Clustering Lines in High Dimensional Space: Classification of Incomplete Data

JIE GAO
Stony Brook University
MICHAEL LANGBERG
The Open University of Israel
LEONARD J. SCHULMAN
California Institute of Technology

A set of k balls $B_1, \ldots, B_k$ in a Euclidean space is said to cover a collection of lines if every line intersects some ball. We consider the k-center problem for lines in high dimensional space: Given a set of n lines $\mathcal{L} = \{\ell_1, \ldots, \ell_n\}$ in $\mathbb{R}^d$, find k balls of minimum radius which cover $\mathcal{L}$. We present a 2-approximation algorithm for the cases $k = 2, 3$ of this problem, having running time quasi-linear in the number of lines and the dimension of the ambient space.

Our result for 3-clustering is strongly based on a new result in discrete geometry which may be of independent interest: a Helly-type theorem for collections of axis-parallel "crosses" in the plane. The family of crosses does not have finite Helly number in the usual sense. Our Helly theorem is of a new type: it depends on ε-contracting the sets.

In statistical practice, data is often incompletely specified; we consider lines as the most elementary case of incompletely specified data points. Clustering of data is a key primitive in nonparametric statistics. Our results provide a way of performing this primitive on incomplete data, as well as imputing the missing values.

1. INTRODUCTION

A set of k balls $B_1, \ldots, B_k$ in a Euclidean space is said to cover a collection of lines if every line intersects some ball. We consider the following problem: Given a set of n lines $\ell_1, \ldots, \ell_n$ in $\mathbb{R}^d$, find k balls of minimum radius which cover the lines. We refer to this problem as the "k-center problem" (for lines). Loosely speaking, our main result is an $\tilde{O}(nd)$-time 2-approximation algorithm for this problem for the cases $k = 2, 3$.

This question is motivated by the challenge of coping with incomplete data in statistics. The usual starting point for statistical theory, learning theory, or estimation for control, is an input set consisting of a list of empirically gathered data points in $\mathbb{R}^d$. One of the serious gaps between statistical theory and practice, however, lies with incompletely-specified data. Essentially, the issue is that a high-dimensional data point is not specified by one "measurement" but by many, and that some of those measurements may be missing.

From a geometric viewpoint, a data item for which some measurements are missing is simply an affine subspace. In some cases this subspace is axis-parallel, for example in data obtained from sliding-bar surveys; however, in others it need not be. The latter appears for example in the context of sensor networks, where the individual data gathered may not have a mutually consistent orientation. In this work, we address one dimensional affine subspaces (i.e., lines, not necessarily axis-parallel), corresponding to a single degree of freedom in our ignorance of each item. Hence the bearing of the k-center problem on statistical inference.

The statistical literature on this subject is not on the same rigorous footing typical of other statistical literature. Incomplete data is coped with either by filling in missing entries in various ways, computing correlations only from available entries, or as a last resort (often the default in statistical packages), discarding incomplete items. Heuristics for filling in entries include "hot deck imputation" which substitutes for the missing entry the corresponding entry of the most similar data item; "mean substitution" in which the missing data is replaced with the sample mean; or replacing a missing element using a learning algorithm or criterion (EM, max likelihood). See [Little and Rubin 2002; Allison 2002]. This very heuristic state of affairs appears to be unavoidable since probabilistic inference requires some kind of external information about the correlations between the present and missing data. Quoting from [Allison 2002]: "...it is essential to keep in mind that these methods, like all the others, depend on certain easily violated assumptions for their validity. Not only that, there is no way to test whether the most crucial assumptions are satisfied." Under these circumstances, it is natural to use aggregate geometric information about the data to dictate the imputations—as has always been the case with regression in statistics. Our work suggests a particular method of imputation: given a "data line," find a ball intersecting it, and choose the point on the line closest to the center of that ball. We introduced this notion in our earlier work on the 1-center problem for lines [Gao et al. 2008]; the present work progresses to the much more difficult task of clustering, necessary for the analysis of inhomogeneous data.

Of course, we consider the problem of clustering lines to be a very natural one in computational geometry. Relevant literature is reviewed later.

Algorithmic results. The complexity of the task at hand increases rapidly with k. Indeed, the k-center problem for lines is at least as hard as the k-center problem

for points in Euclidean space, which cannot be approximated up to a factor of 1.822 in polynomial time unless $\mathbf{P} = \mathbf{NP}$ [Feder and Greene 1988]. There are simple polynomial-time algorithms for every fixed k, but the primary motivation for the problem lies in handling *huge* data sets in *high dimensional* Euclidean space. Therefore in order to be applicable in our applications, we focus on approximation algorithms whose running time is **quasilinear in n and d**, and try to make k as large as possible under this restriction. (Notice that a quasilinear time algorithm in n and d is almost the best one can hope for. The input size for n lines in d dimensional space is $\Omega(nd)$ so any algorithm dealing with these lines will have running time $\Omega(nd)$.) Throughout the paper, we do not restrict the size of d (*i.e.*, it may be the case that d is a function of n). We have found that it is not too hard to solve the case $k = 2$ but that $k = 3$ is challenging and has required, besides some nontrivial use of data structures, a new Helly-type theorem. It is an open question whether there is a quasilinear-time algorithm for $k = 4$. Our algorithmic results can be summarized in the following theorem. Given a set of lines $\mathcal{L}$, let r be the minimum radius k-clustering of $\mathcal{L}$. An α-approximation algorithm for the k-center problem returns a k-clustering of $\mathcal{L}$ of radius αr.

THEOREM 1.1. *For a set of n lines $\mathcal{L}$ in $\mathbb{R}^d$ and constant $\varepsilon > 0$, the $(2 + \varepsilon)$-approximate k-center problem can be solved in (randomized) time $O\left(nd \log 1/\varepsilon + n \log n \log 1/\varepsilon\right)$ for $k = 2$; and $O\left(nd \log 1/\varepsilon + \frac{n \log^2 n \log 1/\varepsilon}{\varepsilon}\right)$ for $k = 3$.*

A few remarks are in place. Notice that the approximation ratio appearing in the theorem is slightly larger than 2. This ε slackness is crucial for our analysis and follows (among other reasons) from our use of an *approximate* Helly-type theorem described below. An approximation ratio better than 2 can be obtained at the price of an increased running time which is exponential in d (but still quasilinear in n). Such exponential time algorithms follow from an easy extension of Theorem 1.1 or from standard "coreset" based techniques (discussed later in the Introduction). We present these algorithms in Section C of the Appendix. The algorithm of Theorem 1.1 is randomized (and succeeds with any constant probability approaching 1). The only part of our algorithm which is randomized is that described in Appendix B. Finally, we would like to note that the major obstacle in extending Theorem 1.1 to values of k greater than 3 lies in obtaining an extension of Theorem 1.3 (below) to higher dimension.

A Helly type theorem for "crosses". Our 3-center result in Theorem 1.1 is dependent on a new result in discrete geometry which may be of independent interest: an extension of Helly's theorem to certain non-convex sets. Helly's Theorem is one of the classical results in discrete geometry [Helly 1923]. It states that if every $d + 1$ convex sets in a set $\mathcal{S}$ of convex sets in $\mathbb{R}^d$ have non-empty intersection then all of the sets in $\mathcal{S}$ have non-empty intersection. Equivalently, if all the convex sets in $\mathcal{S}$ have empty intersection, then there is a subset $\mathcal{S}' \subset \mathcal{S}$ (a *witness*) of size $d + 1$ which also has empty intersection. In general, Helly's theorem establishes the locality of certain properties of convex sets. Over the years the basic Helly theorem has spawned numerous generalizations and variants, e.g., [Eckhoff 1993; Wenger 2004]. These Helly-type theorems usually have the following format: if every m members of a family of objects have property $\mathcal{P}$ then the entire family has property $\mathcal{P}$. In this work we study a certain extension of Helly theorem to objects that are not convex, but are limited unions of convex sets. The sets we study are

"crosses": the union of two axis-parallel strips in the plane. We remark that in existing literature, e.g., [Eckhoff 1993; Wenger 2004], Helly-type theorems exist for certain special cases which involve the union of convex sets. But these do not apply to our case of crosses.

Unfortunately—but not surprisingly—Helly's theorem fails badly for crosses. In Section 3 (Lemma 3.1) we show that a set of n crosses may have empty intersection, yet every $n-1$ of the crosses intersect.

