Clustering in High Dimensions

by

Mihai Bădoiu

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer Science and Engineering and Master of
Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2003

Author . . . .
Department of Electrical Engineering and Computer Science
May 9, 2003

Certified by . . . .
Piotr Indyk
Assistant Professor
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Clustering in High Dimensions

by

Mihai Bădoiu

Submitted to the Department of Electrical Engineering and Computer Science
on May 9, 2003, in partial fulfillment of the
requirements for the degree of
Bachelor of Science in Computer Science and Engineering and Master of Engineering in
Electrical Engineering and Computer Science

Abstract

In this thesis we show that, for several clustering problems, we can extract a small set of points, so that, using these *core-sets*, we can approximate clustering efficiently. The cardinality of these core-sets is independent of the dimension.

Using these sets, we develop a $(1+\varepsilon)$-approximation algorithm for the k-center clustering problem (with or without outliers)and the k-median clustering problem in Euclidean space. The running time of our algorithm has linear dependency on the number of points and in the dimension, and exponential dependency on $1/\varepsilon$ and k. Our algorithm runs substantially faster than the previously known algorithms.

We present a $(1+\varepsilon)$-approximation algorithm for the 1-cylinder clustering problem.

We show how to quickly construct small core-sets and the existence of optimal core-sets. We also present experimental results showing that in practice these core-sets are smaller than in the worst case.

Thesis Supervisor: Piotr Indyk
Title: Assistant Professor

Acknowledgments

The author wishes to thank Professor Piotr Indyk, Dr Kenneth L. Clarkson and Professor Sariel Har-Peled for their extensive guidance on this project, the MIT Theory of Computation Group for their support.

Contents

1 Introduction **5**

2 k-center clustering **8**

2.1 Notation 8

2.2 k-center clustering 9

2.3 k-center clustering with outliers 12

3 The 1-cylinder problem **15**

3.1 Introduction 15

3.2 Computing the approximate 1-cylinder 17

4 k-median clustering **21**

5 Smaller Core-Sets for Balls **28**

5.1 Introduction 28

5.2 Core-sets for 1-centers 29

5.3 Simple algorithm for 1-center 30

6 Optimal Core-Sets for Balls **31**

6.1 Introduction 31

6.2 A Lower Bound for Core-Sets 32

6.3 Optimal Core-Sets 32

6.4 Experimental Results 36

Chapter 1

Introduction

The problem of clustering a set of points is one of the central problems in computer-science. Clustering is related to unsupervised learning, classification, databases, spatial range-searching, data-mining, and other problems. Clustering has received a lot of attention over the last twenty years. There is a large literature on this topic, including numerous variants of clustering, see [7, p. 517-601] and [14, p. 296-346].

In this thesis, we present several algorithms for clustering of a finite set of points P in $\mathbb{R}^d$, where d is large. We focus on the following variants of clustering:

- **k-center:** We want to find a set S of k points such that we minimize the maximum distance from $v \in P$ to its closest point in S.

- **k-median:** We want to find a set S of k points such that we minimize the sum of distances from each point in P to its closest point in S.

- **k-center with outliers:** We want to solve the k-center problem in the case when we are allowed to ignore $\alpha|P|$ points.

- **1-cylinder:** We want to find a line l such that we minimize the maximum distance from $v \in P$ to l.

Our results rely on a new technique that extracts a small subset that "represents" this point set ε-well, in the sense that solving the problem for this subset gives a solution for the original problem of cost no more than a factor $(1 + \varepsilon)$ away for given ϵ. An important property of these sets is that their cardinality is *independent* of the dimension. The existence of similar subsets for various approximation problems was known before, but their cardinality depended polynomially or exponentially on the dimension, see [19, 2, 12, 13, 17].

We show that, for any ε such that $0 < \varepsilon < 1$, we can extract a subset of cardinality $O(1/\varepsilon^2)$, so that the minimum enclosing ball of the subset approximates the minimum enclosing ball of a set of points $P \subset \mathbb{R}^d$ by a factor of at most $(1+\varepsilon)$. We call such a subset a *core-set.* Using this core-set, we present a $2^{O((k \log k)/\varepsilon^2)} \cdot dn$-time algorithm that computes an $(1+\varepsilon)$-approximate k-center clustering of a point set P. No algorithm for this problem was known previously, although, as pointed out in [16], by using the techniques of [20], we can achieve a much slower algorithm with running time $n^{O(k^2/\varepsilon^2)}$. Agarwal and Procopiuc [1] developed a $O(n \log k) + (k/\varepsilon)^{O(dk^{1-1/d})}$-time algorithm for this problem, but their algorithm is exponential in d, and it is very inefficient for higher d.

For the special case where $k = 1$, the core-set technique yields an algorithm with running time $O\left(dn/\varepsilon^2 + (1/\varepsilon)^{O(1)}\right)$. This time is significantly better than before. Previously, the best running time was $O(d^3 n \log(1/\varepsilon))$, obtained by the ellipsoid algorithm, see [11].

We also present an $(1+\varepsilon)$-approximation algorithm for the problem of computing the cylinder of minimum radius that covers P. The running time of our algorithm is $n^{O(\log(1/\varepsilon)/\varepsilon^2)}$. The fastest algorithm previously ran in time $O(n+1/\varepsilon^{O(d)})$, which is exponential in d, see [13]. Our algorithm uses a core-set for 1-center clustering, dimensionality reduction, and convex programming.

We also show that by using random sampling, one can find a $O(1/\epsilon^{O(1)})$-size set of points R, such that the flat spanned by those points contains a point ε-close to the 1-median of the point-set. The only previous result of this type [17] used a sample of size linearly dependent on the dimension. Using the sampling technique, we present a $2^{(k/\varepsilon)^{O(1)}} d^{O(1)} n \log^{O(k)} n$ expected time algorithm that computes a $(1+\varepsilon)$-approximation to the optimal k-median for P (i.e., finds k points-medians in $\mathbb{R}^d$, such that the sum of the distances from all points P to their closest medians is minimized). Previously, the fastest known algorithm with polynomial dependence on d was due to Ostrovsky and Rabani [20] and it ran in $n^{(k+1/\varepsilon)^{O(1)}}$ time. For relevant results, see [3].

We also present an efficient algorithm for solving the k-center problem with *outliers.*

We prove the existence of optimal-sized core-sets for k-center clustering. We have also performed experimental tests and observed that in practice the error is much lower than the error that is guaranteed for a variety of core-set construction algorithms and the gradient-descent algorithm explained in [4].

In chapter 2 we present the k-center algorithm. In chapter 3 we present the algorithm for the cylinder of minimum radius problem. In chapter 4 we present the k-median algorithm. In chapter 5 we show how to construct small core-sets. In chapter 6 we show the existence of optimal core-sets along with experimental results.

Chapter 2

k-center clustering

2.1 Notation

We begin by defining some notions and notation.

Definition 2.1.1 For a minimization problem, an *α-approximation algorithm* is an algorithm that, for any input, guarantees a solution of cost at most αc where c is the minimum cost possible.

Definition 2.1.2 A closed ball with center at c and radius r can be written as $\mathrm{Ball}(c, r)$.

Definition 2.1.3 For a finite set of points $S \in \mathbb{R}$, let $\delta(S) = \max_{p,q \in S} ||p - q||$. We call $\delta(S)$ the *diameter* of S.

Definition 2.1.4 A *metric space* is a pair (X, dist) where X is a set and

$$\mathrm{dist}\colon X \times X \to [0, \infty)$$

is a function, called a *metric* satisfying $\mathrm{dist}(x, y) = 0$ if and only if $x = y$, $\mathrm{dist}(x, y) = \mathrm{dist}(y, x)$, and $\mathrm{dist}(x, y) + \mathrm{dist}(y, z) \geq \mathrm{dist}(x, z)$. For our purposes, $\mathrm{dist}(x, y) = ||x - y||$.

Definition 2.1.5 Let $f\colon X \to Y$ be a mapping of metric spaces with metric dist and dist'. Let $\alpha \geq 1$. Then f is called an *α-distortion embedding* if there exists a number $r > 0$ such that, for all $x, y \in X$,

$$r \cdot \mathrm{dist}(x, y) \leq \mathrm{dist}'(f(x), f(y)) \leq \alpha r \cdot \mathrm{dist}(x, y)$$

Definition 2.1.6 Given a finite set $P \in \mathbb{R}^d$, an *optimal k-center* is defined to be a set $S = \{c_1, \dots, c_k\}$ that minimizes

$$r_{cen}(P, k) = \min_{c_1,\dots,c_k \in \mathbb{R}} \max_{p \in P} \min_i ||c_i - p||.$$

Furthermore, $r_{cen}(P, k)$ is called the *radius* of the k-center clustering of P.

Definition 2.1.7 Given a finite set $P \in \mathbb{R}^d$, a *minimum enclosing closed ball* is the ball with radius $r_{cen}(P, 1)$ and center at c_1 where $\{c_1\}$ is the optimal 1-center for P.

Definition 2.1.8 Given a finite set $P \in \mathbb{R}^d$, its 1-*cylinder* is defined to be the closed cylinder of minimum radius that encloses P.

2.2 k-center clustering

In this section, we present an efficient approximation algorithm for of finding an optimal k-center clustering.

We start by restating the following lemma, proved originally in [9]. For the sake of completeness, we give the proof here.

Lemma 2.2.1 *Let B be a minimum enclosing closed ball of a finite point set $P \subset \mathbb{R}^d$. Let c_B be the center of B, and r the radius of B. Let H be a closed half-space that contains c_B. Then H must also contain at least one point of P that is at distance r from c_B.*

Proof: Suppose H does not contain any common point of P and B. Since H is closed and P is finite, there exists an $\varepsilon > 0$ such that the minimum distance between the points of $P \setminus H$ and H is greater than ε. Fix ε such that the distance between any of the points in $P \bigcap H$ and c_B is at most $r - \varepsilon$. We can translate the ball B in the direction perpendicular to H by $\varepsilon/2$ such that the ball still encloses all the points of P. None of the points of P lie exactly on the boundary of the translated ball. Thus we can shrink the radius of the ball, and we have a smaller ball that contains P. ■

Lemma 2.2.2 *Given a finite set $P \subset \mathbb{R}$, and ε for which $1 > \varepsilon > 0$, there exists a subset of points $S \subseteq P$ such that the distance between the center of the minimum enclosing ball of S*

and any point in P is at most $(1+\varepsilon)$ times the radius of the minimum enclosing ball of P and $|S| = O(1/\varepsilon^2)$.

Proof: Start with an arbitrary point $x \in P$, and let y be a point in P farthest away from x. By the triangle inequality, $\|x - y\| \geq \operatorname{diam}/2$ where $\operatorname{diam} = \max_{p,q \in P} ||p - q||$.

Set $S_0 = \{x, y\}$. In the following, we maintain a set S_i of points and its minimum enclosing ball $B_i = \mathrm{Ball}(c_i, r_i)$. Clearly, $r_0 \geq \operatorname{diam}/4$.

Let B_{opt} be the minimum enclosing ball of P with the center at c_{opt} and radius r_{opt}. There are two possibilities: First suppose there is no point $p \in P$ such that $\|p - c_i\| \geq (1+\varepsilon)r_i$. Then we are done. Indeed, the ball $B = \mathrm{Ball}(c_i, (1+\varepsilon)r_i)$ encloses P, and has a radius of at most $(1+\varepsilon)r_{opt}$. Hence B is a $(1+\varepsilon)$-approximation.

Second, there exists a point $p \in P$ such that $\|p - c_i\| \geq (1+\varepsilon)r_i$. In this case, we set $S_{i+1} = S_i \cup \{p\}$.

We now prove that, for $0 < \varepsilon < 1$, we have $r_{i+1} \geq \left(1 + \dfrac{\varepsilon^2}{16}\right) r_i$. If

$$\|c_i - c_{i+1}\| < (\varepsilon/2)r_i,$$

then, by the triangle inequality, we have

$$||p - c_{i+1}|| \geq ||p - c_i|| - ||c_i - c_{i+1}|| \geq (1+\varepsilon)r_i - \frac{\varepsilon}{2}r_i = \left(1 + \frac{\varepsilon}{2}\right) r_i.$$

Otherwise, let H be the hyperplane that passes through c_i and is orthogonal to $c_i c_{i+1}$. Let H^- be the open half-space bounded by H and containing p. See Figure 2-1.

