochastic subgradient method.

Data: A family of non-empty sets $\mathcal{P} = \{P_1, \dots, P_N\}$, where $P_i \subset \mathbb{R}^d$

Result: A center $\tilde{c} \in \mathbb{R}^d$

- 1 Determine an initial center c_0
 - 2 Fix a step size s
 - 3 Fix the number of iterations ℓ
 - 4 **for** $i \leftarrow 1$ **to** ℓ **do**
 - 5 Choose an index $j \in [N]$ uniformly at random and compute $\tilde{g}(c_{i-1})$, cf. Lemma 6
 - 6 $c_i = c_{i-1} - s \cdot \tilde{g}(c_{i-1})$
 - 7 **return** $\tilde{c} \in \operatorname{argmin}_{c \in \{c_i | i=0, \dots, \ell\}} f(c)$
-

The following theorem bounds in expectation the quality of the output that our subgradient algorithm returns. It is a probabilistic adaptation of a result in convex optimization [23].

► **Theorem 7.** *Consider Algorithm 1 on input $\mathcal{P} = \{P_1, \dots, P_N\}$ for the set median problem with objective function f , see Definition 1. Let $c^* \in \operatorname{argmin}_{c \in \mathbb{R}^d} f(c)$. Let $R = \|c_0 - c^*\|$. Then*

$$\mathbb{E}_{\tilde{c}} [f(\tilde{c}) - f(c^*)] \leq N \cdot \frac{R^2 + (\ell + 1)s^2}{2(\ell + 1)s},$$

where the expectation is taken over the random variable $\tilde{c} \in \operatorname{argmin}_{c \in \{c_i | i=0, \dots, \ell\}} f(c)$, i.e., the output of Algorithm 1.

Our aim now is to choose the parameters ℓ , s , and c_0 of Algorithm 1 in such a way that the bound given in Theorem 7 becomes at most $\varepsilon f(c^*)$. To this end we can choose the initial center c_0 and bound its initial distance $R = \|c_0 - c^*\|$ proportional to the average cost $O(f(c^*)/N)$ with constant probability using a simple Markov argument.

► **Lemma 8.** *Choose a set P from $\mathcal{P} = \{P_1, \dots, P_N\}$ uniformly at random and let c_0 be an arbitrary point of P . Then for any constant $0 < \delta_1 < 1$ it holds that $R = \|c_0 - c^*\| \leq f(c^*)/(\delta_1 N)$ with probability at least $1 - \delta_1$.*

Assume that we know the value of R , and set the step size to $s = R/\sqrt{\ell+1}$, then Theorem 7 and Lemma 8 imply that for some constant C

$$\mathbb{E}_{\tilde{c}} [f(\tilde{c}) - f(c^*)] \leq \frac{NR}{\sqrt{\ell+1}} \leq \frac{Cf(c^*)}{\sqrt{\ell+1}}$$

holds with constant probability. We thus only need to run the algorithm for $\ell \in O(1/\varepsilon^2)$ iterations to get within $\varepsilon f(c^*)$ error. But choosing this particular step size requires to know the optimal center in advance. To get around this, we attempt to estimate the average cost. More formally, we are interested in a constant factor approximation of $f(c^*)/N$. It turns out that we can do this based on a small sample of the input sets unless our initial center is already a good approximation. But in the latter case we do not care about all the subsequent steps or step sizes, since we are already done after the initialization. The proof technique is originally from [19] and is adapted here to work in our setting with sets of points.

► **Lemma 9.** *There exists an algorithm that based on a sample $\mathcal{S} \subseteq \mathcal{P}$ of size $|\mathcal{S}| = 1/\varepsilon$ returns an estimate \tilde{R} and an initial center c_0 in time $O(dn/\varepsilon)$ such that with constant probability one of the following holds:*

- a) $\varepsilon f(c^*)/N \leq \tilde{R} \leq (2/\varepsilon^3) \cdot f(c^*)/N$ and $\|c_0 - c^*\| \leq 8f(c^*)/N$;
- b) $\|c_0 - c^*\| \leq 4\varepsilon f(c^*)/N$.

Lemma 9 has the following consequence. Either the initial center c_0 is already a $(1+4\varepsilon)$ -approximation, in which case we are done. Or we are close enough to an optimal solution and have a good estimate on the step size to find a $(1+4\varepsilon)$ -approximation in a constant number of iterations.