Thus we slightly relax the property required from the witness $\mathcal{S}'$. Instead of requiring an empty intersection, we require that slight *contractions* of the crosses in $\mathcal{S}'$ have empty intersection. As this work addresses approximation algorithms for the 3-center problem, it turns out that such an *approximate* Helly-type theorem suffices for our needs. A few definitions, followed by our extension of Helly's theorem to crosses, are given below.

Let I and J be two intervals in $\mathbb{R}^1$. A cross $S = I \dagger J$ defined by I and J is the set of points $\{(p,q) \in \mathbb{R}^2 \,|\, p \in I$ or $q \in J\}$. The ε-contraction of an interval $I = [a,b]$ is $I^{-\varepsilon} = [a + \varepsilon(b-a), b - \varepsilon(b-a)]$. The ε-contraction $S^{-\varepsilon}$ of the cross S is $I^{-\varepsilon} \dagger J^{-\varepsilon}$.

THEOREM 1.2. *Let $\mathcal{S} = \{S_1, \ldots, S_n\}$ where each S_i is a cross. If $\cap_{i=1}^n S_i = \emptyset$ then there is a subset $\mathcal{S}'$ of $\mathcal{S}$ of size $\frac{4}{\varepsilon} + 4$ s.t. $\cap_{S \in \mathcal{S}'} S^{-\varepsilon} = \emptyset$.*

Theorem 1.2 is a key tool for our approximation algorithm for the case $k = 3$. More specifically, in the algorithm we design we will need to find the set $\mathcal{S}'$ that acts as a *witness* for the event $\cap_{i=1}^n S_i = \emptyset$ in an efficient manner. Accordingly we prove the following Theorem.

THEOREM 1.3. *The witness set $\mathcal{S}'$ in Theorem 1.2 can be found in time $O\left(n \log^2 n + \frac{\log^2 n}{\varepsilon}\right)$. Further, the set $\mathcal{S}'$ can be dynamically maintained in (amortized) time $O\left(\frac{\log^2 n}{\varepsilon}\right)$ per update, under insertion and deletion of crosses.*

Our contribution. To summarize, the contribution of this paper consists of: (i) An extension of the classical Helly theorem to a setting in which the Helly theorem does *not* hold, yet it holds with an ε-contraction of the objects. This is the first Helly theorem of this type. In the later work of [Langberg and Schulman 2007], a more general study of ε-contractions and the Helly number of sets of non-convex sets (and additional related problems) is addressed. (ii) The algorithmic application of this Helly result in a $\tilde{O}(nd)$-time 2-approximation k-clustering algorithm for lines in high-dimensional space, for the case $k = 2, 3$. For this algorithm the new Helly theorem plays a critical role in removing the 'curse of dimensionality'. This is a new technique with potentially more applications in high dimensional space.

Related work. Clustering and shape fitting problems on points have been actively studied in recent years. For the k-center problem on points in $\mathbb{R}^d$, efficient approximation algorithms with running time polynomially dependent on d are available. A simple greedy algorithm [González 1985; Hochbaum and Shmoys 1985; 1986] finds a 2-approximation and can be implemented in optimal time $O(nd \log k)$ [Feder and Greene 1988]. A recent result using coreset based techniques achieves a $(1 + \varepsilon)$-approximation algorithm with running time $k^{O(k/\varepsilon^2)} dn$ [Bǎdoiu et al. 2002]. These algorithms for points in $\mathbb{R}^d$ do not generalize to the case of incomplete data (i.e., lines). A major difficulty in such a generalization lies in the lack of a triangle inequality when considering lines. The problem is not that the triangle inequality is

slightly violated, but that *no* relaxation of it holds. No matter how far apart lines a and c are, there is always a line b that intersects both.

Nevertheless, in a previous work of ours [Gao et al. 2008], we addressed the 1-center problem (i.e., the case $k = 1$ in which we are to find a single ball intersecting all input lines). From a computational point of view, the 1-center problem significantly differs from the 2 and 3-center problems considered in this work. The 1-center problem is a convex optimization problem and therefore fundamentally easier than the cases of k-center for $k \geq 2$. What is in common is the statistical motivation and specifically the notion that the region of intersection of a line with a ball is a useful imputation of the missing data on that line.

In our prior work [Gao et al. 2008] we also required a certain modification of Helly's theorem. The two contributions are unrelated. In the prior work we had a situation in which Helly's theorem already applied (affine subspaces of $\mathbb{R}^d$) and needed a version in which the Helly number depended not on the ambient dimension d but on the dimension of the affine subspaces. In the current work we have a situation in which Helly's theorem does not apply and indeed would be false. However, we develop a version which is true when multiplicative contractions are considered.

There is an interesting literature on Helly-type theorems that involve the union of two or more convex sets (in our case we are considering the union of two axis parallel strips). For a nice survey on Helly type theorems in general see Eckhoff [Eckhoff 1993] or Wenger [Wenger 2004]. Let $\mathcal{C}_k^d$ be the family of all sets in $\mathbb{R}^d$ that are the union of at most k convex sets. The intersection of members in $\mathcal{C}_k^d$ are not necessarily in $\mathcal{C}_k^d$, and in general, it is not hard to verify that, subfamilies of $\mathcal{C}_k^d$ do not have finite Helly number (*i.e.,* there is not a finite witness for empty intersection). Nevertheless, it was shown independently by Matoušek [Matoušek 1997] and Alon and Kalai [Alon and Kalai 1995] that if $\mathcal{S}$ is a finite subfamily of $\mathcal{C}_k^d$ such that the intersection of every subfamily of $\mathcal{S}$ is in $\mathcal{C}_k^d$, then $\mathcal{S}$ has finite Helly number. Let $\mathcal{K}_k^d$ be the family of all sets in $\mathbb{R}^d$ that are the union of at most k *pairwise disjoint* convex sets. As before $\mathcal{K}_k^d$ does not have finite Helly number. Helly type theorems for subfamilies $\mathcal{S}$ of $\mathcal{K}_k^d$ such that the intersection of every subfamily of $\mathcal{S}$ is in $\mathcal{K}_k^d$ have been studied. Grunbaum and Motzkin [Grunbaum and Motzkin 1961] showed that for $k = 2$ the Helly number of such $\mathcal{S}$ is $2(d + 1)$, and for general k conjectured it to be $k(d + 1)$ (which is tight). The case $k = 3$ was proven by Larman [Larman 1968], and the general case by Morris [Morris 1973]. An elegant proof (based on the notion of LP-type problems) was presented by Amenta [Amenta 1996] and recently generalized to a topological setting in [Kalai and Meshulam 2008]. The results above do not apply for crosses. Further, our results on crosses do not depend on a restriction on the intersections of subfamilies of $\mathcal{S}$.

In a recent paper, [Demouth et al. 2008] study the following *approximate* Helly type theorem. Given a set system $\mathcal{F}$ of convex sets and a convex set U, approximately cover U by a small subset $\mathcal{F}'$ of $\mathcal{F}$. In [Demouth et al. 2008], a family of sets approximately covers a set U if it covers all but an ε fraction of its volume. Bounds on the size of $\mathcal{F}'$, which depend only on ε and the *fatness* of the sets in $\mathcal{F}$ are presented. More related to our problem, [Demouth et al. 2008] also consider the scenario in which both U and F consist of axis parallel squares in R^2. Although

the problems studied in [Demouth et al. 2008] differ from the ones studied in this work, one may find some resemblance between their proof techniques.

We would like to note that there has been work on "clustering points with k lines" [Agarwal and Procopiuc 2000; Agarwal et al. 2003; Har-Peled and Varadarajan 2002], where one finds a set of lines $\mathcal{L}$ such that the set of cylinders with radius r about these lines covers all the input points S. The problem we study in this paper can be phrased as "clustering lines with k points". There is no obvious duality between the problems, and they appear unrelated beyond their superficial similarity.

Comparison with coreset based techniques. In the study of approximate clustering, one of the powerful algorithmic techniques used is to devise a "coreset". For a set of data elements S, a coreset $S' \subseteq S$ is a small subset of representative points such that the optimization problems on S' is a good approximation to the optimal solution on S, *e.g.* [Agarwal et al. 2005]. Precisely, a subset S' is a $(1+\varepsilon)$-coreset of S if $(1+\varepsilon)\mu(S') \geq \mu(S)$, where μ is a monotonic measure function. Thus one can apply brute-forth algorithms on the small coreset S' and obtain efficient approximation algorithms for the optimization problems on S.