From Lemma 2.2.1, we know that there exists a point $x \in B_i$ with $x \notin H^-$. Therefore, for $0 < \varepsilon < 1$,

$$r_{i+1} \geq ||c_{i+1} - x|| \geq \sqrt{r_i^2 + \frac{\varepsilon^2}{4} r_i^2} \geq \left(1 + \frac{\varepsilon^2}{16}\right) r_i.$$

Since $r_0 \geq \operatorname{diam}/4$, and at each step we increase the radius of our solution by at least $(\operatorname{diam}/4)\varepsilon^2/16 = \operatorname{diam}\varepsilon^2/64$, it follows that we cannot encounter this case more than $64/\varepsilon^2$ times, as diam is an upper bound of the radius on the minimum enclosing ball of P. ■

Theorem 2.2.3 *For any finite point set $P \subset \mathbb{R}^d$ and $1 > \varepsilon > 0$, there is a subset $S \subset P$ such that $|S| = O(1/\varepsilon^2)$, and such that if o is the 1-center of S, then o is a $(1+\varepsilon)$-approximate*

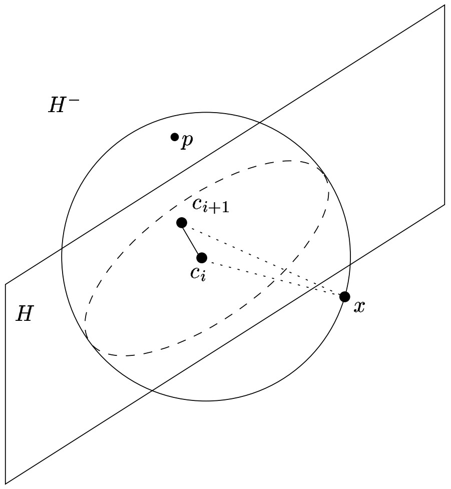

Figure 2-1: The ball B.

1-*center of* P*. The set* S *can be found in time* $O\big(dn/\varepsilon^2 + (1/\varepsilon)^{10}\log(1/\varepsilon)\big)$.

Proof: The proof of Lemma 2.2.2 is constructive, so it yields an algorithm for computing S. The algorithm requires computing $O(1/\varepsilon^2)$ times a $(1+\varepsilon)$-approximate enclosing ball of at most $O(1/\varepsilon^2)$ points in $\mathbb{R}^{O(1/\varepsilon^2)}$. The algorithm also requires reading the points $O(1/\varepsilon^2)$ times, which takes time $O(nd/\varepsilon^2)$. Thus, computing a $(1+\varepsilon)$-approximate enclosing ball can be done in time $O(nd/\varepsilon^2 + 1/\varepsilon^{10}\log(1/\varepsilon))$, using convex programming techniques [11]. ■

Theorem 2.2.4 *For any finite point set* $P \subset \mathbb{R}^d$ *and* $1 > \varepsilon > 0$*, a* $(1+\varepsilon)$*-approximate* 2*-center for* P *can be found in* $2^{O(1/\varepsilon^2)}dn$ *time.*

Proof: We start from two empty sets of points S_1 and S_2. At each stage, let B_1 and B_2 denote the smallest enclosing balls of S_1 and S_2. In the ith iteration of the algorithm, we pick the point p_i farthest from B_1 and B_2. To decide whether to put p_i in S_1 or in S_2, we make a guess. Clearly, if our guesses are correct, after $O(1/\varepsilon^2)$ iterations, we are done by Theorem 2.2.3. Thus, the running time of this algorithm is $O(dn/\varepsilon^2 + (1/\varepsilon)^{10})$.

To eliminate the guessing, we exhaustively enumerate all possible guesses. Thus, the running time of the algorithm $2^{O(1/\varepsilon^2)}$ for each guess sequence. The total running time of the algorithm is $dn2^{O(1/\varepsilon^2)}$. ■

Theorem 2.2.5 *For any finite point set* $P \subset \mathbb{R}^d$ *and* $1 > \varepsilon > 0$*, a* $(1+\varepsilon)$*-approximate* k*-center for* P *can be found in* $2^{O((k\log k)/\varepsilon^2)}dn$ *time.*

Proof: The algorithm is a straightforward extension of that of Theorem 2.2.4, where each guess now is a number between 1 and k, and we have to generate $O(k/\varepsilon^2)$ guesses. ■

2.3 k-center clustering with outliers

Definition 2.3.1 For a point-set P in $\mathbb{R}^d$, let $r_{cen}(P,k)$ denote the radius of the k-center clustering of P. This is the problem in which one wishes to find the k centers (i.e., points), so that the maximum distance of a point to a center is minimized.

Let $r_{cen}(P,k,\alpha)$ denote the minimal radius clustering with outliers; namely, we allow to throw out $\alpha|P|$ outliers. Computing this value is the (k,α)-center problem. Formally,

$$r_{cen}(P,k,\alpha) = \min_{S \subseteq P, |S| \geq (1-\alpha)|P|} r_{cen}(S,k).$$

The problem of computing k-center with outliers is interesting, as the standard k-center clustering is very sensitive to outliers.

Theorem 2.3.2 *For any point-set $P \subset \mathbb{R}^d$, parameters $1 > \varepsilon, \alpha > 0, \mu > 0$, a random sample R of $O(1/(\varepsilon\mu))$ points from P spans a flat containing a $(1+\varepsilon, 1, \mu+\alpha)$-approximate solution for the 1-center with α-outliers for P. Namely, there is a point $p \in \mathrm{span}(R)$, such that a ball of radius $(1+\varepsilon)r_{cen}(P,1,\alpha)$ centered at p, contains $(1-\alpha-\mu)$ points of P.*

Furthermore, we can compute such a cluster in $O(f(\varepsilon,\mu)nd)$ time, where $f(\varepsilon,\mu) = 2^{O\left(\log^2 \frac{1}{\varepsilon\mu}/(\varepsilon\mu)\right)}$

Proof: The proof follows closely the proof of Theorem 4.0.5. Let c_{opt} denote the center of the optimal solution, $r_{opt} = r_{cen}(P,1,\alpha)$ denote its radius, and $B_{\mathrm{opt}} = \mathrm{Ball}(c_{opt}, r_{opt})$. Let $s_1, \ldots, s_i$ be our random sample, $F_i = \mathrm{span}(s_1,\ldots,s_i)4$, and $c_i = \mathrm{proj}(c_{opt}, F_i)$, and we set $\beta = \sqrt{\varepsilon}$, and

$$U_i = \left\{ x \,\middle|\, \pi/2 - \beta \leq \angle c_{opt} c_i x \leq \pi/2 + \beta \right\}$$

be the complement of the cone of angle $\pi - \beta$ emanating from c_i, and having $c_i c_{opt}$ as its axis.

Let $P_i = U_i \cap P \cap B_{\mathrm{opt}}$. For any point $p \in P_i$, we have

$$\|pc_i\| \leq \sqrt{x_p^2 + y_p^2} \leq r_{opt}\sqrt{1+4\beta^2} = r_{opt}(1+O(\varepsilon)).$$

Namely, as far as the points of P_i are concerned, c_i is a good enough solution.

Let $Q_i = P \setminus P_i$. As long as $|Q_i| \geq (\alpha + \mu)|P|$, we have a probability of $1/\mu$ to sample a random point that is in Q_i and is not an outlier. Arguing as in the proof of Theorem 4.0.5, in such a case, the distance of the current flat to c_{opt} shrank down by a factor of $(1-\beta^2/4)$. Thus, as in the proof of Theorem 4.0.5, we perform the random sampling in rounds. In our case, we need $O((1/\varepsilon)\log(1/\varepsilon))$ rounds, and each round requires in expectation $1/\mu$ random samples. Thus, overall, if we sample $M = O((1/(\varepsilon\mu))\log(1/\varepsilon))$ points, we know that with constant probability, $\mathrm{span}(s_1, \ldots, s_M)$ contains a point in distance εr_{opt} from c_{opt}.

As for the algorithm, we observe that using random sampling, we can approximate $r_{opt}(P, 1, \alpha)$ up to a factor of two in $o(dn)$ time. Once we have this approximation, we can generate a set of candidates, as done in Theorem 4.0.7. This would result in

$$f(\varepsilon, \mu) = O\left(\left(\frac{10|R|}{\varepsilon}\right)^{|R|} \cdot |R|\right) = 2^{O((1/(\varepsilon\mu))\log^2(1/(\varepsilon\mu)))}$$

candidates. For each candidate we have to spend $O(nd)$ time on checking its quality. Overall, we get $f(\varepsilon, \mu)nd$ running time. ■

Remark 2.3.3 Theorem 2.3.2 demonstrates the robustness of the sampling approach we use. Although we might sample outliers into the sample set R, this does not matter, as in our argumentation we concentrate only on the points that are sampled correctly. Essentially, $\|F_i c_{opt}\|$ is a monotone decreasing function, and the impact of outliers, is only on the size of the sample we need to take.

Theorem 2.3.4 *For any point-set $P \subset \mathbb{R}^d$, parameters $1 > \varepsilon, \alpha > 0, \mu > 0, k > 0$, one can compute a $(k, \alpha+\mu)$-center clustering with radius smaller than $(1+\varepsilon)r_{cen}(P, k, \alpha)$ in $2^{(k/\varepsilon\mu)^{O(1)}}nd$ time. The result is correct with constant probability.*

Proof: Let P' be the set of points of P covered by the optimal k-center clustering $C_1, \ldots, C_k$, with αn outliers. Clearly, if any of the C_is contain less than $(\mu/k)n$ points of P', we can just skip it altogether.

To apply Theorem 2.3.2, we need to sample $O(k/(\varepsilon\mu))$ points from each of those clusters (we apply it with μ/k for the allowed fraction of outliers). Each such cluster, has size at least $(\mu/k)n$, which implies that a sample of size $O(k^2/(\mu^2\varepsilon))$ would contain enough points from C_i. To improve the probabilities, we would sample $O(k^3/(\mu^2\varepsilon))$ points. Let R be this random sample.

With constant probability (by the Markov inequality), $|R \cap C_i| = \Omega(k/(\varepsilon\mu))$, for $i = 1, \ldots, k$. We exhaustively enumerate for each point of R to which cluster it belongs. For such partition we apply the algorithm of Theorem 2.3.2. The overall running time is $2^{O((k/(\varepsilon\mu)^{O(1)}))}nd$. ■

Chapter 3

The 1-cylinder problem

3.1 Introduction

Let P be a set of n points in $\mathbb{R}^d$. We are interested in finding a line that minimizes the maximum distance from the line to the points. More specifically, we are interesting in finding a $(1+\varepsilon)$-approximation to this problem in polynomial time.

In the following, let ℓ_{opt} denote the axis of the optimal cylinder, r_{opt} denote the radius

$$r_{opt} = \text{radius}(P, \ell_{opt}) = \max_{p \in P} ||\ell_{opt} - p||,$$

and H_{opt} denote the hyperplane perpendicular to ℓ_{opt} that passes through the origin.

In this chapter, we prove the following theorem.

Theorem 3.1.1 *Given a set of n points in $\mathbb{R}^d$, and a parameter $\varepsilon > 0$, we can compute, in $n^{O(\log(1/\varepsilon)/\varepsilon^2)}$ time, a line l such that* $\text{radius}(P, l) \leq (1+\varepsilon) r_{opt}$ *where r_{opt} is the radius of the minimal* 1-*cylinder that contains P.*

First, observe that we can compute a 2-approximation to the 1-cylinder problem by finding two points $p, q \in P$ that are farthest apart and by taking the minimum cylinder having pq as its axis and containing P. It is easy to verify that the radius R of this cylinder is at most $2r_{opt}$ where r_{opt} is the radius of the minimum cylinder. Let t be a point on the cylinder. The radius of the minimum cylinder that encloses p, q and y, which constitutes a lower bound for r_{opt}, is half the radius of our cylinder. Computing R can be done in $O(n^2 d)$ time.

Let l be the center line of a solution with cost r_{opt}. Assume that we know the value of r_{opt}

up to a factor very close to 1, since we can enumerate all potential values of r_{opt} in the range $R/2, \ldots, R$. The main idea of the algorithm is to compute a $(1+\varepsilon)$ distortion embedding of the n points from $\mathbb{R}^d$ into $\mathbb{R}^{\log n/\varepsilon^2}$, to guess the solution to the problem there, and then to pull back the solution to the original space. However, a simple application of this method is known not to work for clustering problems (and the 1-cylinder problem in particular), due to the following difficulties:

1. The optimal solution found in the low-dimensional space does not have to correspond to any solution in the original space.

2. Even if (1) were true, it would not be clear how to pull back the solution from the low-dimensional one.

To overcome these difficulties, we proceed as follows:

1. In the first step, we find a point h lying on ℓ_{opt}. To be more precise, we guess h by enumerating polynomially many candidates instead of finding h; moreover, h does not lie on l_{opt}, but only sufficiently close to it.