Another issue that we need to take care of, is finding the best center in the last line of Algorithm 1 efficiently. We cannot do this exactly since evaluating the cost even for one single center takes time $O(dnN)$. However, we can find a point that is a $(1+\varepsilon)$ -approximation of the best center in a finite set of candidate centers using a result from the theory of discrete metric spaces.

To this end we can apply our next theorem which is originally due to Indyk and Thorup in [17, 29] and adapted here to work in our setting. 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 and the distance measure is the maximum distance (see Lemma 2).

► **Theorem 10.** *Let \mathcal{Q} be a set of uniform samples with repetition from \mathcal{P} . Let \mathcal{C} be a set of candidate solutions. Let $a \in \mathcal{C}$ minimize $\sum_{Q \in \mathcal{Q}} m(a, Q)$ and let $\hat{c} = \operatorname{argmin}_{c \in \mathcal{C}} f(c)$. Then*

$$\Pr \left[\sum_{P \in \mathcal{P}} m(a, P) > (1+\varepsilon) \sum_{P \in \mathcal{P}} m(\hat{c}, P) \right] \leq |\mathcal{C}| \cdot e^{-\varepsilon^2 |\mathcal{Q}|/64}.$$

Putting all pieces together we have the following Theorem.

► **Theorem 11.** *Consider an input $\mathcal{P} = \{P_1, \dots, P_N\}$, 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 1). Its running time is $O(dn/\varepsilon^4 \cdot \log^2 1/\varepsilon)$.*

2.1 On reducing the size of the input sets

We would like to remove additionally the linear dependence on n for the maximum distance computations. This is motivated from the streaming extension of [22] where one aims at reading the input once in linear time and all subsequent computations should be sublinear,

or preferably independent of n . To this end a grid based strong coreset of size $1/\varepsilon^{\Theta(d)}$ was used. 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. Pagh et al. [24] showed that if the coreset is a subset of the input and approximates furthest neighbor queries to within less than a factor of roughly $\sqrt{2}$, then it must consist of $\Omega(\min\{n, \exp(d^{1/3})\})$ points. This was shown via a carefully constructed input point set of Agarwal and Sharathkumar [2] who used it to prove lower bounds on streaming algorithms for several extent problems.

In the next theorem, we review the techniques of the latter reference to show a slightly stronger result, namely 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 [2, 24], it is not limited to the streaming setting, and it is not restricted to subsets of the input.

► **Theorem 12.** *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 $\alpha < \sqrt{2}(1 - 2/d^{1/3})$, requires $\Omega(\min\{n, \exp(d^{1/3})\})$ bits of storage.*

On the positive side it was shown by Goel et al. [13] that a $\sqrt{2}$ -approximate furthest neighbor to the point $c \in \mathbb{R}^d$ can always be found on the surface of the smallest enclosing ball of the sets P_i , using linear preprocessing time $\tilde{O}(dn)$ and $\tilde{O}(d^2)$ query time. Thus, if we plug in the coresets of Bădoiu and Clarkson [5, 6] of size $O(1/\varepsilon)$ instead of the entire sets $P_i \in \mathcal{P}$ to evaluate $m(c, P_i)$, we would have a sublinear time algorithm (in n) after reading the input via a $\sqrt{2}(1 + \varepsilon)$ -approximation to any query $m(c, P_i)$. In a streaming setting the same bound can be achieved via the *blurred-ball-cover* of Agarwal and Sharathkumar [2] of slightly larger size $O(1/\varepsilon^3 \cdot \log 1/\varepsilon)$. Otherwise, Goel et al. [13] have shown that using $O(dn^{1+1/(1+\varepsilon)})$ preprocessing time and $\tilde{O}(dn^{1/(1+\varepsilon)})$ query time, one can obtain a $(1 + \varepsilon)$ -approximation for the furthest neighbor problem. Note that in this case the preprocessing time is already superlinear in n , and in particular the exponent is already larger than 1.7 for $1 + \varepsilon < \sqrt{2}$.