The proof techniques used in this work differ substantially from those based on coresets. It is thus natural to ask whether coreset based techniques apply to the clustering of incomplete data. In Theorem 1.1 we obtain a quasi-linear algorithm with approximation ratio $\simeq 2$ for the 2, 3 clustering of lines. In the Appendix B, we show that a weaker ratio of 58 (with a comparable running time) can be obtained using standard coreset based techniques. This algorithm works for any constant k. The design and analysis of the algorithm presented in Appendix B was motivated by the remarks of an anonymous referee. An efficient coreset based algorithm with an approximation ratio that matches that of Theorem 1.1 seems beyond reach.

Organization. The remainder of this work is organized as follows. In Section 2 we give a high level description of our 2- 3-center algorithms, and state the main Lemmas that need to be proven to obtain the running time specified in Theorems 1.1. In Section 3 we present our Helly type theorem for crosses (Theorem 1.2) and its dynamic implementation (Theorem 1.3). Our 3-center algorithm is strongly based on the results of this section. Finally, in Section 4 we prove the Lemmas specified in Section 2. We present the proofs of some of our claims in the Appendix. In addition, a constant approximation algorithm based on coresets for the k-center problem on lines, together with two independent $(1 + \varepsilon)$ approximation algorithms which run in time quasilinear in n but exponential in d and $\log 1/\varepsilon$ are presented in Sections B and C of the Appendix.

2.   PRELIMINARIES AND HIGH-LEVEL DESCRIPTION OF ALGORITHM

Given a set of n lines $\mathcal{L} = \{\ell_1, \cdots, \ell_n\}$ in $\mathbb{R}^d$, we consider the problem of finding k balls with minimum radius such that each line is covered by at least one ball. The optimal k-clustering radius of $\mathcal{L}$ is denoted by $r_k(\mathcal{L})$, and is referred to as the *intersecting radius* (in this work we assume all balls have the same radius). The centers are defined as the *intersecting k-centers* (or just centers for short). We denote by $B_r(c)$ a ball centered at a center c with radius r in $\mathbb{R}^d$. We denote by $d(.,.)$ the Euclidean distance function. The distance between two points p, q is also written as $|pq|$. The distance from a point p to a set Q, $d(p, Q)$, is defined as the minimum distance from p to any point in Q.

Decision vs. general problem. Most of the following paper discusses a variant of the k-center problem, that has the flavor of a decision problem: Namely, given a set of lines $\mathcal{L}$ and a radius r, find a k-clustering of $\mathcal{L}$ of radius at most r or report that $r < r_k(\mathcal{L})$. We refer to this problem as the *decision k-center* problem. Given a set of lines $\mathcal{L}$ and a radius r, an α-approximation to the decision k-center problem will find a k-clustering of $\mathcal{L}$ of radius at most αr or report that $r < r_k(\mathcal{L})$.

Any approximation algorithm for decision k-center is also an approximation algorithm for k-center once $r_k(\mathcal{L})$ is given as part of the input. Thus, to turn our algorithm into one for the k-center problem we need to compute the value of $r_k(\mathcal{L})$; or a good estimate to $r_k(\mathcal{L})$. We use the results of a coreset based algorithm (presented in Section B) to efficiently obtain a constant approximation r to the optimal radius $r_k(\mathcal{L})$ for the k-center problem on $\mathcal{L}$. Now, using a standard binary search, we can run our algorithms for decision k-center at most $\log 1/\varepsilon$ times (to obtain a $1 + \varepsilon$ approximation to the radius). The running time of the algorithm will have a multiplicative factor of $O(\log 1/\varepsilon)$ of the running time of the decision version k-center. The algorithm of Section B is randomized, and it is the only source of randomness in our overall algorithm for k-clustering.

Problem structure. For a set of lines $\mathcal{L}$, assume that there are k centers c_i, $i = 1, \ldots, k$, with intersecting radius r. Also assume that there are k lines, say $\ell_1, \ldots, \ell_k$ with the property that they belong to the clusters corresponding to the centers $c_1, \ldots, c_k$. Namely, the k lines satisfy $d(\ell_i, c_i) \leq r$. We call these lines *representative lines*. Consider the projection $\hat{c}_i$ of c_i onto ℓ_i. By the triangle inequality, $\hat{c}_i$'s are the centers of a k-cluster for $\mathcal{L}$ of radius $2r$. In this paper we actually look for centers that *stay* on the lines of $\mathcal{L}$ and focus on 2 or $2 + \varepsilon$-approximation to the decision k-center problem. The reason for this is that the centers on $\mathcal{L}$ have some special structures we can exploit.

Consider a representative line ℓ_i, $1 \leq i \leq k$. We define a set of intervals on ℓ_i, one interval per line $\ell \in \mathcal{L}$. For each line ℓ_j in $\mathcal{L}$ we define the interval I_j^i on ℓ_i, where I_j^i consists of points on ℓ_i that are *close* to ℓ_j, defined to be within distance $2r$ from ℓ_j. Namely $I_j^i = \{p \in \ell_i | d(p, \ell_j) \leq 2r\}$. We call the interval I_j^i the $2r$ *proximity interval* of ℓ_j on ℓ_i. Notice that any point p_i on ℓ_i for which $p_i \in I_j^i$ covers ℓ_j with radius at most $2r$. We denote by $\mathcal{L}(p_i)$ the set of lines covered by p_i with radius $2r$. For each line $\ell_j \in \mathcal{L}(p_i)$ it holds that $p_i \in I_j^i$. There are at most n proximity intervals on each line ℓ_i. Taking the arrangement of these intervals, we get at most $2n$ different regions, called *canonical regions*, such that each region is an interval and any point p_i in a region covers the same set of lines with radius $2r$. Thus we are looking for a set of k centers p_i, $1 \leq i \leq k$, with $p_i \in \ell_i$ such that collectively they cover all the lines in $\mathcal{L}$. There are only a finite number of choices for p_i as we only need to choose one from each canonical region. The challenge is that we do not know the representative lines ℓ_i and we need to find the centers p_i efficiently.

We start by describing a naive 2-approximation algorithm for the decision k-center problem. For $\mathcal{L}$ of size n, the naive 2-approximation algorithm we present for decision k-center will run in time proportional to $n^{2k}d$. The $(2 + \varepsilon)$-approximation algorithms we describe for $k = 2$ and $k = 3$ (for the k-center problem) achieve time $\tilde{O}(nd)$ and are considerably more involved, however it is easiest to explain them after the description of the naive algorithm.

2.1 The naive algorithm for the decision k-center problem

Consider a k-clustering of $\mathcal{L}$ and a given radius r, we want to check whether we can find k centers such that any line is within distance $2r$ from them. The naive algorithm we present has two steps. In the first step we find the k representative lines, $\ell_1, \ldots, \ell_k$ that belong to different clusters. In fact, we exhaustively search each and every subset of lines of size k. For each choice of a subset of k lines, we run the second step to look for the cluster centers on these lines ℓ_i. The second step is guaranteed to succeed given a valid set of representative lines. If the second step fails, then we go back to step 1 to try another set of representative lines.

In the second step of the algorithm, we have a set of representative lines $\ell_1, \ldots, \ell_k$. Again, denote by c_i the optimal center and $\hat{c}_i$ the projection of c_i on ℓ_i. We now describe an inductive procedure which finds the points $p_i \in \ell_i$ that cluster $\mathcal{L}$ with radius $2r$. For the first point p_1, we exhaustively test p_1 in every canonical region. One of the canonical regions contain the projection $\hat{c}_1$ of the optimal center c_1. When p is in the same canonical region as $\hat{c}_1$, the remaining lines $\mathcal{L} \setminus \mathcal{L}(p_1)$ can be $(k-1)$-clustered with radius $2r$ by the points $\hat{c}_2, \ldots, \hat{c}_k$ that lie on $\ell_2, \ldots, \ell_k$ respectively. So, once p_1 is found, we can recurse and find p_i for $i = 2, \ldots, k$.

The running time of the algorithm is roughly n^{2k}, and it results with a 2-approximation for the k-center problem. This can easily be improved to run in time $\simeq n^k$, by deleting Step 1 above and performing the second step on sets of lines $\ell_1, \ldots, \ell_k$ that are not fixed in advance but rather vary according to the points p_i picked in the iterations. Namely, the points $p_1, \ldots, p_i$ under consideration will define a set of lines $\mathcal{L}_i$ covered by $p_1, \ldots, p_i$, and one can define ℓ_{i+1} to be any line not in $\mathcal{L}_i$. However, such an algorithm slightly deviates in structure from the efficient algorithm we describe in the upcoming section.