2. We remove from P all the points within a distance of $(1+\varepsilon)r_{opt}$ from h. Since our final line passes through h, it is sufficient to find a solution for the smaller set P.

3. We embed the whole space into a low-dimensional space, such that, with high probability for all points $p \in P$, the angle between $\overrightarrow{ph}$ and ℓ_{opt} is approximately preserved.

   As we discuss in Steps 3 and 4 of Section 3.1, such an embedding A is guaranteed by the Johnson–Lindenstrauss Lemma.

4. We guess an approximate low-dimensional image Al of the line l by exploring polynomially many possibilities. By the properties of the embedding A, we can detect which points in P lie on one side of the hyperplane H passing through h and orthogonal to l. We modify the set P by replacing each point p on the other side side of the hyperplane by its reflection around h. Note that this operation increases the solution cost only by at most εr_{opt}.

5. It is now sufficient to find an optimal half-line beginning at h, which minimizes the maximum distance from $p \in P$ to the half-line. Moreover, we know that the points are on the same side of a hyperplane that passes through h. This problem can be solved using convex programming tools.

Thus, we use the low-dimensional image of l to discover the "structure" of the optimal solution, namely, which points lie on which side of the hyperplane H. Knowing this structure allows us to reduce our non convex problem to a convex one.

3.2 Computing the approximate 1-cylinder

In this section, we elaborate on each step in computing the cylinder of minimum radius containing a given point set.

Step 1. Finding a point near the line.

Definition 3.2.1 Let U be any set of points in $\mathbb{R}^d$, and $\varepsilon > 0$. Let $\mathcal{CH}(U)$ be the convex hull of U. We say that a finite subset A of points is an *ε-net* for U if, for any line ℓ that intersects $\mathcal{CH}(U)$, there is a $v \in A$ such that $\mathrm{dist}(v, \ell) \le \frac{\varepsilon}{2} r_{opt}$.

Lemma 3.2.2 *Let U be a set of points in $\mathbb{R}^d$, and $\varepsilon > 0$. We can compute an ε-net $A(U)$ for U in $(|U|^{2.5}/\varepsilon)^{O(|U|)}$ time. The cardinality of $A(U)$ is $(|U|^{2.5}/\varepsilon)^{O(|U|)}$.*

Proof: Let $M = |U|$. Let H be the $(M-1)$-dimensional affine subspace spanned by U. Note that $M \le |U|$. Let $\mathcal{E} \subseteq H$ be an ellipsoid such that $\mathcal{E}/(M+1)^2 \subseteq \mathcal{CH}(U) \subseteq \mathcal{E}$ where $\mathcal{E}/(M+1)^2$ is $\mathcal{E}$ scaled down around its center by a factor of $1/(M+1)^2$. Such an ellipsoid exists, and can be computed in polynomial time in $|U|$, see [11]. Let $\mathcal{B}$ be the minimum bounding box of $\mathcal{E}$ that is parallel to the main axises of $\mathcal{E}$. We claim that $\mathcal{B}/\sqrt{M}$ is contained inside $\mathcal{E}$. Indeed, there exists a linear transformation $\mathcal{T}$ that maps $\mathcal{E}$ to a unit ball $\mathcal{S}$. The point $\mathbf{q} = (1/\sqrt{M}, 1/\sqrt{M}, \dots, 1/\sqrt{M})$ lies on the boundary of this sphere. Clearly, $\mathcal{T}^{-1}(\mathbf{q})$ is a corner of $\mathcal{B}/\sqrt{M}$, and is on the boundary of $\mathcal{E}$. In particular,

$$\begin{aligned} \mathrm{diam}(\mathcal{B}) &= \sqrt{M}\,\mathrm{diam}(\mathcal{B}/\sqrt{M}) \le \sqrt{M}\,\mathrm{diam}(\mathcal{E}) \\ &\le \sqrt{M}(M+1)^2\,\mathrm{diam}(\mathcal{E}/(M+1)^2) \le \sqrt{M}(M+1)^2\,\mathrm{diam}(U). \end{aligned}$$

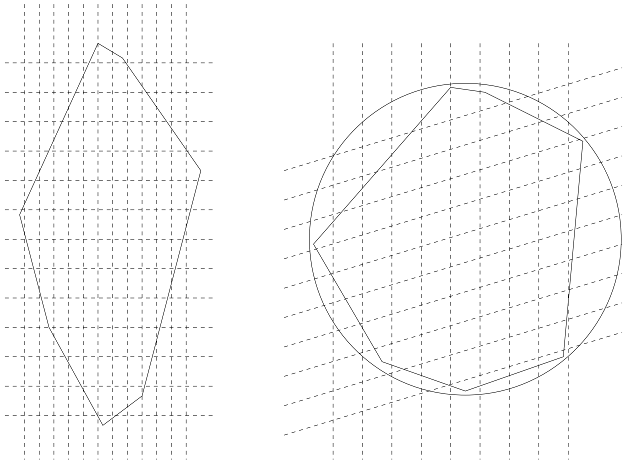

Figure 3-1: "Helper" points in a convex body after the embedding.

For any line ℓ, the same argument works for the projection of those objects in the hyperplane perpendicular to ℓ. Let $\mathcal{TP}_\ell$ denote this projection. Then we have

$$\begin{aligned}\operatorname{diam}(\mathcal{TP}_\ell(\mathcal{B})) &\leq \sqrt{M}(M+1)^2 \operatorname{diam}(\mathcal{TP}_\ell(U)) \\ &\leq 2\sqrt{M}(M+1)^2 \operatorname{dist}(U,\ell).\end{aligned}$$

Next, we partition $\mathcal{B}$ into a grid such that each cell is a translated copy of

$$\mathcal{B}_\varepsilon = \frac{\varepsilon}{2}\mathcal{B}/(2\sqrt{M}(M+1)^2).$$

This grid has $(M^{2.5}/\varepsilon)^{O(M)}$ vertices. Let $A(U)$ denote this set of vertices.

Let ℓ be any flat intersecting $\mathcal{CH}(U)$. We claim that one of the points in $A(U)$ is within distance $\frac{\varepsilon}{2}\operatorname{dist}(U,\ell)$ from ℓ. Indeed, let z be any point in $\mathcal{CH}(U)\cap\ell$. Let $\mathcal{B}''_\varepsilon$ be the grid cell containing z, and let v be one of its vertices. Clearly,

$$\begin{aligned}\operatorname{dist}(v,\ell) &\leq \|\mathcal{TP}_\ell(v)\mathcal{TP}_\ell(z)\| \leq \operatorname{diam}(\mathcal{TP}_\ell(\mathcal{B}''_\varepsilon)) \\ &= \frac{\varepsilon}{2}\cdot\frac{1}{2\sqrt{M}(M+1)^2}\operatorname{diam}(\mathcal{TP}_\ell(\mathcal{B})) \leq \frac{\varepsilon}{2}\operatorname{dist}(U,\ell).\end{aligned}$$

Thus our assertion is established. ■

In order to find the point h, we need to add "helper" points. For each subset S of P with $|S| = O(1/\varepsilon^2)$, we compute an ε-net on the interior of the convex body spanned by the points of S, see Figure 3-1.

Let $A(S)$ be an ε-net for S. Clearly, $|A(S)| = (|S|^{2.5}/\varepsilon)^{O(|S|)}$ where $|S| = O(1/\varepsilon^2)$. We have

$$|A(S)| = 2^{O(\log(1/\varepsilon)/\varepsilon^2)}.$$

Lemma 3.2.3 *Under the conditions of Lemma 3.2.2, consider the points in $G(S)$ for all sets S. At least one point is at most εr_{opt} away from ℓ_{opt}.*

Proof: Project all the points into a hyperplane H_{opt} that is orthogonal to ℓ_{opt}, and denote this set of points by P'. Let o be the point of intersection of ℓ_{opt} with H_{opt}. Since all the points are at distance at most r_{opt} from ℓ_{opt}, all the points projected into H are at distance at most r_{opt} from o. Compute the minimum enclosing ball of the point set P'. It is easy to see that, if the origin of the minimum enclosing ball is not o, then we can come up with a solution for the minimum fitting line of cost lower than r_{opt} by just translating l to intersect the center of the ball. Therefore, the minimum enclosing ball of P' has the origin in o.

By Theorem 2.2.3, there exists a set $S \subset P'$ such that $|S| = O(1/\varepsilon^2)$, and such that the minimum enclosing ball of the S is at most $(\varepsilon/2)r_{opt}$ away from o and since the center of any minimum enclosing ball of a set of points can be written as a convex combination of the points, we conclude that there exists a point p, a convex combination of the points of S such that $D(p, o) \leq \varepsilon r_{opt}$. Also, distance from p to the closest point of $G(S)$ is at most $(\varepsilon/2)r_{opt}$. Therefore, there exists a point in our ε-net that is at most εr_{opt} away from the optimal fitting line. ■

Step 2. Removing the points near h. For simplicity of exposition, from now on, we assume that h *lies* on the optimal line ℓ_{opt}. We remove from P all the points within distance $(1+\varepsilon)r_{opt}$ from h.

Clearly, the removal step can be implemented in linear time in the input size. Observe that, after this step, for all points p, the angle between $\overrightarrow{ph}$ and ℓ_{opt} is in the range $[0, \pi/2 - \varepsilon/2] \cup [\pi/2 + \varepsilon/2, \pi]$ for small enough ε. As we will see next, this property implies that the angles do not change in value from less than $\pi/2$ to greater than $\pi/2$ after the dimensionality reduction is applied.

Step 3. Random projection. We show how to find a mapping $A\colon \mathbb{R}^d \to \mathbb{R}^{d'}$ for $d' = O(\log n/\varepsilon^2)$ that preserves all angles $\angle\overrightarrow{hp}\ell_{opt}$ for $p \in P$ up to an additive factor of $\varepsilon/3$. For this purpose, we use the Johnson–Lindenstrauss Lemma. It is not difficult to verify (see [8]) that if we set the error parameter of the lemma to ε/C for large enough constant C, then all the angles are preserved up to an additive factor of, say, $\varepsilon/4$. Hence, for each $p \in P$, the image of p is on the same side of the image of the hyperplane H as the original points p. Moreover, for any $p \in P$, the angle $\angle(\overrightarrow{hp}, \ell_{opt})$ is in the range $[0, \pi/2 - \varepsilon/4] \cup [\pi/2 + \varepsilon/4, \pi]$.

Step 4. Guessing the image of l. We now need to approximate the image $A\ell_{opt}$ where A is the mapping generated by the Johnson–Lindenstrauss Lemma. For this purpose, we need to know the direction of l, since we already know one point through which the line $A\ell_{opt}$ passes. Our approach is to enumerate all the different directions of a line $A\ell_{opt}$. Obviously, the number of such directions is infinite. However, since we use the line *exclusively* for the purpose of separating the points $p \in P$ according to their angle $\angle\overrightarrow{hp}\ell_{opt}$, and those angles are separated from $\pi/2$ by $\varepsilon/4$, it is sufficient to find a direction vector that is within angular distance $\varepsilon/4$ from the direction of l. Thus, it is sufficient to enumerate all directions from an $\varepsilon/4$-net for the set of all directions. It is known that such spaces of cardinality $n^{O(\log(1/\varepsilon)/\varepsilon^2)}$ exist, and are constructible. Thus, we can find the right partition of points in P by enumerating a polynomial number of directions.

After finding the right partition of P, say, into P_L and P_R, we replace each point in P_L by its reflection through h; say the resulting set is $P'_L = \{2h - p \,\Big|\, p \in P\}$. Note that there is a one-to-one correspondence between the 1-cylinder solutions for P that pass through h and the 1-half-line solutions for $P' = P'_L \cup P_R$. By definition, the 1-half-line problem is to find a half-line r that has an endpoint at h and minimizes the maximum, over all input points p, of the distance from p to r. Thus, it remains to solve the 1-half-line problem for P'.

Step 5. Solving the 1-half-line problem using convex programming. We focus on the decision version of this problem. Assume we want to check if there is a solution with cost at most T. For each point p, let C_p be the cone of all half-lines with endpoints in h and that are within distance T from p. Clearly, C_p is convex. The problem is now to check if an intersection of all cones C_p is nonempty. This problem is one in convex programming, and thus can be solved up to arbitrary precision in polynomial time [11].

Chapter 4

k-median clustering

In this chapter, we present an efficient approximation algorithm for the k-median problem.