3 Applications

3.1 Probabilistic smallest enclosing ball

We apply our result to the probabilistic smallest enclosing ball problem, as given in [22]. In such a setting, the input is a set $\mathcal{D} = \{D_1, \dots, D_n\}$ 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 consisting of probabilistic points is also a random variable, where for each random choice of indices $(j_1, \dots, j_n) \in [z]^n$ there is a realization $P_{(j_1, \dots, j_n)} = X(j_1, \dots, j_n) = (q_{1,j_1}, \dots, q_{n,j_n})$. By independence of the distributions D_i , $i \in [n]$, it holds that $\Pr[X = P_{(j_1, \dots, j_n)}] = \prod_{i=1}^n p_{i,j_i}$.

The probabilistic smallest enclosing ball problem is defined as follows. Here we may assume that the distance of any point $c \in \mathbb{R}^d$ to the empty set is 0.

► **Definition 13** ([22]). 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 probabilistic smallest enclosing ball problem is to find a center $c^* \in \mathbb{R}^d$ that minimizes the expected smallest enclosing ball cost, i.e.,

$$c^* \in \operatorname{argmin}_{c \in \mathbb{R}^d} \mathbb{E}_X [m(c, X)],$$

where the expectation is taken over the randomness of $X \sim \mathcal{D}$.

The authors of [22] showed a reduction of the probabilistic smallest enclosing ball problem to computing a solution for the set median problem of Definition 1. Their algorithm distinguishes between two cases. In the first case the probability of obtaining a nonempty realization $P \neq \emptyset$ is small, more formally $\sum_{q_{i,j} \in Q} p_{i,j} \leq \varepsilon$, where $Q = \{q_{i,j} \mid q_{i,j} \neq \perp, i \in [n], j \in [z]\}$, and thus we have little chance of gaining information by sampling realizations. However, it was shown that

$$(1 - \varepsilon) \cdot \mathbb{E}_X \left[\sum_{p \in X} \|c - p\| \right] \leq \mathbb{E}_X [m(c, X)] \leq \mathbb{E}_X \left[\sum_{p \in X} \|c - p\| \right],$$

where $\mathbb{E}_X \left[\sum_{p \in X} \|c - p\| \right] = \sum_{i,j} p_{i,j} \cdot \|c - q_{i,j}\|$ is a weighted version of the deterministic 1-median problem, cf. [11], and thus also a weighted instance of the set median problem. In the second case, the probability that a realization contains at least one point is reasonably large. Therefore by definition of the expected value and $m(c, \emptyset) = 0$ we have

$$\mathbb{E}_X [m(c, X)] = \sum_{P \neq \emptyset} \Pr[X = P] \cdot m(c, P),$$

which is a weighted version of the set median problem. Depending on these two cases, we sample a number of elements, non-empty locations or non-empty realizations, and solve the resulting set median problem using the samples in Theorem 11 for computing the approximate subgradients.

Algorithm 2 adapts this framework. It differs mainly in three points from the previous algorithm of [22]. 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, as discussed in Section 2.1. Third, the running time of the actual optimization task is reduced via Theorem 11 instead of an exhaustive grid search.

► **Theorem 14.** 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 $\tilde{c} \in \mathbb{R}^d$ denote the output of Algorithm 2 on input \mathcal{D} , and let the approximation parameter be $\varepsilon < 1/9$. Then with constant probability the output is a $(1 + \varepsilon)$ -approximation for the probabilistic smallest enclosing ball problem. I.e., it holds that

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

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

Comparing to the result of [22], the running time is reduced from $O(dnz/\varepsilon^{O(1)} + 1/\varepsilon^{O(d)})$ to $O(dnz/\varepsilon^{O(1)})$, i.e., our dependence on the dimension d is no longer exponential but only linear. Note, in particular, that 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 will be crucial in the next application.

Algorithm 2: Probabilistic smallest enclosing ball.