2.2 The 2-center problem

The naive algorithm described above is expensive computationally since we use exhaustive search in both of the steps. There are a few opportunities that we can speed up the algorithm. We explain an efficient algorithm for the simple case of the 2-center problem, as an appetizer for the more involved case of $k = 3$ described later. Throughout this section we will assume the given set of lines $\mathcal{L}$ does not have a 1-clustering of radius $2r$ with a center on a line in $\mathcal{L}$ (otherwise such a clustering can be found in time $O(nd)$ as in [Gao et al. 2008]).

Our algorithm follows the same outline as the naive algorithm and has two steps. In the first step, we look for 2 representative lines ℓ_1, ℓ_2, each from a different cluster. In the second step, we find centers p_i on ℓ_i to cluster $\mathcal{L}$ with radius $2r$.

LEMMA 2.1 (STEP 1). *Let $\mathcal{L}$ be a set of n lines in $\mathbb{R}^d$. Let c_1, c_2 be the centers and r the radius of a 2-clustering for $\mathcal{L}$. If $\mathcal{L}$ does not have a 1-clustering of radius $2r$ with center on a line in $\mathcal{L}$, then one can find in time $O(nd)$ two pairs of lines (ℓ_1^1, ℓ_2^1), and (ℓ_1^2, ℓ_2^2) s.t. either $d(\ell_j^1, c_j) \leq r$ for $j = 1, 2$ or $d(\ell_j^2, c_j) \leq r$ for $j = 1, 2$.*

PROOF. Let ℓ_1 be an arbitrary line. In the r radius 2-clustering, ℓ_1 is covered by a center point, say c_1. $d(c_1, \ell_1) \leq r$. Set $\ell_1^1 = \ell_1^2 = \ell_1$. For $\ell_j \neq \ell_1$ let $I_j = \{p \in \ell_1 | d(p, \ell_j) \leq 2r\}$. By our assumption the intervals I_j have empty intersection (otherwise $\mathcal{L}$ would have a 1-clustering of radius $2r$ with center on ℓ_1). Thus, by Helly's theorem, there exists a pair of intervals $I_j, I_{j'}$ with empty intersection. Let ℓ_j and $\ell_{j'}$ be their corresponding lines and set $\ell_2^1 = \ell_j$ and $\ell_2^2 = \ell_{j'}$.

It follows that either $d(c_1, \ell_2^1) > r$ or $d(c_1, \ell_2^2) > r$ (otherwise I_j and $I_{j'}$ would intersect). Thus either $d(c_2, \ell_2^1) \leq r$ or $d(c_2, \ell_2^2) \leq r$. One of the pairs, $\{\ell_1^1, \ell_2^1\}$, $\{\ell_1^2, \ell_2^2\}$ qualifies as a pair of representative lines. □

We remark that we make use of the Helly theorem in the contrapositive – if a set of convex sets in $\mathbb{R}^d$ does not have a common intersection, there must be a small *witness* set of size $d+1$ that does not have a common intersection. With the above Lemma we get a pair of candidate representative lines in linear time, at least one of them is a valid set of representative lines. In particular, the proximity intervals I_j can be build in time $O(nd)$. Let $I_j = [p_j, q_j]$. We consider the case in which the p_j's and q_j's are finite (a similar analysis can be given otherwise). The intervals I_j intersect if and only if $\min_j q_j \geq \max_j p_j$. Thus, by identifying the interval I_j with minimum q_j and that with maximum p_j we complete the implementation of Step 1 in linear time.

For Step 2, assume we have representative lines ℓ_1 and ℓ_2. We will look for two points $p_1 \in \ell_1$ and $p_2 \in \ell_2$ that altogether cover $\mathcal{L}$ with radius $2r$. For that, we consider points p_1 in each canonical region defined by the arrangement of proximity intervals I_j^1 of ℓ_j on ℓ_1. The point p_1 covers the set of lines in $\mathcal{L}(p_1)$ with radius $2r$. Then we ask the question whether the set of lines $\bar{\mathcal{L}}(p_1) = \mathcal{L} \setminus \mathcal{L}(p_1)$, *not* covered by p_1 with radius $2r$, can be covered by a point p_2 on ℓ_2 with radius $2r$. An immediate observation is that the subsets $\bar{\mathcal{L}}(p_1)$ corresponding to points p_1 in *adjacent* canonical regions differ in at most a single line, thus one may benefit by checking whether $\bar{\mathcal{L}}(p_1)$ has a good 1-clustering in a **dynamic** manner. In the next Lemma, we design a dynamic data structure which will update and answer the desired queries in time $O(d + \log n)$.

LEMMA 2.2 (STEP 2). *Let ℓ_2 be a line and r a given radius. There exists a dynamic data structure on a set of lines $\mathcal{L}$ of size at most n which answers queries of the form "Is there a 1-cluster of $\mathcal{L}$ with center on ℓ_2 of radius at most r" with the following properties: (i) Insert line: time $O(d + \log n)$. (ii) Delete line: time $O(\log n)$. (iii) Constructive query: if answer is positive returns appropriate center on ℓ_2, query time $O(\log n)$.*

PROOF. Assume w.l.o.g. that ℓ_2 lies on a main axis in $\mathbb{R}^d$. Namely, we can associate with each point on ℓ_2 a number p in $\mathbb{R}^1$. Let ℓ_j be a line. Let $I_j^2 = \{p \in \ell_2 | d(p, \ell_j) \leq r\} = [p_j, q_j] \subset \mathbb{R}^1$. As before, we consider the case in which the p_j's and q_j's are finite (a similar analysis can be given otherwise). We maintain a dynamic data structure for the intersection of I_j^2 over $\ell_j \in \mathcal{L}$ which returns a point in the intersection of the intervals in case they intersect. This suffices as $\mathcal{L}$ has a 1-cluster with center on ℓ_2 of radius at most r if and only if the intervals I_j^2 for $\ell_j \in \mathcal{L}$ intersect. The intervals I_j^2 intersect if and only if $\min_j q_j \geq \max_j p_j$. Thus it suffices to maintain the values $\min_j q_j$ and $\max_j p_j$ dynamically. This can be achieved by two heaps. □

With Step 1 and Step 2 together, we have an algorithm for the decision 2-center problem of n lines. Namely, for the representative lines obtained in Step 1, we *slide* a candidate center p_1 on ℓ_1. For each canonical region that p_1 visits, we check whether there is a 1-center of radius $2r$ with the center p_2 on the line ℓ_2 to cover the rest of the lines. This is done using the dynamic algorithm explained in Step 2. The total running time is $O(nd + n \log n)$.

2.3 Overview for the 3-center problem

The 3-center problem is significantly more complex than the 2-center problem. Our solution involves a new Helly theorem for crosses (presented in Section 3). In the following, we first establish the connection of the 3-center problem with our new Helly theorem on crosses.

For the approximate decision 3-center problem on a set of lines $\mathcal{L}$ and a given radius r, we look for centers on the lines with intersecting radius $(2 + \varepsilon)r$. The general proof paradigm follows that of the 2-center case. First we assume that the lines $\mathcal{L}$ can not be clustered by 2 centers with radius $(2 + \varepsilon)r$. Otherwise we can run the 2-center algorithm in Section 2.2. We start with Step 1 in which we find a set of representative line triplets $\{(\ell_1^i, \ell_2^i, \ell_3^i)\}$ with the property that for one such triplet it holds that the corresponding lines belong to different clusters in some r radius 3 clustering of $\mathcal{L}$. We start with the same algorithm used in the first step for the 2-center problem and identify two pairs of lines $(\ell_1, \ell_2), (\ell_1', \ell_2')$ with at least one pair belonging to different clusters. In the following assume that ℓ_1, ℓ_2 belong to different clusters. Almost as before, for every other line $\ell_j \notin \{\ell_1, \ell_2\}$, define the proximity interval of radius $(2 + \varepsilon)r$ on them, denoted as I_j^1 and I_j^2 respectively. This is the point in which "crosses" come in.