Definition 4.0.4 For a set P of n points in $\mathbb{R}^d$, let $\mathrm{med}_{\mathrm{opt}}(P,k) = \min_{K \subseteq \mathbb{R}^d, |K|=k} \sum_{p \in P} \mathrm{dist}(K,p)$ denote the optimal price of the k-median problem, where $\mathrm{dist}(K,p) = \min_{x \in K} \|xp\|$. Let $\mathrm{AvgMed}(P,k) = \mathrm{med}_{\mathrm{opt}}(P,k)/|P|$ denote the average radius of the k-median clustering.

For any sets $A, B \subset P$, we use the notation $\mathrm{cost}(A,B) = \sum_{a \in A, b \in B} \|ab\|$. If $A = \{a\}$, we write $\mathrm{cost}(a,\cdot)$ instead of $\mathrm{cost}(\{a\},\cdot)$, similarly for b. Moreover, we define $\mathrm{cost}(x \vee y, A) = \sum_{a \in A} \min\left(\|ax\|, \|ay\|\right)$.

For a set of points $X \subseteq \mathbb{R}^d$, let $\mathrm{span}(X)$ denote the affine subspace spanned by the points of X. We refer to $\mathrm{span}(X)$ as the *flat* spanned by X.

Theorem 4.0.5 *Let P be a point-set in $\mathbb{R}^d$, $1 > \varepsilon > 0$, and let X be a random sample of $O(1/\varepsilon^3 \log 1/\varepsilon)$ points from P. Then with constant probability, the following two events happen: (i) The flat $\mathrm{span}(X)$ contains a $(1+\varepsilon)$-approximate 1-median for P, and (ii) X contains a point in distance $\leq 2\,\mathrm{AvgMed}(P,1)$ from the center of the optimal solution.*

Proof: Let $\mathrm{med}_{\mathrm{opt}} = \mathrm{med}_{\mathrm{opt}}(P,1)$ be the price of the optimal 1-median, $R = \mathrm{AvgMed}(P,1)$, and let $s_1, \ldots, s_u$ be our random sample. In the following, we are going to partition the random sample into rounds: A round continues until we sample a point that has some required property. The first round continues till we encounter a point s_i, such that $\|s_i c_{opt}\| \leq 2R$, where c_{opt} is the center of the optimal 1-median. By the Markov inequality, $\|s_i c_{opt}\| \leq 2R$, with probability $\geq 1/2$.

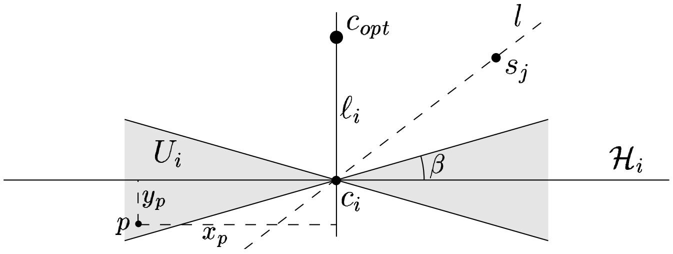

Figure 4-1: A round terminates as soon as we pick a point outside U_i.

Let's assume that s_i is a sample that just terminated a round, and we start a new sampling round. Let F_i be the flat spanned by the first i points in our sample: $s_1, \ldots, s_i$. Observe that if $\|F_i c_{opt}\| \leq \varepsilon R$, then we are done, as the point $\mathrm{proj}(c_{opt}, F_i)$ is the required approximation, where $\mathrm{proj}(c_{opt}, F_i)$ denotes the projection of c_{opt} into F_i.

Note that the distance from c_{opt} to F_i is monotone decreasing. That is $d_{i+1} = \|F_{i+1} c_{opt}\| \leq d_i = \|F_i c_{opt}\|$. We would next argue that either after taking enough sample points, d_i is small enough so that we can stop, or otherwise almost all the points of P lie very close to our spanned subspace, and we can use P to find our solution.

Indeed, let $c_i = \mathrm{proj}(c_{opt}, F_i)$, and let

$$U_i = \left\{ x \;\middle|\; x \in \mathbb{R}^d \text{ s.t. } \pi/2 - \beta \leq \angle c_{opt} c_i x \leq \pi/2 + \beta \right\}$$

be the complement of the cone of angle $\pi - \beta$ emanating from c_i, and having $c_i c_{opt}$ as its axis, where $\beta \leq \varepsilon/16$. See Figure 4-1. Let $\mathcal{H}_i$ be the $(d-1)$-dimensional hyperplane passing through c_i and perpendicular to $c_i c_{opt}$. For a point $p \in P$, let x_p be the distance of p to the line ℓ_i passing through c_{opt} and c_i, and let y_p be the distance between p and $\mathcal{H}_i$.

If $p \in U_i$, then $y_p \leq x_p \tan\beta \leq x_p \dfrac{\sin\beta}{\cos\beta} \leq 4\beta x_p \leq \dfrac{\varepsilon x_p}{4} \leq \dfrac{\varepsilon}{4} \|p c_{opt}\|$, as $\beta < 1/16$. In particular,

$$\|p c_i\| \leq x_p + y_p \leq (1 + \varepsilon/4) \|p c_{opt}\| .$$

Namely, if we move our center from c_{opt} to c_i, the error generated by points inside U_i is smaller than $\mathrm{med}_{\mathrm{opt}} \varepsilon/4$.

Thus, if the number of points in $Q_i = P \setminus U_i$ is smaller than $n\varepsilon/4$, then we are done. As the maximum error encountered for a point of Q_i when moving the center from c_{opt} to c_i is at most $2R$.

Thus, it must be that $|Q_i| \geq n\varepsilon/4$. We now perform a round of random sampling until we pick a point that is in Q_i. Let $s_j \in Q_i$ be this sample point, where $j > i$. Clearly, the line l connecting c_i to s_j must belong to F_j, as $c_i \in H_i \subset H_j$. Now, the angle between l and $\ell_i = \mathrm{line}(c_i, c_{opt})$ is smaller than $\pi/2 - \beta$. Namely,

$$\|c_{opt} l\| \leq \|c_{opt} c_i\| \sin(\pi/2 - \beta) = \|c_{opt} c_i\| \cos(\beta) \leq (1 - \beta^2/4) \|c_{opt} c_i\| .$$

Thus, after each round, the distance between F_i and c_{opt} shrinks by a factor of $(1-\beta^2/4)$. Namely, either we are close to the optimal center, or alternatively, we make reasonable progress in each round.

In the first round, we picked a point s_u such that $\|s_u c_{opt}\| \leq 2R$. Either during our sampling, we had $\|F_i c_{opt}\| \leq \varepsilon R$, or alternatively, we had reduced in each round the distance between our sample flat and c_{opt} by a factor of $(1 - \beta^2/4)$. On the other hand, once this distance drops below εR, we stop, as we had found a point that belongs to H_i and provide a $(1+\varepsilon)$-approximate solution. Furthermore, as long as $|Q_i| \geq \varepsilon n/4$, the probability of success is at least $\varepsilon/4$. Thus, the expected number of samples in a round until we pick a point of Q_i (and thus terminating the i-th round) is $\lceil 4/\varepsilon \rceil$. The number of rounds we need is

$$M = \left\lceil \log_{1-\beta^2/4} \frac{\varepsilon}{2} \right\rceil = \left\lceil \frac{\log(\varepsilon/2)}{\log(1-\beta^2/4)} \right\rceil = O\left(\frac{1}{\varepsilon^2} \log \frac{2}{\varepsilon} \right).$$

Let X be the random variable which is the number of random samples till we get M successes. Clearly, $E[X] = O(1/\varepsilon^3 \log 1/\varepsilon)$. It follows, by the Markov inequality, that with constant probability, if we sample $O(1/\varepsilon^3 \log(1/\varepsilon))$ points, then those points span a subspace that contains a $(1+\varepsilon)$-approximate 1-median center. ■

We are next interested in solving the k-median problem for a set of points in $\mathbb{R}^d$. We first normalize the point-set.

Lemma 4.0.6 *Given a point-set P in $\mathbb{R}^d$, and a parameter k, one can can scale-up space and compute a point-set P', such that: (i) The distance between any two points in P' is at least one. (ii) The optimal k-median cost of the modified data set is at most n^b for $b = O(1)$, where $n = |P|$. (iii) The costs of any k-median solutions in both (old and modified) data sets are the same up to a factor of $(1 + \varepsilon/5)$. This can be done in $O(nkd)$ time.*

Proof: Observe that by using Gonzalez [10] 2-approximation algorithm for the k-center clustering, we can compute in $O(nkd)$ time a value L (the radius of the approximate k-center clustering), such that $L/2 \leq \mathrm{med}_{\mathrm{opt}}(P, k) \leq nL$.

We cover space by a grid of size $L\varepsilon/(5nd)$, and snap the points of P to this grid. After scaling, this is the required point-set. ■

From this point on, we assume that the given point-set is normalized.

Theorem 4.0.7 *Let P be a normalized set of n points in $\mathbb{R}^d$, $1 > \varepsilon > 0$, and let R be a random sample of $O(1/\varepsilon^3 \log 1/\varepsilon)$ points from P. Then one can compute, in $O\left(d2^{O(1/\varepsilon^4)} \log n\right)$ time, a point-set $S(P, R)$ of cardinality $O\left(2^{O(1/\varepsilon^4)} \log n\right)$, such that with constant probability (over the choice of R), there is a point $q \in S(P, R)$ such that cost$(q, P) \leq (1 + \varepsilon)\mathrm{med}_{\mathrm{opt}}(P, 1)$.*

Proof: Let's assume that we had found a t such that $t/2 \leq \mathrm{AvgMed}(P, 1) \leq t$. Clearly, we can find such a t by checking all possible values of $t = 2^i$, for $i = 0, \ldots, O(\log n)$, as P is a normalized point-set (see Lemma 4.0.6).

Next, by Theorem 4.0.5, we know that with constant probability, there is a point of R with distance $\leq 2\,\mathrm{AvgMed}(P, 1) \leq 2t$ from the optimal 1-median center c_{opt} of P. Let $H = \mathrm{span}(R)$ denote the affine subspace spanned by R. For each point of $p \in R$, we construct a grid $G_p(t)$ of side $\varepsilon t/(10|R|) = O(t\varepsilon^4 \log(1/\varepsilon))$ centered at p on H, and let $B(p, 3t)$ be a ball of radius $2t$ centered at p. Finally, let $S'(p, t) = G_p(t) \cap B(p, 3t)$. Clearly, if $t/2 \leq \mathrm{AvgMed}(P, 1) \leq t$, and $\|pc_{opt}\| \leq 2t$, then there is a point $q \in S'(p, t)$ such that $\mathrm{cost}(q, P) \leq (1 + \varepsilon)\mathrm{med}_{\mathrm{opt}}(P, 1)$.

Let $S(P, R) = \displaystyle\bigcup_{i=0}^{O(\log n)} \bigcup_{p \in R} S'(p, 2^i)$. Clearly, $S(P, R)$ is the required point-set, and furthermore,

$$|S(P, R)| = O\left((\log n)\,|R| \left(\frac{1}{\varepsilon^4} \log \frac{1}{\varepsilon}\right)^{O(|R|)}\right) = O\left(2^{O(1/\varepsilon^3 \log^2 \frac{1}{\varepsilon})} \log n\right) = O\left(2^{O(1/\varepsilon^4)} \log n\right). \quad ■$$

Theorem 4.0.8 *For any point-set $P \subset \mathbb{R}^d$ and $0 < \varepsilon < 1$, a $(1 + \varepsilon)$-approximate 2-median for P can be found in $O(2^{(1/\varepsilon)^{O(1)}} d^{O(1)} n \log^{O(1)} n)$ expected time, with high-probability.*

Proof: In the following, we assume that the solution is irreducible, i.e., removing a median creates a solution with cost at least $1 + \Omega(\varepsilon)$ times the optimal. Otherwise, we can focus on solving the 1-median instead.

Let c_1, c_2 be the optimal centers and P_1, P_2 be the optimal clusters. Without loss of generality we assume that $|P_1| \geq |P_2|$. The algorithm proceeds by considering whether P_2 is large or small when compared with the size of P_1. In both cases the algorithm returns an approximate solution with constant probability. By exploring both cases in parallel and repeating the computation several times we can achieve an arbitrarily large probability of success.