Data: A set \mathcal{D} of n point distributions over z locations in \mathbb{R}^d , a parameter $\varepsilon < 1/9$

Result: A center $\hat{c} \in \mathbb{R}^d$

- 1 $Q \leftarrow \{q_{i,j} \mid q_{i,j} \neq \perp, i \in [n], j \in [z]\}$ /* the set of non-empty locations */
- 2 Set a sample size $k \in O(1/\varepsilon^2 \log(1/\varepsilon))$
- 3 **if** $\sum_{q_{i,j} \in Q} p_{i,j} \leq \varepsilon$ **then**
- 4 - Pick a random sample R of k locations from $\mathcal{P} = Q$, where for every $r \in R$ we have $r = q_{ij}$ with probability proportional to p_{ij}
- 5 - Compute $\hat{c} \in \mathbb{R}^d$ that is a $(1 + \varepsilon)$ -approximation using the sampled points R one-by-one for computing the approximate subgradients in the algorithm of Theorem 11
- 6 **else**
- 7 - Sample a set R of k non-empty realizations from the input distributions \mathcal{D}
- 8 - Compute $\hat{c} \in \mathbb{R}^d$ that is a $(1 + \varepsilon)$ -approximation using the sampled realizations R one-by-one for computing the approximate subgradients in the algorithm of Theorem 11
- 9 **return** \hat{c}

3.2 Probabilistic support vector data description

Now we turn our attention to the support vector data description (SVDD) problem [28] and show how to extend it to its probabilistic version. To this end, let $K: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ be a positive semidefinite kernel function. It is well known by Mercer's theorem, cf. [26], that such a function implicitly defines the inner product of a high dimensional Hilbert space \mathcal{H} , say \mathbb{R}^D where $D \gg d$. This means we have $K(x, y) = \langle \varphi(x), \varphi(y) \rangle$, where $\varphi: \mathbb{R}^d \rightarrow \mathcal{H}$ is the so called feature mapping associated with the kernel. Examples for such kernel functions include polynomial transformations of the standard inner product in \mathbb{R}^d such as the constant, linear or higher order polynomial kernels. In these cases D remains bounded but grows as a function of d raised to the power of the polynomials' degree. Other examples are the exponential, squared exponential, Matérn, or rational quadratic kernels, which are transformations of the Euclidean distance between the two low dimensional vectors. The dimension D of their implicit feature space is in principle unbounded. Despite the large dimension $D \gg d$, all these kernels can be evaluated in time $O(d)$ [25].

It is known that the SVDD problem is equivalent to the smallest enclosing ball problem in the feature space induced by the kernel function [30], i.e., given the kernel K with implicit feature mapping $\varphi: \mathbb{R}^d \rightarrow \mathcal{H}$, and an input set $P \subseteq \mathbb{R}^d$, the task is to find

$$c^* \in \operatorname{argmin}_{c \in \mathcal{H}} \max_{p \in P} \|c - \varphi(p)\| = \operatorname{argmin}_{c \in \mathcal{H}} m(c, \varphi(P)), \quad (1)$$

where $\varphi(P) = \{\varphi(p) \mid p \in P\}$.

Now we extend this to the probabilistic setting as we did in Section 3.1. The input is again a set \mathcal{D} of n discrete and independent probability distributions, where $D_i \in \mathcal{D}$ is defined over a set of z locations $q_{i,j} \in \mathbb{R}^d \cup \{\perp\}$. Note that the mapping φ maps the locations $q_{i,j}$ from \mathbb{R}^d to $\varphi(q_{i,j})$ in \mathcal{H} , and we assume $\varphi(\perp) = \perp$. Then the probabilistic SVDD problem is given by the following adaptation of Definition 13.

► **Definition 15.** 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 $K: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ be a kernel function with associated feature map $\varphi: \mathbb{R}^d \rightarrow \mathcal{H}$. The probabilistic support vector data description (pSVDD) problem is to find a center $c^* \in \mathcal{H}$ that minimizes the expected SVDD cost, i.e.,

$$c^* \in \operatorname{argmin}_{c \in \mathcal{H}} \mathbb{E}_X [m(c, \varphi(X))],$$

where the expectation is taken over the randomness of $X \sim \mathcal{D}$.

Note that the deterministic problem, see Equation (1), is often stated with squared distances [28, 30]. It does not matter whether we minimize the maximum distance or any of its powers or any other monotone transformation. In the probabilistic case this is not true. Consider for instance squared distances. The resulting problem would be similar to a 1-means rather than a 1-median problem. Huang et al. [16] have observed that minimizing the expected maximum squared distance corresponds to minimizing the expected area of an enclosing ball in \mathbb{R}^2 . This observation can be generalized to the expected volume of an enclosing ball in \mathbb{R}^p when the p -th powers of distances are considered. Considering $p = 2$ might also have advantages when dealing with Gaussian input distributions due to their strong connection to squared Euclidean distances. In a general setting of the probabilistic smallest enclosing ball problem however, it is natural to minimize in expectation the maximum Euclidean distance, as in Definitions 13 and 15, since its radius is the primal variable to minimize.