Consider the set of crosses $\mathcal{S} = \{S_j\}$ defined by $S_j = I_j^1 \dagger I_j^2$. When defining S_j we view the intervals I_j^1 and I_j^2 as subsets of $\mathbb{R}^1$ via some mapping of ℓ_1 and ℓ_2 to $\mathbb{R}^1$. Thus, in this rough overview, it is easiest to imagine that ℓ_1 coincides with the x-axis and that ℓ_2 coincides with the y-axis. Let p_1 be a point on ℓ_1, p_2 be a point on ℓ_2, and let $\mathbf{p} = (p_1, p_2)$. Our main observation in this modeling is that when $\mathbf{p}$ stabs a cross S_j, it holds that the line ℓ_j is covered by either p_1 or p_2 (or both) with radius $(2+\varepsilon)r$. Now, since $\mathcal{L}$ cannot be covered by any two centers p_1, p_2 with radius $(2 + \varepsilon)r$, this implies that the $n - 2$ crosses $\{S_j\}$ cannot be stabbed by any point $\mathbf{p} = (p_1, p_2)$ with $p_i \in \ell_i$. Equivalently, the crosses do not have a common intersection. Now, recall our Helly theorem for crosses (Theorem 1.2). If the crosses $\mathcal{S}$ do not have common intersection, there is a small witness set $\mathcal{S}'$ of size $O(1/\varepsilon)$ such that the contracted crosses do not have a common intersection. Each such contracted cross corresponds to a certain line ℓ_j. With a few computations, we can show (in Section 4) that one such line ℓ_j cannot share a cluster with neither ℓ_1 or ℓ_2. This implies a *small* set of triplets as desired: $\{(\ell_1, \ell_2, \ell_j)\}_{S_j \in \mathcal{S}'}$.

For Step 2, assume we have already obtained the representative lines (ℓ_1, ℓ_2, ℓ_3). As before, we consider a point p_1 in each of the canonical regions of ℓ_1, and would like to ask whether the remaining lines (not covered by p_1) denoted by $\bar{\mathcal{L}}(p_1)$ can be covered by two other centers $p_2 \in \ell_2$, $p_3 \in \ell_3$. Here we use the concept of crosses once more. Namely, we identify with each line ℓ_j, its proximity intervals I_j^2 on ℓ_2 and I_j^2 on ℓ_3, and the corresponding cross $S_j = I_j^2 \dagger I_j^3$. It now holds that the lines in $\bar{\mathcal{L}}(p_1)$ can be covered by a center p_2 on ℓ_2 and p_3 on ℓ_3 iff their corresponding crosses have a common intersection point. So the question is, can one efficiently find such a point (that stabs all these crosses) if one exists or declare that none such points exist? As before, when p_1 moves to an adjacent canonical region on ℓ_1, there is only a slight change of a single line in $\bar{\mathcal{L}}(p_1)$. Thus at most 1 cross is added or removed from the corresponding cross set for each shift of p_1. In our proofs, we show using our Helly theorem for crosses, that the question specified above can be answered very efficiently, even in a dynamic scenario in which the cross set over

goes local changes.

In the next section we present the proofs for the Helly theorem on crosses. Then we will explain the details of the 3-center algorithm. In this work we do not address the k-center problem for $k \geq 4$. We have noticed that the case of $k \geq 4$ can be dealt with in a similar manner given theorems analogous to Theorems 1.2 and 1.3 which address crosses in higher dimensions (not only in $\mathbb{R}^2$). However, this currently remains out of reach.

3. HELLY THEOREM FOR CROSSES

In this section we prove Theorems 1.2 and 1.3. Recall that the ε multiplicative contraction of an interval $I = [a, b]$ is $I^{-\varepsilon} = [a + \varepsilon(b - a), b - \varepsilon(b - a)]$, for $\varepsilon > 0$. For $\varepsilon > 0$, we have $I^{\varepsilon} = [a - \varepsilon(b - a), b + \varepsilon(b - a)]$ which is actually multiplicative expansion. We use superscript ε to denote the expansion (contraction) of an interval, a cross, or a set of crosses. It is useful to note that for any interval I it holds that $(I^{-\varepsilon})^{\frac{\varepsilon}{1-2\varepsilon}} = (I^{\varepsilon})^{-\frac{\varepsilon}{1+2\varepsilon}} = I$. Here, and throughout, we assume $\varepsilon > 0$.

PROOF. **(of Theorem 1.2)** To simplify notation we will prove the following theorem which implies Theorem 1.2: *Let $\mathcal{S} = \{S_1, \ldots, S_n\}$ where each S_i is a cross. If $\cap_{i=1}^{n} S_i^{\varepsilon} = \emptyset$ then there is a subset $\mathcal{S}'$ of $\mathcal{S}$ of size $\frac{4}{\varepsilon} + 4$ s.t. $\cap_{S \in \mathcal{S}'} S = \emptyset$.*

We will iteratively construct a *small* subset $\mathcal{S}'$ of $\mathcal{S}$ with empty intersection. Initially, let $\mathcal{S}'$ be the empty set. With each intermediate set $\mathcal{S}'$ we associate a set $\mathcal{B}$ (for *bad*) which is the intersection of the elements of $\mathcal{S}'$. Initially $\mathcal{B} = \mathbb{R}^2$ and we would like $\mathcal{B}$ to be $\emptyset$.

We start by giving some notation (depicted in Figure 1 (i) and (ii)). Let $S_i = I_i \dagger J_i$ where $I_i = [x_i, y_i]$ and $J_i = [u_i, v_i]$. The cross S_i defines four regions: $A_i = \{(p, q) \mid p < x_i \text{ and } p > v_i\}$, $B_i = \{(p, q) \mid p > y_i \text{ and } p > v_i\}$, $C_i = \{(p, q) \mid p > y_i \text{ and } p < u_i\}$, and $D_i = \{(p, q) \mid p < x_i \text{ and } p < u_i\}$. Similar definitions are given for the sets S_i^{ε}. In such cases the defining parameters will be enhanced with the symbol $()^{\varepsilon}$. Let A be the union of the sets A_i in $\mathcal{S}$, B the union of the sets B_i in $\mathcal{S}$, and C and D defined with respect to C_i and D_i. Let a, b, c, and d be the boundary of A, B, C, D respectively. Let ab be the intersection point of a and b. Similarly, let cd be the intersection of c and d. As $\cap_{i=1}^{n} S_i = \emptyset$, in both cases since the curves are x and y-monotone the intersection exists and is at most a horizontal or vertical line. In what follows we assume that the intersection is a single point, similar proof can be given otherwise.

We now start to construct $\mathcal{S}'$. We start by finding two crosses S_i and S_j such that $A_i \cup B_j$ covers the half plane above the point ab. Assume that the point ab is on a vertical portion of the boundary a and on a horizontal portion of the boundary b (other cases can be dealt with in a very similar manner). We take S_i to be the cross defining the boundary of A at the point ab, and S_j to be the cross defining the boundary of B at the point ab. It is not hard to verify that S_i and S_j have the properties desired.

In a similar manner we find two additional crosses $S_{i'}$ and $S_{j'}$ such that $C_{i'} \cup D_{j'}$ cover the half space below the point cd. Let $\mathcal{S}' = \{S_i, S_j, S_{i'}, S_{j'}\}$. If the point ab is below the point cd then we are done (namely the set $\mathcal{B}$ corresponding to $\mathcal{S}'$ is empty). Otherwise assume ab is to the right and above of cd (other cases can be dealt with in a similar manner). Notice that all points in $\mathcal{B}$ are in the horizontal strip defined by the points between ab and cd. We will now start adding sets to $\mathcal{S}'$ as to shrink the size of $\mathcal{B}$.
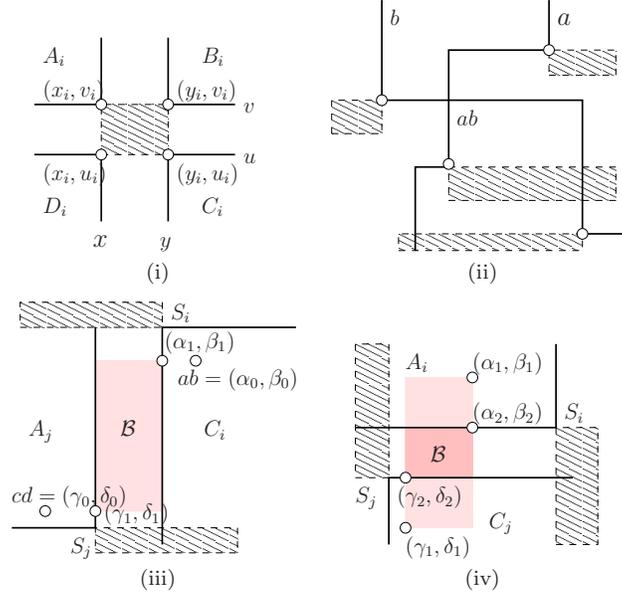
Fig. 1. (i) A cross defined by the intervals $[x_i, y_i]$ and $[u_i, v_i]$, together with the regions A_i, B_i, C_i and D_i. (ii) The boundaries a and b and the intersection point ab. The dotted rectangles correspond to the *center* of each cross. (iii) Find the cross S_i^ε such that ab is in C_i^ε and y_i^ε is minimal; Find the cross S_j^ε such that cd is in A_j^ε and x_j^ε is maximal. (iv) Find two crosses that shrink $\mathcal{B}$ along the vertical direction substantially.