Case 1: $|P_1| \geq |P_2| \geq |P_1|\varepsilon$. In this case we sample a random set of points R of cardinality $O(1/\varepsilon^4 \log 1/\varepsilon)$. We now exhaustively check all possible partitions of R into $R_1 = P_1 \cap R$ and $R_2 = P_2 \cap R$ (there are $O(2^{O(1/\varepsilon^4 \log 1/\varepsilon)})$ such possibilities). For the right such partition, R_i is a random sample of points in P_i of cardinality $\Omega(1/\varepsilon^3 \log 1/\varepsilon)$ (since $E[|R \cap P_i|] = \Omega(1/\varepsilon^3 \log 1/\varepsilon)$). By Theorem 4.0.7, we can generate point-sets S_1, S_2 that with constant probability contain $c'_1 \in S_1, c'_2 \in S_2$, such that $\mathrm{cost}(c'_1 \vee c'_2, P) \leq (1+\varepsilon)\mathrm{med}_{\mathrm{opt}}(P, 2)$. Checking each such pair c'_1, c'_2 takes $O(nd)$ time, and we have $O(|S_1||S_2|)$ pairs. Thus the total running time is $O\left(nd2^{O(1/\varepsilon^4 \log 1/\varepsilon)} \log^2 n\right)$.

Case 2: $|P_1|\varepsilon > |P_2|$. In this case we proceed as follows. First, we sample a set R of $\lambda = O(1/\varepsilon^3 \log 1/\varepsilon)$ points from P_1. This can be done just by sampling λ points from P, since with probability $2^{-O(1/\varepsilon^3 \log 1/\varepsilon)}$ such a sample contains only points from P_1; we can repeat the whole algorithm several times to obtain a constant probability of success. Next, using Theorem 4.0.7, we generate a set $\mathcal{C}_1$ of candidates to be center points of the cluster P_1. In the following, we check all possible centers $c'_1 \in \mathcal{C}_1$. With constant probability, there exists $c'_1 \in \mathcal{C}_1$ such that $\mathrm{cost}(c'_1, P_1) \leq (1+\varepsilon/3)\mathrm{cost}(c_1, P_1)$.

Let (P'_1, P'_2) denote optimal 2-median clustering induced by median c'_1 (as above), and let c'_2 denote the corresponding center of P'_2. We need to find c''_2 such that $\mathrm{cost}(c'_1 \vee c''_2, P) \leq (1+\varepsilon/3)\mathrm{cost}(c'_1 \vee c'_2, P) \leq (1+\varepsilon)\mathrm{med}_{\mathrm{opt}}(P, 2)$. In order to do that, we first remove some elements from P_1, in order to facilitate random sampling from P_2.

First, observe that $\mathrm{cost}(c'_1, P'_2) \leq |P'_2| \cdot \|c'_2 c'_1\| + \mathrm{cost}(c'_2, P'_2)$ and therefore we can focus on the case when $|P'_2| \cdot \|c'_2 c'_1\|$ is greater than $O(\varepsilon) \cdot \mathrm{cost}(c'_1 \vee c'_2, P)$, since otherwise $c'_2 = c'_1$ would be a good enough solution.

We exhaustively search for the value of two parameters (guesses) $t, \mathcal{U}$, such that $t/2 \leq \|c'_1 c'_2\| \leq t$ and $\mathcal{U}/2 \leq \mathrm{med}_{\mathrm{opt}}(P, 2) \leq \mathcal{U}$. Since P is normalized this would require checking $O(\log^2 n)$ possible values for t and $\mathcal{U}$. If $t > 4\mathcal{U}$, then $t > 4\mathrm{med}_{\mathrm{opt}}(P, 2)$ and for any $p, q \in P_i$ we have $\|pq\| \leq \mathcal{U}$. Moreover, for any $p \in P_1, q \in P_2$ we have $\|pq\| \geq \|c'_1 c'_2\| - \|c'_1 p\| - \|c'_1 q\| > 2\mathcal{U}$. Thus, take all the points in distance $\leq 2\mathcal{U}$ from c'_1 to be in P'_1, and take all the other points to be in P_2'. The problem is thus solved, as we partitioned the points into the correct clusters, and can compute an approximated 1-median for each one of them directly.

Otherwise, $t \leq 4\mathcal{U}$ and let $S = \left\{ p \,\middle|\, \|pc_1'\| \leq t/4 \right\}$. Clearly, $S \subset P_1'$. Moreover, we claim that $|P_2'| \geq \varepsilon |P_1' \setminus S|$, since otherwise we would have

$$|P_2'| \, \|c_2'c_1'\| \leq \varepsilon |P_1' \setminus S| \, \|c_2'c_1'\|$$

and

$$\mathrm{cost}(c_1', P_1' \setminus S) \geq \frac{t}{4} |P_1' \setminus S| \geq \frac{\|c_2'c_1'\|}{8} |P_1' \setminus S|.$$

Thus, $|P_2'| \, \|c_2'c_1'\| \leq 8\varepsilon \mathrm{cost}(c_1', P_1' - S)$ and thus $\mathrm{cost}(c_1', P) \leq (1 + 8\varepsilon)\mathrm{cost}(c_1' \vee c_2', P)$. This implies that we can solve the problem in this case by solving the 1-median problem on P, thus contradicting our assumption.

Thus, $|P_2'| \geq \varepsilon |P_1' \setminus S|$. We create $P' = P \setminus S = P_1'' \cup P_2'$, where $P_1'' = P_1' \setminus S$. Although P_1'' might now be considerably smaller than P_2'', and as such case 1 does not apply directly. We can overcome this by adding enough copies of c_1' into P_1'', so that it would be of size similar to P_2'.

To carry that out, we again perform an exhaustive enumeration of the possible cardinality of P_2' (up to a factor of 2). This requires checking $O(\log n)$ possibilities. Let $\mathcal{V}$ be the guess for the cardinality of P_2', such that $\mathcal{V} \leq |P_2'| \leq 2\mathcal{V}$.

We add $\mathcal{V}$ copies of c_1' to P_1''. We can now apply the algorithm for the case when the cardinalities of both clusters are comparable, as long as we ensure that the algorithm reports c_1' as one of the medians. To this end, it is not difficult to see that by adding copies of c_1' to P_1'' we also ensured that for any 2 medians x and y, replacing at least one of them by c_1' yields a better solution. Therefore, without loss of generality we can assume that the algorithm described above, when applied to P', reports c_1' as one of the medians. The complexity of the algorithm is as stated. ■

Theorem 4.0.9 *For any point-set $P \subset \mathbb{R}^d$, $\varepsilon < 1$, and a parameter k, a $(1+\varepsilon)$-approximate k-median for P can be found in $2^{(k/\varepsilon)^{O(1)}} d^{O(1)} n \log^{O(k)} n$ expected time, with high-probability.*

Proof: We only sketch the extension of the algorithm of Theorem 4.0.8 for $k > 2$. As before, we observe that large clusters of cardinality $\geq \varepsilon n/k$ can be handled directly by random sampling and exhaustive enumeration of all partitions of the samples into the different clusters, and using Theorem 4.0.7. Thus, we can focus on the case when there is at least one small cluster.

Let $C_1, \ldots, C_k$ be the clusters in the optimal solutions, with corresponding centers $c_1, \ldots, c_k$. Let $C_1, \ldots, C_i$ be the heavy clusters, and let C_{i+1} be a small cluster, such that its center c_{i+1} is the closest one to $c_1, \ldots, c_i$.

We use an argument, similar to the argument used in Theorem 4.0.8, to show that we can shrink $C_1, \ldots, C_i$ to make their sizes comparable to C_{i+1}.

- Let $\mathrm{AvgMed} = \mathrm{AvgMed}(C_1 \cup \ldots \cup C_i, i)$. If the distance from any c_j, $j > i$ to the nearest $c_1, \ldots, c_i$ is less than $t \leq \mathrm{AvgMed} \leq 2t$, then we can remove all such medians c_j without incurring much cost, as $|C_j| \leq n\varepsilon/k$.

- Otherwise, this means the medians c_j, $j > i$, are at least at distance $\mathrm{AvgMed}/2$ from $c_1, \ldots, c_i$.

- On the other hand, c_{i+1} cannot be too far from $c_1, \ldots, c_i$, because then we could easily separate the points of $C_1, \ldots, C_i$ from the points of $C_{i+1}, \ldots, C_k$.

- Thus, we can "guess" (i.e., enumerate $O(\log n)$ possibilities), up to a factor of two, the distance between c_{i+1} and its closest neighbor in $c_1, \ldots, c_i$. Let t be this guess.

- We can assume that all points within distance $< t/2$ to $c_1, \ldots, c_i$ belong to clusters $C_1, \ldots, C_i$, and focus on clustering the remaining set of points.

- The number of points with distance $> t/2$ from $c_1, \ldots, c_k$ is comparable to the size of C_{i+1}. Thus we can proceed with sampling.

This yields a recursive algorithm that gets rid of one cluster in each recursive call. It performs $2^{(k/\varepsilon)^{O(1)}} \log^{O(1)} n$ recursive calls in each node, and the recursion depth is k. Thus, the algorithm has the running time stated. Note that we need to rerun this algorithm $O\left(2^{O(k)} \log n\right)$ times to get high probability results. ■

Chapter 5

Smaller Core-Sets for Balls

5.1 Introduction

Given a set of points $P \subset R^d$ and value $\epsilon > 0$, a *core-set* $S \subset P$ has the property that the smallest ball containing S is within ϵ of the smallest ball containing P. That is, if the smallest ball containing S is expanded by $1+\epsilon$, then the expanded ball contains P. It is a surprising fact that for any given ϵ there is a core-set whose size is independent of d, depending only on ϵ. This is was shown by Bădoiu *et al.*[6], where applications to clustering were found, and the results have been extended to k-flat clustering.[13].

While the previous result was that a core-set has size $O(1/\epsilon^2)$, where the constant hidden in the O-notation was at least 64, here we show that there are core-sets of size at most $2/\epsilon$. This is not so far from a lower bound of $1/\epsilon$, which is easily shown by considering a regular simplex in $1/\epsilon$ dimensions. Such a bound is of particular interest for k-center clustering, where the core-set size appears as an exponent in the running time.

Our proof is a simple effective construction. We also give a simple algorithm for computing smallest balls, that looks something like gradient descent; this algorithm serves to prove a core-set bound, and can also be used to prove a somewhat better core-set bound for k-flats. Also, by combining this algorithm with the construction of the core-sets, we can compute a 1-center in time $O(dn/\epsilon + (1/\epsilon)^5)$.

In the next section, we prove the core-set bound for 1-centers, and then describe the gradient-descent algorithm.

5.2 Core-sets for 1-centers

Given a ball B, let c_B and r_B denote its center and radius, respectively. Let $B(P)$ denote the 1-center of P, the smallest ball containing it.

We restate the following lemma, proved in [9]:

Lemma 5.2.1 *If $B(P)$ is the minimum enclosing ball of $P \subset \mathbb{R}^d$, then any closed half-space that contains the center $c_{B(P)}$ also contains a point of P that is at distance $r_{B(P)}$ from $c_{B(P)}$.*

Theorem 5.2.2 *There exists a set $S \subseteq P$ of size at most $2/\epsilon$ such that the distance between $c_{B(S)}$ and any point p of P is at most $(1+\epsilon)r_{B(P)}$.*

Proof: We proceed in the same manner as in [6]: we start with an arbitrary point $p \in P$ and set $S_0 = \{p\}$. Let $r_i \equiv r_{B(S_i)}$ and $c_i \equiv c_{B(S_i)}$. Take the point $q \in P$ which is furthest away from c_i and add it to the set: $S_{i+1} \leftarrow S_i \bigcup \{q\}$. Repeat this step $2/\epsilon$ times. It is enough to show that the maximum distance from one of the centers c_i to the points of P is at most $\hat{R}$.