Next we want to show how to find a $(1 + \varepsilon)$ -approximation for the pSVDD problem. Explicitly computing any center $c \in \mathcal{H}$ takes $\Omega(D)$ time and space which is prohibitive not only when $D = \infty$. Note that for the SEB and SVDD problems, any reasonable center lies in the convex hull of the input points. Since taking the expectation is simply another linear combination over such centers, we can express any center $c \in \mathcal{H}$ as a linear combination of the set of non-empty locations, i.e.,

$$c = \sum_{q_{u,v} \in Q} \gamma_{u,v} \varphi(q_{u,v})$$

The idea is to exploit this characterization to simulate Algorithm 1 and thereby Algorithm 2 to work in the feature space \mathcal{H} by computing the centers and distances only implicitly.

For now, assume that any distance computation can be determined. Note that sampling a set $P_i \subset \mathbb{R}^d$ is the same as sampling the set $\varphi(P_i)$ of corresponding points in \mathcal{H} from the same distribution. We assume that we have a set of locations or realizations $\mathcal{P} = \{P_1, \dots, P_N\}$, with $P_i \subset \mathbb{R}^d$. The remaining steps are passed to Theorem 11 which is based on Algorithm 1. First, we show the invariant that each center c_i reached during its calls to Algorithm 1 can be updated such that we maintain a linear combination $c_i = \sum_{u,v} \gamma_{u,v} \varphi(q_{u,v})$, where at most $i + 1$ terms have $\gamma_{u,v} \neq 0$.

- The initial center $c_0 \in \mathcal{H}$ is chosen by sampling uniformly at random a set $P \in \mathcal{P}$ via Lemma 8. We take any point $q \in P$, $q \neq \perp$, which maps to $c_0 = \varphi(q)$. Thus the invariant is satisfied at the beginning, where the corresponding coefficient is $\gamma = 1$ and all other coefficients are zero.
- In each iteration we randomly sample a set $P \in \mathcal{P}$ to simulate the approximate subgradient $\tilde{g}(c_i)$ at the current point c_i . The vector $\tilde{g}(c_i)$ is a vector between c_i and some point $p_{j,k} = \varphi(q_{j,k}) \in \mathcal{H}$, such that $q_{j,k}$ maximizes $\|c_i - \varphi(q')\|$ over all $q' \in P$ (cf. Lemma 6). To implicitly update to the next center c_{i+1} note that (cf. Algorithm 1)

$$c_{i+1} = c_i - s \cdot \frac{c_i - \varphi(q_{j,k})}{\|c_i - \varphi(q_{j,k})\|} = \left(1 - \frac{s}{\|c_i - \varphi(q_{j,k})\|}\right) \cdot c_i + \frac{s}{\|c_i - \varphi(q_{j,k})\|} \cdot \varphi(q_{j,k}).$$

Assume the invariant was valid that c_i was represented as $c_i = \sum_{u,v} \gamma_{u,v} \varphi(q_{u,v})$ with at most $i + 1$ non-zero coefficients. Then it also holds for the point c_{i+1} since the previous non-zero coefficients of $\varphi(q_{u,v})$ are multiplied by $1 - s / \|c_i - \varphi(q_{j,k})\|$ and the newly added $\varphi(q_{j,k})$ is assigned the coefficient $s / \|c_i - \varphi(q_{j,k})\|$. So there are at most $i + 2$ non-zero coefficients.

Therefore, we do not have to store the points c_i explicitly while performing Algorithm 1. The implicit representation can be maintained via a list storing points that appear in the approximate subgradients and their corresponding non-zero coefficients.