Consider the point $ab = (\alpha_0, \beta_0)$. It is not in the intersection of the sets in $\mathcal{S}^\varepsilon$ (as they have empty intersection). Thus there is at least one cross S_i^ε such that ab lies in one of the four regions: A_i^ε, B_i^ε, C_i^ε, or D_i^ε. It is not hard to verify that ab can only be in C_i^ε. For example assume ab is in D_i^ε. Since cd is to the left and below ab, the region D_i^ε includes in its interior the point cd which contradicts the fact that cd is on the boundary of D. A similar argument can be given for A_i^ε and B_i^ε.

Let S_i^ε be the cross for which ab is in C_i^ε and y_i^ε is minimal. Similarly let S_j^ε be the cross for which $cd = (\gamma_0, \delta_0)$ is in A_j^ε and x_j^ε is maximal. Let $(\alpha_1, \beta_1) = (y_i, \beta_0)$ and $(\gamma_1, \delta_1) = (x_j, \delta_0)$. Now by adding S_i and S_j to $\mathcal{S}'$ we study the corresponding set $\mathcal{B}$. We consider two cases: if $\alpha_1 < \gamma_1$ then $\mathcal{B} = \emptyset$ and we are done. Otherwise, it can now be seen that $\mathcal{B}$ is included in the rectangle defined by the points (α_1, β_1) (the upper right corner) and (γ_1, δ_1) (the lower bottom corner). See Figure 1 (iii).

We now turn to proceed in several iterations in which the size of the rectangle including $\mathcal{B}$ gets smaller and smaller. In each iteration we add two sets to $\mathcal{S}'$ for so called *vertical progress* and two sets for *horizontal progress*. In general, the analysis of each iteration is similar to that described above. We will now define a single iteration giving vertical progress. Iterations giving horizontal progress are analogous.

Consider the point (α_1, β_1). It is not in the intersection of the sets in $\mathcal{S}^\varepsilon$ (as they have empty intersection). Thus there is at least one cross S_i^ε such that (α_1, β_1) is

in one of the four regions A_i^ε, B_i^ε, C_i^ε, or D_i^ε. It is not hard to verify that (α_1, β_1) can only be in A_i^ε. If (α_1, β_1) is in C_i^ε, then S_i^ε should have been chosen in previous steps since its y value is smaller (recall that we chose the cross defining (α_1, β_1) to have minimal y value). If (α_1, β_1) is in B_i^ε, B_i^ε includes in its interior the point ab which contradicts the fact that ab is on the boundary of B. A similar argument can be given for the last case.

Now let S_i be the cross for which (α_1, β_1) is in A_i^ε and v_i^ε is minimal. Similarly we find a set S_j for which (γ_1, δ_1) is in C_j^ε and u_j^ε is maximal. Let $(\alpha_2, \beta_2) = (\alpha_1, v_i)$ and $(\gamma_2, \delta_2) = (\gamma_1, u_j)$. It is now not hard to verify that by adding S_i and S_j to $\mathcal{S}'$ the corresponding set $\mathcal{B}$ is either empty and we are done (if $\delta_2 > \beta_2$) or included in the rectangle defined by the pair of points (α_2, β_2) and (γ_2, δ_2). The **crucial thing to notice** is that the *size* of the rectangle $\mathcal{B}$ has shrunken significantly. Namely v_i is significantly smaller than β_1 and u_j is significantly larger than δ_1. To show that v_i is significantly smaller than β_1 notice that our definitions imply that $u_i^\varepsilon \le u_i \le \delta_0$ (otherwise the set D_i corresponding to S_i would include the point cd in its interior). Now consider two cases: if v_i^ε is significantly smaller than β_1 then so is v_i, otherwise $v_i^\varepsilon - u_i^\varepsilon$ is *large* with respect to $\beta_0 - \delta_0$ and we can show that $v_i^\varepsilon - v_i$ (the gap obtained by shrinking the cross) is also large with respect to $\beta_0 - \delta_0$. See Figure 1 (iv). Basic calculations show that we are guaranteed that $\beta_1 - v_i = \beta_1 - \beta_2$ is at least $\frac{\varepsilon}{4}(\beta_0 - \delta_0)$. Similarly we have that $u_j - \delta_1 = \delta_2 - \delta_1$ is at least $\frac{\varepsilon}{4}(\beta_0 - \delta_0)$. All in all, we obtain $(1 - \frac{\varepsilon}{2})(\beta_0 - \delta_0) > \beta_2 - \delta_2$. Thus, in this iteration we have made progress of $\frac{\varepsilon}{2}$ in our goal to shrink the size of $\mathcal{B}$. The progress was made is in the *vertical* axis of the rectangle $\mathcal{B}$. One can analogously now make $\frac{\varepsilon}{2}$ progress in the horizontal axis of $\mathcal{B}$ also. Namely one can add two sets to $\mathcal{S}'$ and obtain a corresponding set $\mathcal{B}$ contained in a rectangle defined by the points (α_3, β_3) and (γ_3, δ_3) where $(1 - \frac{\varepsilon}{2})(\alpha_0 - \gamma_0) > \alpha_3 - \gamma_3$.

We continue such iterations in a similar manner, in which each iteration will include both horizontal and vertical progress. To be precise, we make vertical progress with respect to the point (α_i, β_i) only if $\beta_0 - \beta_i \le (\beta_0 - \delta_0)/2$, and vertical progress with respect to the point (γ_i, δ_i) only if $\delta_i - \delta_0 \le (\beta_0 - \delta_0)/2$. A similar condition holds for horizontal progress w.r.t. the points (α_i, β_i) and (γ_i, δ_i). The progress made will always be of magnitude $\frac{\varepsilon}{2}$ times $(\beta_0 - \delta_0)$ for vertical phases and $(\alpha_0 - \gamma_0)$ for horizontal phases. For example, in the next iteration, in which we define the boundaries (α_4, β_4) and (γ_4, δ_4) of $\mathcal{B}$, it will hold that $(1 - 2 \cdot \frac{\varepsilon}{2})(\beta_0 - \delta_0) > \beta_4 - \delta_4$. This follows from the fact that the cross S_i added to $\mathcal{S}'$ corresponding to (α_3, β_3) has $u_i \le \delta_0$ and the cross S_j added to $\mathcal{S}'$ corresponding to (γ_3, δ_3) has $v_j \ge \beta_0$.

All in all, with m iterations we obtain a subset $\mathcal{S}'$ with a corresponding set $\mathcal{B}$ which is included in a rectangle defined by the points (α, β) and (γ, δ) for $(1 - \frac{m\varepsilon}{2})(\beta_0 - \delta_0) > \beta - \delta$ and $(1 - \frac{m\varepsilon}{2})(\alpha_0 - \gamma_0) > \alpha - \gamma$. Hence taking m to be $\frac{2}{\varepsilon}$ will yield a subset $\mathcal{S}'$ of size at most $\frac{4}{\varepsilon} + 4$ as desired. $\square$

COROLLARY 3.1. *Let $\mathcal{S} = \{S_1, \ldots, S_n\}$ where each S_i is a cross. If $\cap_{i=1}^n S_i \neq \emptyset$, then running the algorithm specified in the proof of Theorem 1.2 will result in a point $p \in \cap_{i=1}^n S_i^\varepsilon$.*

PROOF. We use the same notation as in the proof of Theorem 1.2. Given $\mathcal{S}$, the algorithm of Theorem 1.2 must fail to produce $\mathcal{S}'$ such that $\cap_{s \in \mathcal{S}'} S = \emptyset$. This will happen only if (i) The boundaries a or b do not intersect. (ii) The boundaries c and d do not intersect. (iii) For some iteration, one of the points (α_i, β_i) or (γ_i, δ_i)

is in $\cap_{i=1}^n S_i^\varepsilon$. In all cases above a point in $\cap_{i=1}^n S_i^\varepsilon$ is implied.   □

We now prove Theorem 1.3 stated in the Introduction. Theorem 1.3 addresses an efficient implementation of the algorithm described in Theorem 1.2.