Let $c \equiv c_{B(P)}$, $R \equiv r_{B(P)}$, $\hat{R} \equiv (1+\epsilon)R$, $\lambda_i \equiv r_i/R$, $d_i \equiv ||c - c_i||$ and $K_i \equiv ||c_{i+1} - c_i||$. Since the radius of the minimum enclosing ball is R, there is at least one point $q \in P$ such that $||q - c_i|| \geq R$. If $K_i = 0$ then we are done, since the maximum distance from c_i to any point is at most R. If $K_i > 0$, let H be the hyperplane that contains c_i and is orthogonal to (c_i, c_{i+1}). Let H^+ be the closed half-space bounded by H that does not contain c_{i+1}. By Lemma 6.3.1, there must be a point $p \in S_i \bigcap H^+$ such that $||c_i - p|| = r_i = \lambda_i R$, and so $||c_{i+1} - p|| \geq \sqrt{\lambda_i^2 R^2 + K_i^2}$. Therefore,

$$\lambda_{i+1} R \geq \max(R - K_i, \sqrt{\lambda_i^2 R^2 + K_i^2}) \tag{5.1}$$

We want a lower bound on λ_{i+1} that depends only on λ_i. Observe that the bound on λ_{i+1} is smallest with respect to K_i when

$$\begin{aligned} R - K_i &= \sqrt{\lambda_i^2 R^2 + K_i^2} \\ R^2 - 2K_i R + K_i^2 &= \lambda_i^2 R^2 + K_i^2 \\ K_i &= \frac{(1-\lambda_i^2)R}{2} \end{aligned}$$

Using (5.1) we get that

$$\lambda_{i+1} \geq \frac{R - \frac{(1-\lambda_i^2)R}{2}}{R} = \frac{1+\lambda_i^2}{2} \tag{5.2}$$

Substituting $\gamma_i = \frac{1}{1-\lambda_i}$ in the recurrence (5.2), we get $\gamma_{i+1} = \frac{\gamma_i}{1-1/(2\gamma_i)} = \gamma_i(1 + \frac{1}{2\gamma_i} + \frac{1}{4\gamma_i^2} \dots) \geq \gamma_i + 1/2$. Since $\lambda_0 = 0$, we have $\gamma_0 = 1$, so $\gamma_i \geq 1 + i/2$ and $\lambda_i \geq 1 - \frac{1}{1+i/2}$. That is, to get $\lambda_i > 1 - \epsilon$, it's enough that $1 + i/2 \geq 1/\epsilon$, or enough that $i \geq 2/\epsilon$. ■

5.3 Simple algorithm for 1-center

The algorithm is the following: start with an arbitrary point $c_1 \in P$. Repeat the following step $1/\epsilon^2$ times: at step i find the point $p \in P$ farthest away from c_i, and move toward p as follows: $c_{i+1} \leftarrow c_i + (p - c_i)\frac{1}{i+1}$.

Claim 5.3.1 *If $B(P)$ is the 1-center of P with center $c_{B(P)}$ and radius $r_{B(P)}$, then $||c_{B(P)} - c_i|| \leq r_{B(P)}/\sqrt{i}$ for all i.*

Proof: Proof by induction: Let $c \equiv c_{B(P)}$. Since we pick c_1 from P, we have that $||c - c_1|| \leq R \equiv r_{B(P)}$. Assume that $||c - c_i|| \leq R/\sqrt{i}$. If $c = c_i$ then in step i we move away from c by at most $R/(i+1) \leq R/\sqrt{i+1}$, so in that case $||c - c_{i+1}|| \leq R/\sqrt{i+1}$. Otherwise, let H be the hyperplane orthogonal to (c, c_i) which contains c. Let H^+ be the closed half-space bounded by H that does not contain c_i and let $H^- = \mathbb{R} \setminus H^+$. Note that the furthest point from c_i in $B(P) \bigcap H^-$ is at distance less than $\sqrt{||c_i - c||^2 + R^2}$ and we can conclude that for every point $q \in P \bigcap H^-$, $||c_i - q|| < \sqrt{||c_i - c||^2 + R^2}$. By Lemma 6.3.1 there exists a point $q \in P \bigcap H^+$ such that $||c_i - q|| \geq \sqrt{||c_i - c||^2 + R^2}$. This implies that $p \in P \bigcap H^+$. We have two cases to consider:

- if $c_{i+1} \in H^+$, by moving c_i towards c we only increase $||c_{i+1} - c||$, and as noted before if $c_i = c$ we have $||c_{i+1} - c|| \leq R/(i+1) \leq R/\sqrt{i+1}$. Thus, $||c_{i+1} - c|| \leq R/\sqrt{i+1}$
- if $c_{i+1} \in H^-$, by moving c_i as far away from c and p on the sphere as close as possible to H^-, we only increase $||c_{i+1} - c||$. But in this case, (c, c_{i+1}) is orthogonal to (c_i, p) and we have $||c_{i+1} - c|| = \frac{R^2/\sqrt{i}}{R\sqrt{1+1/i}} = R/\sqrt{i+1}$.

■

Chapter 6

Optimal Core-Sets for Balls

6.1 Introduction

In the previous chapter we showed that there are core-sets of size at most $2/\epsilon$, but the worst-case lower bound, easily shown by considering regular simplices, is only $\lceil 1/\epsilon \rceil$.[4] In this chapter we show that the lower bound is tight: there are always ϵ-core-sets of size $\lceil 1/\epsilon \rceil$. A key lemma in the proof of the upper bound is the fact that the bound for Löwner-John ellipsoid pairs is tight for simplices.

The existence proof for these optimal core-sets is an algorithm that repeatedly tries to improve an existing core-set by local improvement: given $S \subset P$ of size k, it tries to swap a point out of S, and another in from P, to improve the approximation made by S. Our proof shows that a $1/k$-approximate ball can be produced by this procedure. (That is, if the smallest ball containing the output set is expanded by $1 + 1/k$, the resulting ball contains the whole set.) While it is possible to bound the number of iterations of the procedure for a slightly sub-optimal bound, such as $1/(k-1)$, no such bound was found for the optimal case. However, we give experimental evidence that for random pointsets, the algorithm makes no change at all in the core-sets produced by the authors' previous procedure, whose guaranteed accuracy is only $2/k$. That is, the algorithm given here serves as a fast way of verifying that the approximation ϵ is $1/k$, and not just $2/k$.

We also consider an alternative local improvement procedure, with no performance guarantees, that gives a better approximation accuracy, at the cost of considerably longer running time.

Some notation: Given a ball B, let c_B and r_B denote its center and radius, respectively. Let

$B(P)$ denote the 1-center of P, the smallest ball containing it.

The next two sections give the lower and upper bounds, respectively; The experimental results are given in the last section.

6.2 A Lower Bound for Core-Sets

Theorem 6.2.1 *Given $\epsilon > 0$, there is a pointset P such that any ϵ-core-set of P has size at least $\lceil 1/\epsilon \rceil$.*

Proof: We can take P to be a regular simplex with $d + 1$ vertices, where $d \equiv \lfloor 1/\epsilon \rfloor$. A convenient representation for such a simplex has vertices that are the natural basis vectors $e_1, e_2, \ldots, e_{d+1}$ of $\mathbb{R}^{d+1}$, where e_i has the i'th coordinate equal to 1, and the remaining coordinates zero. Let core-set S contain all the points of P except one point, say e_1. The circumcenter of the simplex is $(1/(d+1), 1/(d+1), \ldots, 1/(d+1))$, and its circumradius is

$$R \equiv \sqrt{(1 - 1/(d+1))^2 + d/(d+1)^2} = \sqrt{d/(d+1)}.$$

The circumcenter of the remaining points is $(0, 1/d, 1/d, \ldots, 1/d)$, and the distance R' of that circumcenter to e_1 is

$$R' = \sqrt{1 + d/d^2} = \sqrt{1 + 1/d}.$$

Thus

$$R'/R = 1 + 1/d = 1 + 1/\lfloor 1/\epsilon \rfloor \geq 1 + \epsilon,$$

with equality only if $1/\epsilon$ is an integer. The theorem follows. ■

6.3 Optimal Core-Sets

In this section, we show that there are ϵ-core-sets of size at most $\lceil 1/\epsilon \rceil$. The basic idea is to show that the pointset for the lower bound, the set of vertices of a regular simplex, is the worst case for core-set construction.

We can assume, without loss of generality, that the input set is the set of vertices of a simplex; this follows from the condition that the 1-center of P is determined by a subset $P' \subset P$ of size at most $d+1$: that is, the minimum enclosing ball of P is bounded by the circumscribed sphere of P'. Moreover, the circumcenter of P' is contained in the convex hull of P. That is, the problem of core-set construction for P is reduced to the problem of core-set construction for a simplex $T = \operatorname{conv} P'$, where the minimum enclosing ball $B(T)$ is its circumscribed sphere.

We will need the following lemma, proven in [9].

Lemma 6.3.1 *If $B(P)$ is the minimum enclosing ball of $P \subset \mathbb{R}^d$, then any closed half-space that contains the center $c_{B(P)}$ also contains a point of P that is at distance $r_{B(P)}$ from $c_{B(P)}$. It follows that for any point q at distance K from $c_{B(P)}$, there is a point q' of P at distance at least $\sqrt{r^2_{B(P)} + K^2}$ from q.*

Lemma 6.3.2 *Let B' be the largest ball contained in a simplex T, such that B' has the same center as the minimum enclosing ball $B(T)$. Then*

$$r_{B'} \leq r_{B(T)}/d.$$

Proof: We want an upper bound on the ratio $r_{B'}/r_{B(T)}$; consider a similar problem related to ellipsoids: let $e(T)$ be the maximum volume ellipsoid inside T, and $E(T)$ be the minimum volume ellipsoid containing T. Then plainly

$$\frac{r^d_{B'}}{r^d_{B(T)}} \leq \frac{\operatorname{Vol}(e(T))}{\operatorname{Vol}(E(T))},$$

since the volume of a ball B is proportional to r^d_B, and $\operatorname{Vol}(e(T)) \geq \operatorname{Vol}(B')$, while $\operatorname{Vol}(E(T)) \leq \operatorname{Vol}(B(T))$. Since affine mappings preserve volume ratios, we can assume that T is a regular simplex when bounding $\operatorname{Vol}(e(T))/\operatorname{Vol}(E(T))$. When T is a regular simplex, the maximum enclosed ellipsoid and minimum enclosing ellipsoid are both balls, and the ratio of the radii of those balls is $1/d$. [15] (In other words, any simplex shows that the well-known bound for Löwner-John ellipsoid pairs is tight.[18]) Thus,

$$\frac{r^d_{B'}}{r^d_{B(T)}} \leq \frac{\operatorname{Vol}(e(T))}{\operatorname{Vol}(E(T))} \leq \frac{1}{d^d},$$

and so

$$\frac{r_{B'}}{r_{B(T)}} \leq \frac{1}{d},$$

as stated. ■

Lemma 6.3.3 *Any simplex T has a facet F such that $r^2_{B(F)} \geq (1 - 1/d^2) r^2_{B(T)}$.*

Proof: Consider the ball B' of the previous lemma. Let F be a facet of T such that B' touches F. Then that point of contact p is the center of $B(F)$, since p is the intersection of F with the line through $c_{B(T)}$ that is perpendicular to F. Therefore

$$r^2_{B(T)} = r^2_{B'} + r^2_{B(F)},$$

and the result follows using the previous lemma. ■

Next we describe a procedure for constructing a core-set of size $\lceil 1/\epsilon \rceil$.

As noted, we can assume that P is the set of vertices of a simplex T, such that the circumcenter $c_{B(T)}$ is in T. We pick an arbitrary subset P' of P of size $\lceil 1/\epsilon \rceil$. (We might also run the algorithm of [4] until a set of size $\lceil 1/\epsilon \rceil$ has been picked, but such a step would only provide a heuristic speedup.) Let $R \equiv r_{B(P)}$. Repeat the following until done:

- find the point a of P farthest from $c_{B(P')}$;
- if a is no farther than $R(1+\epsilon)$ from $c_{B(P')}$, then return P' as a core-set;
- Let P'' be $P \cup \{a\}$;
- find the facet F of conv P'' with the largest circumscribed ball;
- Let P' be the vertex set of F.

The first step (adding the farthest point a) will give an increased radius to $B(P'')$, while the second step (deleting the point $P'' \setminus \operatorname{vert} F$) makes the set P' more “efficient”.

Theorem 6.3.4 *Any point set $P \subset \mathbb{R}^d$ has an ϵ-core-set of size at most $\lceil 1/\epsilon \rceil$.*

Proof: Let r be the radius of $B(P')$ at the beginning of an iteration, and let r' be the radius of $B(P')$ if the iteration completes. We will show that $r' > r$.

Note that if $r \geq R(1-\epsilon^2)$, the iteration will exit successfully: applying Lemma 6.3.1 to $c_{B(P')}$ and $c_{B(P)}$ (with the latter in the role of "q"), we obtain that there is a point $q' \in P'$ such that

$$R^2 \geq ||c_{B(P)} - q'||^2 \geq r^2 + ||c_{B(P')} - c_{B(P)}||^2,$$

so that

$$\epsilon^2 R^2 \geq R^2 - r^2 \geq ||c_{B(P')} - c_{B(P)}||^2,$$

implying that $c_{B(P')}$ is no farther than ϵR to $c_{B(P)}$, and so $c_{B(P')}$ is no farther than $R(1+\epsilon)$ from any point of P, by the triangle inequality. We have, if the iteration completes, that

$$\begin{aligned} r^2 &< R(1-\epsilon^2) \leq \hat{R}^2 \frac{1-\epsilon^2}{(1+\epsilon)^2} \\ &= \hat{R}^2 \frac{1-\epsilon}{1+\epsilon}, \end{aligned} \tag{6.1}$$

where $\hat{R} \equiv R(1+\epsilon)$.