To actually compute the coefficients, we need to be able to compute Euclidean distances as well as determine s . Using Lemma 9 we determined the step size s via an estimator $\tilde{R} = \sum_{P \in \mathcal{S}} m(c_0, P)$ based on a small sample \mathcal{S} . In particular this requires distance computations again. To this end, we show how to compute $\|c_i - \varphi(q)\|$ for any location $q \in \mathbb{R}^d$. Recall that the kernel function implicitly defines the inner product in \mathcal{H} . It thus holds that

$$\begin{aligned} \|c_i - \varphi(q)\|^2 &= \left\| \sum_{u,v} \gamma_{u,v} \varphi(q_{u,v}) - \varphi(q) \right\|^2 = \left\| \sum_{w=0}^i \gamma_w \varphi(q_w) - \varphi(q) \right\|^2 \\ &= \left\| \sum_{w=0}^i \gamma_w \varphi(q_w) \right\|^2 + \|\varphi(q)\|^2 - 2 \sum_{w=0}^i \gamma_w \langle \varphi(q_w), \varphi(q) \rangle \\ &= \sum_{w=0}^i \sum_{w'=0}^i \gamma_w \gamma_{w'} K(q_w, q_{w'}) + K(q, q) - 2 \sum_{w=0}^i \gamma_w K(q_w, q), \end{aligned} \quad (2)$$

where $w, w' \in \{0, \dots, i\}$ index the locations $q_w, q_{w'}$ with corresponding $\gamma_w, \gamma_{w'} \neq 0$ in iteration i . Therefore, we have the following Theorem.

► **Theorem 16.** *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 support vector data description problem. I.e., it holds that*

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

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

4 Conclusion and open problems

We studied the set median problem in high dimensions that minimizes the sum of maximum distances to the furthest point in each input set. We presented a $(1 + \varepsilon)$ -approximation algorithm whose running time is linear in d and independent of the number of input sets. We further discussed that in high dimensions the size of the input sets cannot be reduced sublinearly without losing a factor of roughly $\sqrt{2}$. Our work resolves an open problem of [22] and improves the previously best algorithm for the probabilistic smallest enclosing ball problem in high dimensions by reducing the dependence on d from exponential to linear. This enables running the algorithm in high dimensional Hilbert spaces induced by kernel functions which makes it more flexible and viable as a building block in machine learning and data analysis. As an example we transferred the kernel based SVDD problem of [28] to the probabilistic data setting. Our algorithms assume discrete input distributions. It would be interesting to extend them to various continuous distributions. The pSEB problem minimizes the expected maximum distance. When it comes to minimizing volumes of balls or

in the context of Gaussian distributions it might be interesting to study higher moments of this variable. This corresponds to a generalization of the set median problem to minimizing the sum of higher powers of maximum distances. Finally we hope that our methods may help to extend more shape fitting and machine learning problems to the probabilistic setting.