PROOF. **(of Theorem 1.3)** We use the same notation as in Theorem 1.2. Since Theorem 1.2 is constructive, we will explain here the data structures we use as well as how they support the operations. To prove the 3-clustering result in Theorem 1.1, we will be interested in dynamic data structures that support insertion and deletion of crosses (which will correspond to lines in the clustering instance), thus such data structures are presented and analyzed.

The algorithm presented in Theorem 1.2 can be described in two phases. In the first phase we find the points ab and cd. In the second phase we iterate over points (α_i, β_i) and (γ_i, δ_i): for each such point we find a cross that does not include the point. The crosses we find satisfy certain *minimality* properties. In what follows we describe a dynamic data structure that allows to maintain the contours a, b, c, and d and the intersection points ab and cd. For the second phase of the algorithm we use the dynamic data structure specified in [Agarwal et al. 2005] which allows stabbing queries for axis-parallel rectangles with the maximum/minimum value objective function. The amortized update and query time of the dynamic structure in [Agarwal et al. 2005] is $O(\log^2 n)$.

We now explain the data structure for maintaining the boundary a of $A = \cup_{i=1}^n A_i$ for $\mathcal{A} = \{A_1, \ldots, A_n\}$. The remaining boundaries are maintained similarly. We first show how to build the data structure for $\mathcal{A}$, then we elaborate on deletion and insertion of some A_i. We define $p_i = (x_i, v_i)$ as the corner point of a region A_i. The boundary of A_i is denoted as a_i. We sort all the points p_i, for all i, along the 45° line (that is, we project these points onto the line "$x = v$" and sort the corresponding projected points, here x and v are the main axes). We build a balanced binary search tree T with the points p_i as leaves. For each internal node u in T, denote by $N(u)$ the set of nodes in its subtree (sorted along the 45° line). For each node u in the tree, we implicitly maintain the boundary $a(u)$ of $\cup_{i \in N(u)} A_i$. The observation is that $a(u)$ can be simply computed from the boundaries of the children w and v of u, i.e., $a(w), a(v)$. Throughout, we will assume the value of w is smaller than that of v with respect to the 45° line, for example see Figure 2 (i). In this example, $a(u)$ is the concatenation of the first part of $a(w)$ (up to the intersection point) and the second part of $a(v)$ (from the intersection point).

Storing the contour $a(u)$ entirely at each node of our tree will involve too many changes when considering dynamic updates. Thus at each node u we only store the intersection of the contours of its children $a(w)$ and $a(v)$. If such a point does not exist, we store one of two symbols specifying which of the contours of the two children is lowest. For leaves we will store the corresponding point p_i. When computing the intersection point of $a(w)$ and $a(v)$ we use the recursively computed information stored in the subtrees rooted at w and v.

We now show how to find the intersection of the contours of $a(w)$ and $a(v)$. The intersection of the contours $a(w)$ and $a(v)$ is either the intersection of $a(w)$ with the lowest horizontal portion of $a(v)$ or the intersection of $a(v)$ with the rightmost vertical portion of $a(w)$. Accordingly, the basic operation we use finds the intersection of a horizontal (alternatively vertical) line with the boundary of $a(w)$ (alternatively $a(v)$). Let ℓ be a horizontal line. One can use the subtree structure of w to search for the intersection of ℓ and $a(w)$. Namely, we compare the horizontal line ℓ with
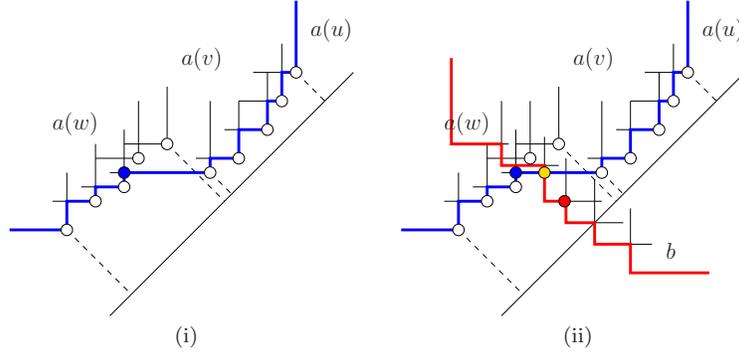
Fig. 2. (i) The data structure that maintains the boundary of A. A node u stores the intersection (drawn in blue) of the boundaries of its children. (ii) To search for the intersection of a and b, we use the binary tree structures for a and b. We compare the intersections stored at the root of each tree, based on which we recurse on the corresponding subtrees.

the intersection stored at the node w. If ℓ is above the intersection, we recurve on the right subtree of w, otherwise we recurse on the left subtree. This binary search costs at most $O(\log |N(w)|)$, where $|N(w)|$ is the number of nodes in the subtree of w. In the case described above ℓ is the lowest line of the contour $a(v)$.

To summarize, our data structure on n crosses can be constructed in time $O(n \log n)$. It also allows dynamic updates in time $O(\log^2 n)$ under insertion and deletion. Upon the insertion of a new region A_0, we first insert the corner p_0 in the sorted list. Then we propagate the boundary update from the leaf to the root. Only the nodes on the path from A_0 to the root of the tree will be updated. The update at each internal node may also trigger another binary search which costs $O(\log n)$. Standard rotation mechanisms may be used here to keep the tree balanced. The total update cost is $O(\log^2 n)$ at most. Deletion can be handled in the same way. Notice that at internal nodes we do not explicitly keep the boundary of the subtree, but instead only an intersection of the boundaries in subtrees. Although the deletion of a region A_0 may suddenly increase the complexity of a (e.g., a portion of the boundaries that was hidden by A_0 now becomes suddenly exposed), the update cost is low.

It is left to show given a data structure for a and one for b how one can find the intersection point ab. See Figure 2 (ii). This can be achieved by traversing down the search trees for a and b. Basically we compare the intersections stored at the root for a and b, based on which we are able to eliminate half of the boundary of a or b and recurse. The cost is $O(\log n)$.

We conclude that the algorithm in Theorem 1.2 can be implemented in time $O(n \log^2 n + \log^2 n/\varepsilon)$. Computing ab and cd can be done in time $O(n \log n)$ and each iteration of the algorithm can be done using the stabbing query data structure mentioned above in time $O(\log^2 n)$ after a preprocessing of time $O(n \log^2 n)$. As all the data structures we considered support insertion and deletion of crosses, the witness $\mathcal{S}'$ can be maintained in (amortized) time $O(\log^2 n/\varepsilon)$ under such updates. $\square$

We now state two lemmas. The first shows that Theorem 1.2 does not hold without the notion of contraction. The second shows that our analysis in Theorem 1.2

is tight (up to constant factors). Both lemmas are proven in the Appendix.

LEMMA 3.1. *There exists a set $\mathcal{S} = \{S_1, \ldots, S_n\}$ of crosses such that $\cap_{i=1}^n S_i = \emptyset$ but every subset $\mathcal{S}'$ of $\mathcal{S}$ of size $n - 1$ has non empty intersection.*

LEMMA 3.2. *For any $\varepsilon > 0$, there exists a set $\mathcal{S} = \{S_1, \ldots, S_n\}$ of crosses such that $\cap_{i=1}^n S_i^\varepsilon = \emptyset$ but every subset $\mathcal{S}'$ of $\mathcal{S}$ of size $1/(4\varepsilon) - 2$ has non empty intersection.*

4.   DECISION 3-CENTER PROBLEM

In this section we will describe the details of the algorithm that computes a $(2+\varepsilon)$-approximate 3-clustering of a set of lines $\mathcal{L}$ in quasilinear time.

Recall that our algorithm has two steps. In the first step we aim to find a set of representative lines that belong to different clusters. In the second step we try to find centers on these representative lines covering all the lines. We start by specifying Step 1 and 2 of our algorithm for decision k-center under the assumption that we are given a radius r satisfying $r_3(\mathcal{L}) \le r \le r_3(\mathcal{L})(1 + \varepsilon)$. We will also assume throughout that the set of lines $\mathcal{L}$ does not have a 2-clustering of radius $(2 + \varepsilon)r$ with centers on lines in $\mathcal{L}$. We start by stating the following technical Lemma whose proof appears in the Appendix.