By reasoning as for the proof of Theorem 2.2 of chapter 5 [4], we have

$$r_{B(P'')} \geq \frac{\hat{R} + r^2/\hat{R}}{2}. \tag{6.2}$$

For completeness, the proof of this bound is the following: since a is at least $\hat{R}$ from the center $c(P')$, we know that

$$\begin{aligned} \hat{R} &= ||a - c(P')|| \\ &\leq ||c(P'') - a|| + ||c(P'') - c(P')|| \\ &\leq r_{B(P'')} + K, \end{aligned}$$

where $K \equiv ||c(P'') - c(P')||$, and by Lemma 6.3.1, there is a point $q' \in P'$ such that

$$||c(P'') - q'|| \geq \sqrt{r_{B(P)}^2 + K^2}.$$

Combining these two lower bounds on $r_{B(P'')}$ and minimizing with respect to K gives the bound (6.2).

We use (6.2) and the lower bound of the previous lemma on the size of $B(F)$ to obtain

$$r' \geq \frac{\hat{R} + r^2/\hat{R}}{2}\sqrt{1 - \frac{1}{\lceil 1/\epsilon \rceil^2}},$$

and so

$$\frac{r'}{r} \geq \frac{\hat{R}/r + r/\hat{R}}{2}\sqrt{1 - \epsilon^2}.$$

The right-hand side is decreasing in $r/\hat{R}$, and so, since from (6.1), $r < \hat{R}\sqrt{(1-\epsilon)/(1+\epsilon)}$, we have

$$\frac{r'}{r} > \frac{\sqrt{\frac{1-\epsilon}{1+\epsilon}} + \sqrt{\frac{1+\epsilon}{1-\epsilon}}}{2}\sqrt{1 - \epsilon^2} = 1.$$

Therefore $r' > r$ when an iteration completes. Since there are only finitely many possible values for r, we conclude that the algorithm successfully terminates with an ϵ-core-set of size $\lceil 1/\epsilon \rceil$. ■

6.4 Experimental Results

Some experimental results on the approximation ratios are shown in Figures 6-1 through 6-8, each for different dimensional random data and distributions. The ordinates are the sizes of the core-sets considered, and the abscissas are the percentage increase in radius needed to enclose the whole set, relative to the smallest enclosing sphere.

In the plots,

- (hot start) a plain line shows results for the algorithm given here, starting from the output of the previous algorithm guaranteeing a $2/k$-core-set;
- (old) a dashed line is for the previous algorithm guaranteeing a $2/k$-core-set;
- (random start) a bullet (•) is for the algorithm given here, starting from a random subset;
- (1-swap) a dot (.) is for an algorithm that is like the one given here, but that works a little harder: it attempts local improvement by swapping a point into the core-set, and another point out of the core-set. The possible points considered for swapping in are the three farthest from the circumcenter of the current core-set, while the points considered for

swapping out are those three whose individual deletion leaves the circumradius as large as possible.

Figures 6-9 through 6-16 show the number of iterations needed for the algorithms, using the same graphing scheme.

Note that the random-start algorithm often does as well or better as hot-start algorithm, although a small but non-trivial number of iterations are required, while often the hot-start algorithm needs few or no iterations: the optimal algorithm serves as a confirmation that the "old" algorithm returns a better result than guaranteed.

We also performed tests of the gradient-descent method described in [4]. The algorithm is quite simple: start with an arbitrary point $c_1 \in p$. Repeat the following step K times: at step i find the point $p \in P$ farthest away from the current center c_i and move towards p as follows: $c_{i+1} \leftarrow c_i + (p - c_i)\frac{1}{i+1}$. For $K = 1/\epsilon^2$, this algorithm produces a point which is at distance at most ϵ away from the true center. For this requirement, it can be shown that this algorithm is tight on the worst case for the case of a simplex. However, if we require that the farthest away point from the point produced is at distance at most $(1+\epsilon)R$, it is not clear if the analysis of the algorithm is tight. In fact, to our surprise, in our experiments the distance between the point produced and the farthest away point is 99.999% of the time under $(1+1/K)R$ and always under $(1 + 1.1/K)R$. We tested the algorithm under normal and uniform distributions. An empiric argument to try to explain this unexpected behaviour is the following: it has been noted that the algorithm picks most (but not all) of the points from a small subset in a repetitive way, i.e., for example one point can appear every $5 - 10$ iterations. Now, if you only pick 2 points A and B in an alternate way (A, B, A, B, ...), (i.e., subcase of the case when the solution is given by 2 points), the solution will converge quickly to the subspace spanned by A and B and it's easy to see that the error within the subspace will be at most $1/K$ after K steps. This empiric argument seems to give some intuition on why the algorithm give so much better error in practice. It may also be possible to prove this algorithm converges much faster theoretically.

Figures 6-17 through **??** show convergence results for the "gradient descent" algorithm. They show the percentage overestimate of the radius of the minimum enclosing ball, as a function of the number of iterations i. The first two figures show results for $d = 2, 3, 10, 100$, and 200, and the final figure shows the results for point distributed in an annulus with $d = 10$. Note that the error is often less than $1/i$ and never more than a small multiple of it.