References

- 1 Marcel R. Ackermann, Johannes Blömer, and Christian Sohler. Clustering for metric and nonmetric distance measures. *ACM Transactions on Algorithms*, 6(4):59:1–59:26, 2010.
- 2 Pankaj K. Agarwal and R. Sharathkumar. Streaming Algorithms for Extent Problems in High Dimensions. *Algorithmica*, 72(1):83–98, 2015. doi:10.1007/s00453-013-9846-4.
- 3 Amir Beck and Shoham Sabach. Weiszfeld’s Method: Old and New Results. *Journal of Optimization Theory and Applications*, pages 1–40, 2014.
- 4 Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- 5 Mihai Bădoiu and Kenneth L. Clarkson. Smaller core-sets for balls. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 801–802, 2003.
- 6 Mihai Bădoiu and Kenneth L. Clarkson. Optimal core-sets for balls. *Computational Geometry*, 40(1):14–22, 2008.
- 7 Mihai Bădoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *Proceedings of the 34th ACM Symposium on Theory of Computing, STOC*, pages 250–257, 2002.
- 8 M. T. Chao. A general purpose unequal probability sampling plan. *Biometrika*, 69(3):653–656, 1982.
- 9 Mark Cieliebak, Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. Distributed Computing by Mobile Robots: Gathering. *SIAM Journal on Computing*, 41(4):829–879, 2012. doi:10.1137/100796534.
- 10 Michael B. Cohen, Yin Tat Lee, Gary L. Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In *Proceedings of the 48th ACM Symposium on Theory of Computing, STOC*, pages 9–21, 2016. doi:10.1145/2897518.2897647.
- 11 Graham Cormode and Andrew McGregor. Approximation algorithms for clustering uncertain data. In *Proceedings of the 27th ACM Symposium on Principles of Database Systems, PODS*, pages 191–200, 2008.
- 12 Pavlos S. Efrimidis. Weighted Random Sampling over Data Streams. In *Algorithms, Probability, Networks, and Games*, pages 183–195. Springer International, 2015. doi:10.1007/978-3-319-24024-4_12.
- 13 Ashish Goel, Piotr Indyk, and Kasturi R. Varadarajan. Reductions among high dimensional proximity problems. In *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 769–778, 2001.
- 14 Sudipto Guha and Kamesh Munagala. Exceeding expectations and clustering uncertain data. In *Proceedings of the 28th ACM Symposium on Principles of Database Systems, PODS*, pages 269–278, 2009.
- 15 Lingxiao Huang and Jian Li. Stochastic k -Center and j -Flat-Center Problems. In *Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 110–129, 2017. doi:10.1137/1.9781611974782.8.
- 16 Lingxiao Huang, Jian Li, Jeff M. Phillips, and Haitao Wang. ϵ -Kernel Coresets for Stochastic Points. In *Proceedings of the 24th Annual European Symposium on Algorithms, ESA*, pages 50:1–50:18, 2016.
- 17 Piotr Indyk. *High-dimensional Computational Geometry*. PhD thesis, Stanford University, 2000.
- 18 Ilan Kremer, Noam Nisan, and Dana Ron. On Randomized One-Round Communication Complexity. *Computational Complexity*, 8(1):21–49, 1999. doi:10.1007/s000370050018.

- 19 Amit Kumar, Yogish Sabharwal, and Sandeep Sen. Linear-time approximation schemes for clustering problems in any dimensions. *Journal of the Association for Computing Machinery*, 57(2):5:1–5:32, 2010.
- 20 Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- 21 Alexander Munteanu and Chris Schwiegelshohn. Coresets-Methods and History: A Theoreticians Design Pattern for Approximation and Streaming Algorithms. *Künstliche Intelligenz*, 32(1):37–53, 2018. doi:10.1007/s13218-017-0519-3.
- 22 Alexander Munteanu, Christian Sohler, and Dan Feldman. Smallest enclosing ball for probabilistic data. In *Proceedings of the 30th ACM Symposium on Computational Geometry, SoCG*, pages 214–223, 2014.
- 23 Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Springer, New York, 2004.
- 24 Rasmus Pagh, Francesco Silvestri, Johan Sivertsen, and Matthew Skala. Approximate furthest neighbor with application to annulus query. *Information Systems*, 64:152–162, 2017. doi:10.1016/j.is.2016.07.006.
- 25 Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006. URL: <http://www.worldcat.org/oclc/61285753>.
- 26 Bernhard Schölkopf and Alexander Johannes Smola. *Learning with Kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning series. MIT Press, 2002. URL: <http://www.worldcat.org/oclc/48970254>.
- 27 Marco Stolpe, Kanishka Bhaduri, Kamalika Das, and Katharina Morik. Anomaly Detection in Vertically Partitioned Data by Distributed Core Vector Machines. In *Proceedings of Machine Learning and Knowledge Discovery in Databases, ECML/PKDD Part III*, pages 321–336, 2013. doi:10.1007/978-3-642-40994-3_21.
- 28 David M. J. Tax and Robert P. W. Duin. Support Vector Data Description. *Machine Learning*, 54(1):45–66, 2004. doi:10.1023/B:MACH.0000008084.60811.49.
- 29 Mikkel Thorup. Quick k -Median, k -Center, and Facility Location for Sparse Graphs. *SIAM Journal on Computing*, 34(2):405–432, 2005.
- 30 Ivor W. Tsang, James T. Kwok, and Pak-Ming Cheung. Core Vector Machines: Fast SVM Training on Very Large Data Sets. *Journal of Machine Learning Research*, 6:363–392, 2005.
- 31 Endre Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal*, 43(2):355–386, 1937.
- 32 Endre Weiszfeld. On the point for which the sum of the distances to n given points is minimum. *Annals of Operations Research*, 167:7–41, 2009. Translated from the French original and annotated by Frank Plastria.