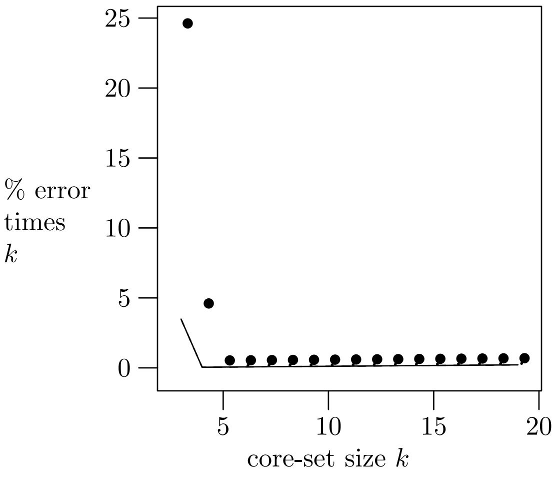

Figure 6-1: $d = 3$, normal

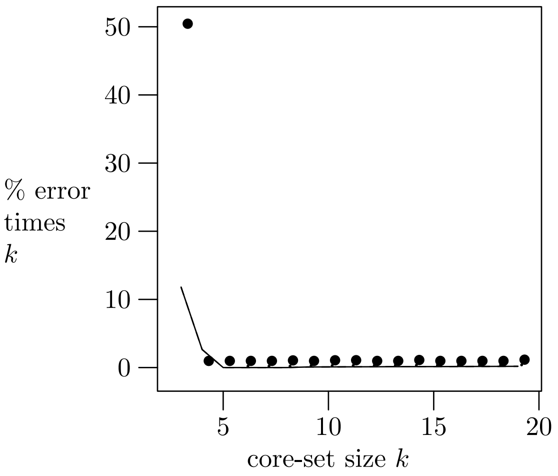

Figure 6-2: $d = 3$, uniform

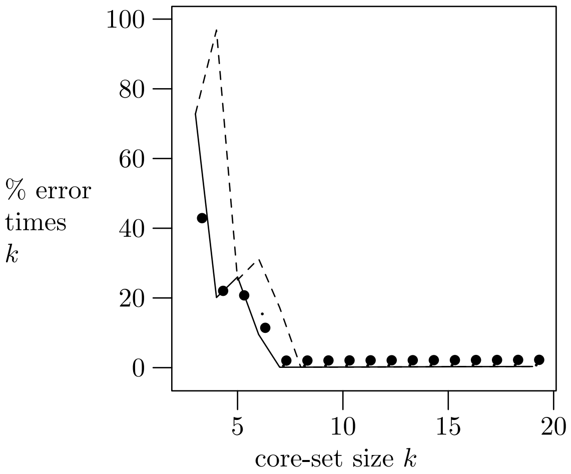

Figure 6-3: $d = 10$, normal

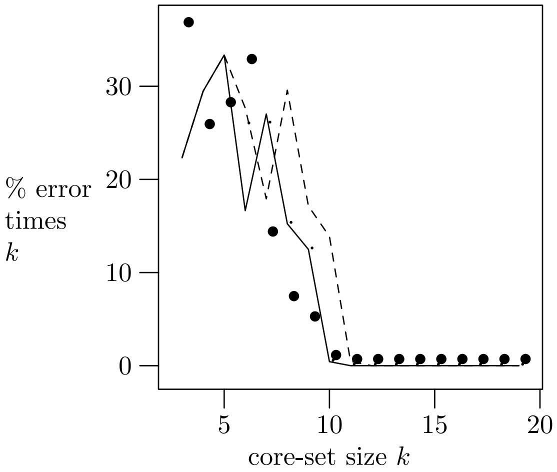

Figure 6-4: $d = 10$, uniform

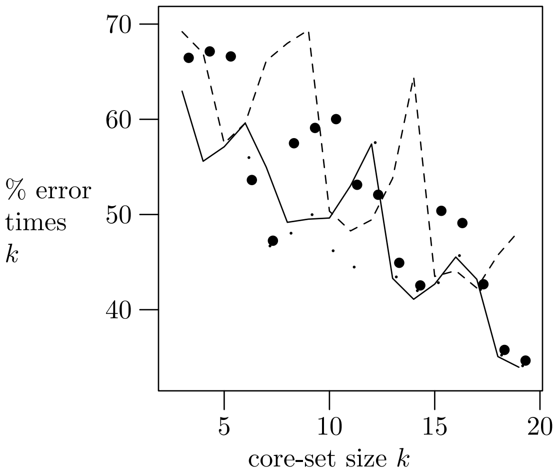

Figure 6-5: $d = 100$, normal

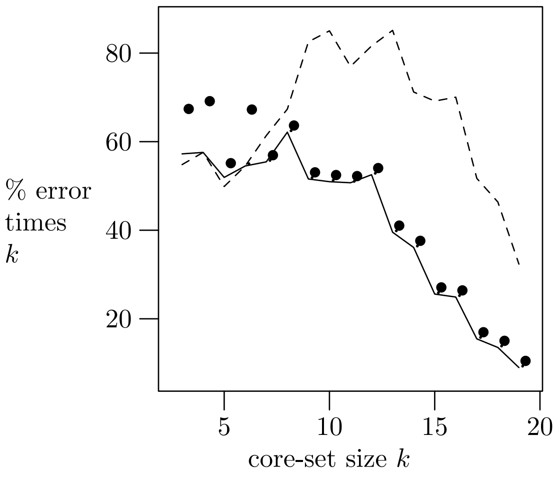

Figure 6-6: $d = 100$, uniform

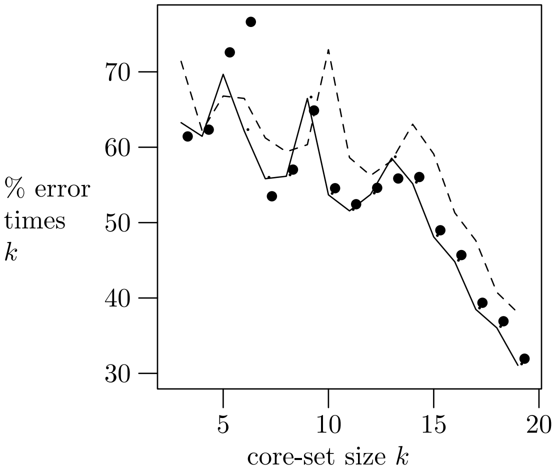

Figure 6-7: $d = 200$, normal

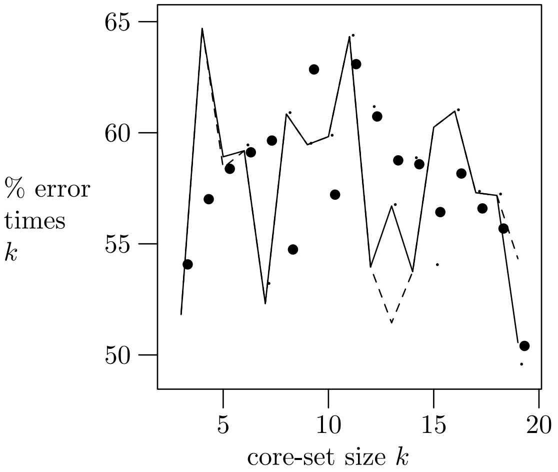

Figure 6-8: $d = 200$, uniform

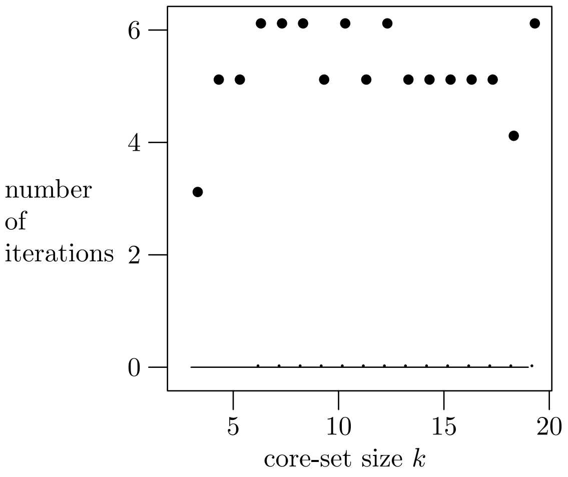

Figure 6-9: $d = 3$, normal

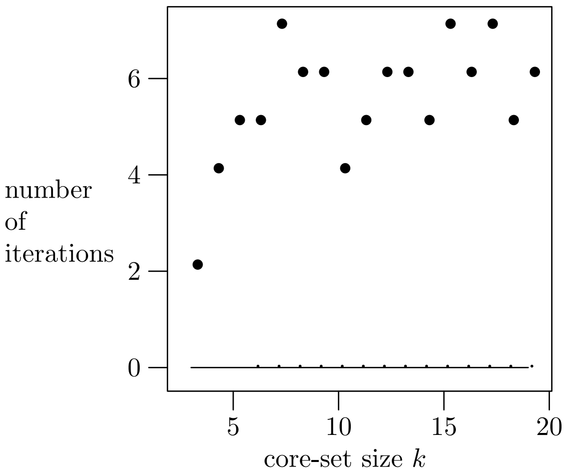

Figure 6-10: $d = 3$, uniform

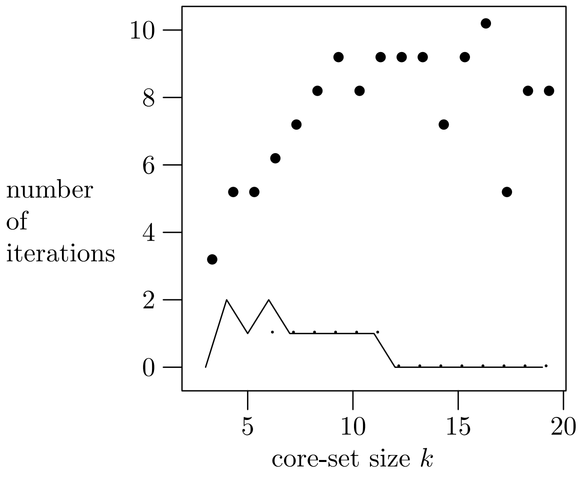

Figure 6-11: $d = 10$, normal

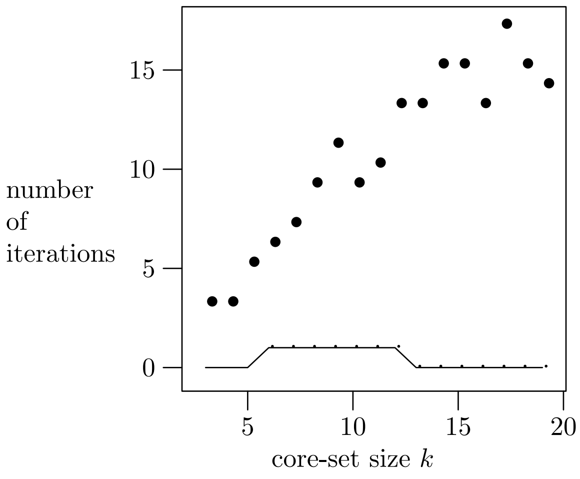

Figure 6-12: $d = 10$, uniform

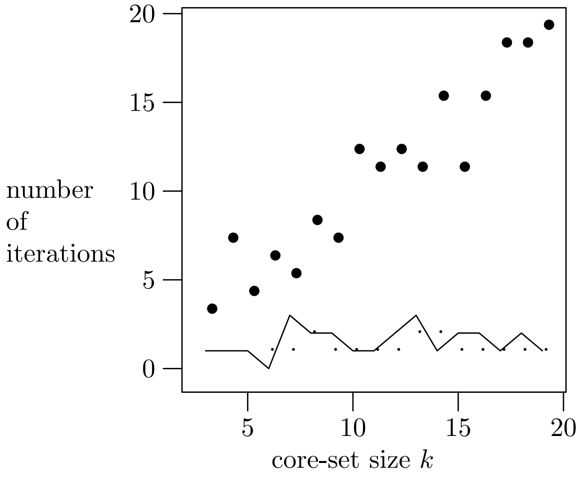

Figure 6-13: $d = 100$, normal

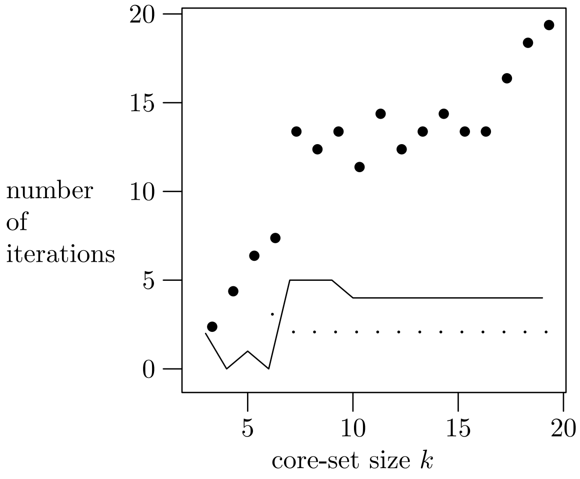

Figure 6-14: $d = 100$, uniform

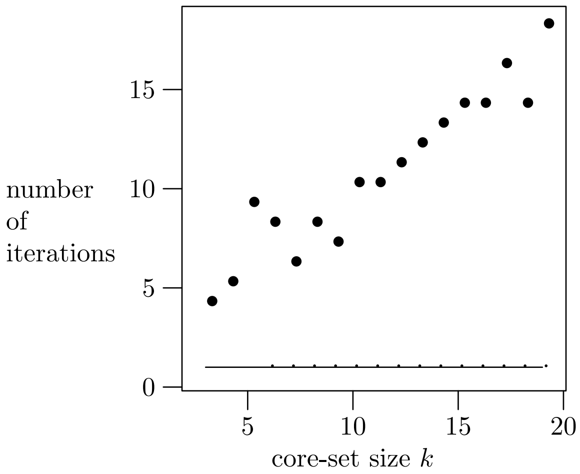

Figure 6-15: $d = 200$, normal

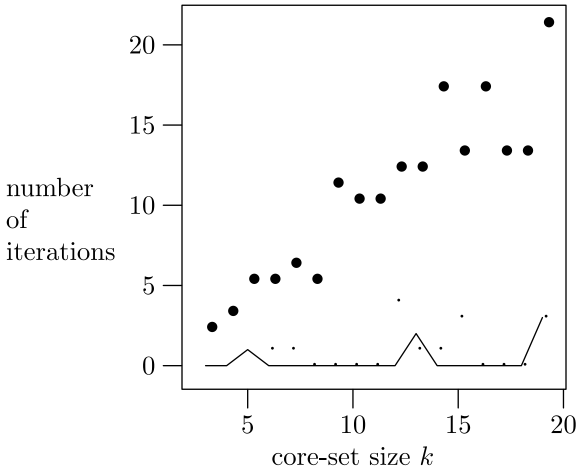

Figure 6-16: $d = 200$, uniform

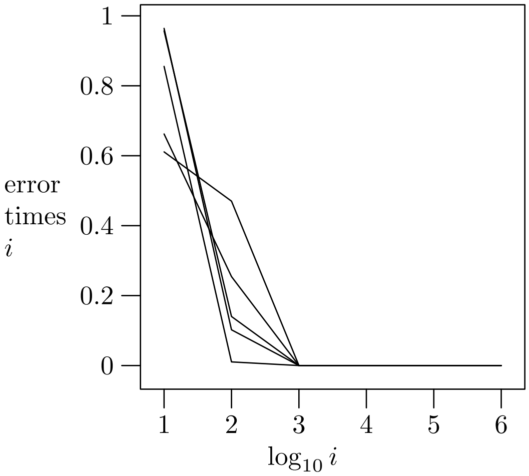

Figure 6-17: normal

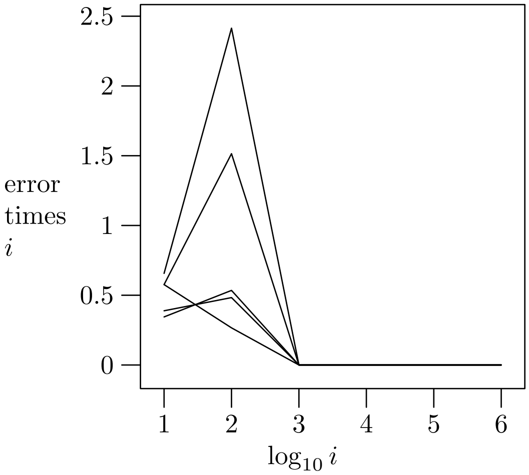

Figure 6-18: uniform

Bibliography

[1] Agarwal, P.K., and Procopiuc, C.M. , *Exact and approximation algorithms for clustering*, Algorithmica **33** (2002), no. 2, 201–226.

[2] Alon, N., Dar, S., Parnas, M., and Ron, D., *Testing of clustering*, Proc. 41th Annu. IEEE Sympos. Found. Comput. Sci., 2000, pp. 240–250.

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

[4] Bădoiu, M. and Clarkson, K.L., *Smaller Core-Sets for Balls*, Proc. 14th ACM-SIAM Sympos. on Discrete Algorithms, 2003, pp. 801–803.

[5] Bădoiu, M. and Clarkson, K.L., *Optimal Core-Sets for Balls*, submitted.

[6] Bădoiu M., Har-Peled S., and Indyk P., *Approximate clustering via core-sets*, Proc. 34th Sympos. on Theory of Computing, 2002, pp. 250–258.

[7] Duda, R.O., Hart, P.E., and Stork, D.G., "Pattern classification," 2nd ed., Wiley-Interscience, New York, 2001.

[8] Engebretsen, L., Indyk, P., and O'Donnell, R., *Derandomized dimensionality reduction with applications*, Proc. 13th ACM-SIAM Sympos. Discrete Algorithms, 2002.

[9] Goel, A., Indyk, P., and Varadarajan, K.R. *Reductions among high dimensional proximity problems*, Proc. 12th ACM-SIAM Sympos. Discrete Algorithms, 2001, pp. 769–778.

[10] Gonzalez, T., *Clustering to minimize the maximum intercluster distance*, Theoretical Computer Science, 1985, vol 38, pp. 293–306.

[11] Grötschel, M., Lovász, L., and Schrijver, A., "Geometric algorithms and combinatorial optimization," 2nd ed., Algorithms and Combinatorics, vol. 2, Springer-Verlag, Berlin Heidelberg, 1988, 2nd edition 1994.

[12] Har-Peled, S., *Clustering motion*, Proc. 42nd Annu. IEEE Sympos. Found. Comput. Sci., 2001, pp. 84–93.

[13] Har-Peled, S., and Varadarajan, K.R., *Approximate shape fitting via linearization*, Proc. 42nd Annu. IEEE Sympos. Found. Comput. Sci., 2001, pp. 66–73.

[14] Hochbaum, D., "Approximation algorithms for NP-hard problems," 1996.

[15] Howard, R., *The John Ellipsoid Theorem*, `http://www.math.sc.edu/~howard/Notes/app-convex-note.pdf`, 1997.

[16] Indyk, P., *Algorithmic applications of low-distortion geometric embeddings*, Proc. 42nd Annu. IEEE Sympos. Found. Comput. Sci., 2001, Tutorial, pp. 10–31.

[17] Indyk,P., and Thorup, M., *Approximate* 1*-median*, manuscript, 2000.

[18] John, F., *Extremum problems with inequalities as subsidiary conditions*, Studies and Essays Presented to R. Courant on his 60th birthday, 1948.

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

[20] Ostrovsky, R., and Rabani, Y., *Polynomial time approximation schemes for geometric k-clustering*, Proc. 41st Symp. Foundations of Computer Science, IEEE, Nov 2000, pp. 349–358.