LEMMA 4.1. *Let ℓ and ℓ_1 be two lines. Let $r > 0$. Let $I(r) = \{p \in \ell \mid d(p, \ell_1) \le r\}$. Then $I(r(1 - \varepsilon)) \subseteq I(r)^{-\varepsilon/2} \subseteq I(r)$.*

LEMMA 4.2 (STEP 1). *Let $\mathcal{L}$ be a set of n lines in $\mathbb{R}^d$. Let c_1, c_2, c_3 be the centers and r the radius of a 3-clustering for $\mathcal{L}$. Let $\varepsilon > 0$ be sufficiently small. Assume that $\mathcal{L}$ does not have a 2-clustering of radius $(2 + \varepsilon)r$ with centers on lines in $\mathcal{L}$. Then one can find in time $O\left(nd + n \log^2 n + \frac{\log^2 n}{\varepsilon}\right)$, a list of triplets of lines $\{(\ell_1^i, \ell_2^i, \ell_3^i)\}_{i=1}^{O(1/\varepsilon)}$ s.t. for some i it holds that ℓ_1^i, ℓ_2^i and ℓ_3^i belong to different clusters w.r.t. the centers c_1, c_2 and c_3 and radius r. Namely, it is the case that $d(\ell_j^i, c_j) \le r$ for $j = 1, 2, 3$.*

PROOF. The set of triplets we construct will be of size $k = 48/\varepsilon + 8$. Recall that $\mathcal{L}$ can not be clustered by 2 centers on lines of $\mathcal{L}$ with radius $(2 + \varepsilon)r$. With the same idea as in Lemma 2.1 we can find two sets of lines (ℓ_1, ℓ_2), (ℓ_1', ℓ_2') such that at least one pair of lines do not belong to the same cluster. We set $\ell_1^i = \ell_1$, $\ell_2^i = \ell_2$ for $i = 1, \ldots, k/2$, and $\ell_1^i = \ell_1'$, $\ell_2^i = \ell_2'$ for $i = k/2 + 1, \ldots, k$.

We now proceed (w.l.o.g.) under the assumption that $d(\ell_1, c_1) \le r$, $d(\ell_2, c_1) > r$ and $d(\ell_2, c_2) \le r$. We now look for the third line not in the same cluster as ℓ_1 or ℓ_2. For any other line ℓ_j in $\mathcal{L} \setminus \{\ell_1, \ell_2\}$. Let $I_j^1 = \{p \in \ell_1 \mid d(p, \ell_j) \le (2 + \varepsilon)r\}$ be defined as before. Similarly define the interval $I_j^2 = \{p \in \ell_2 \mid d(p, \ell_j) \le (2 + \varepsilon)r\}$. Consider any isometric mapping ϕ_1 of ℓ_1 and ϕ_2 of ℓ_2 to the one dimensional line $\mathbb{R}^1$. Denote the image of I_j^1 and I_j^2 under these mappings as $\phi_1(I_j^1)$ and $\phi_2(I_j^2)$ respectively. Consider the cross $S_j = \phi_1(I_j^1) \dagger \phi_2(I_j^2)$. We notice that the crosses S_j corresponding to lines in $\mathcal{L} \setminus \{\ell_1, \ell_2\}$ have empty intersection, since $\mathcal{L}$ can not be 2-clustered.

Now, by Theorem 1.2 there exists a family of crosses $\mathcal{S}'$ of size $24/\varepsilon + 4$ such that $\cap_{S \in \mathcal{S}'} S^{-\frac{\varepsilon}{6}} = \emptyset$. For each cross $S_j \in \mathcal{S}'$ corresponding to line ℓ_j define $\ell_3^i = \ell_j$, $i = j$ or $i = 24 + j$. We now claim that one of the lines ℓ_3^i defined above satisfies $d(\ell_3^i, c_3) \le r$.

Consider the centers c_1 and c_2 and project them onto ℓ_1 and ℓ_2 respectively. Denote the corresponding points p'_1 and p'_2. Let $p_1 = \phi_1(p'_1) \in \mathbb{R}^1$ and $p_2 = \phi_2(p'_2) \in \mathbb{R}^1$. The point (p_1, p_2) is not included in $\cap_{S \in \mathcal{S}'} S^{-\frac{\varepsilon}{6}}$, thus there is at least one cross $S \in \mathcal{S}'$ such that $(p_1, p_2) \notin S^{-\frac{\varepsilon}{6}}$. We denote this cross by S_3. Let ℓ_3 be the line corresponding to cross S_3. Notice that ℓ_3 is equal to ℓ_3^i for some i. Hence, it suffices to prove that $d(\ell_3, c_3) \leq r$. We will show that $d(\ell_3, c_1) > r$ and $d(\ell_3, c_2) > r$, this will imply $d(\ell_3, c_3) \leq r$.

In what follows we show that $d(\ell_3, c_1) > r$, a similar argument holds for $d(\ell_3, c_2) > r$. Let $S_3 = \phi_1(I_3^1) \dagger \phi_2(I_3^2)$. Recall that $I_3^1 = \{p \in \ell_1 | d(p, \ell_3) \leq (2 + \varepsilon)r\}$ and $I_3^2 = \{p \in \ell_2 | d(p, \ell_3) \leq (2 + \varepsilon)r\}$. As $(p_1, p_2) \notin S_3^{-\frac{\varepsilon}{6}}$ we have that $p_1 \notin (\phi_1(I_3^1))^{-\frac{\varepsilon}{6}}$ and $p'_1 \notin (I_3^1)^{-\frac{\varepsilon}{6}}$. As I_3^1 was defined with respect to distance $(2+\varepsilon)r$, by Lemma 4.1 we have that $\{p \in \ell_1 | d(p, \ell_3) \leq (2 + \varepsilon)(1 - \frac{\varepsilon}{3})r\} \subseteq (I_3^1)^{-\frac{\varepsilon}{6}}$. Hence, $p'_1 \notin \{p \in \ell_1 | d(p, \ell_3) \leq (2 + \varepsilon)(1 - \frac{\varepsilon}{3})r\}$ which implies that $d(p'_1, \ell_3) > (2 + \varepsilon)(1 - \frac{\varepsilon}{3})r > 2r$. We conclude that it must be the case that $d(\ell_3, c_1) > r$ otherwise $d(p'_1, \ell_3) \leq d(p'_1, c_1) + d(\ell_3, c_1) \leq 2r$. (Note that $d(p'_1, c_1) \leq r$, since p'_1 is the projection of c_1 on ℓ_1 and $d(c_1, \ell_1) \leq r$.) This concludes our existence proof. The running time of finding the k triplets follows from the running time specified in Theorem 1.3. □

Lemma 4.3 (Step 2). *Let $\varepsilon > 0$. Let ℓ_2 and ℓ_3 be two lines in a set of lines $\mathcal{L}$ and given r. There exists a dynamic data structure on $\mathcal{L}$ of size at most n which answers queries of the form "Is there a 2-cluster of $\mathcal{L}$ with one center on ℓ_2 and the other on ℓ_3 of radius smaller than $2r$" with the following properties: (i) Insert a line in amortized time $O(d + \log^2 n)$. (ii) Delete a line in amortized time $O(\log^2 n)$. (iii) Support ε-approximate constructive query: If the optimal 2-clustering with one center on ℓ_2 and the other on ℓ_3 has radius $r^* \leq 2r(1 - \varepsilon)$ answer positively and return corresponding centers (that enable a $2r$ radius 2-cluster). Query time of (iii) is $O(\log^2 n / \varepsilon)$.*

Proof. Consider a set of lines $\mathcal{L}$. For each line $\ell_j \in \mathcal{L}$ we define the intervals $I_j^2 = \{p \in \ell_2 | d(p, \ell_j) \leq 2r\}$ and $I_j^3 = \{p \in \ell_3 | d(p, \ell_j) \leq 2r\}$. As in Lemma 4.2, consider any isometric mapping ϕ_2 of ℓ_2 and ϕ_3 of ℓ_3 to the one dimensional line $\mathbb{R}^1$. Denote the image of I_j^2 and I_j^3 under these mappings as $\phi_2(I_j^2)$ and $\phi_3(I_j^3)$ respectively. Consider the cross $S_j = \phi_2(I_j^2) \dagger \phi_3(I_j^3)$. Let $\mathcal{S}$ be the set of crosses corresponding to $\mathcal{L}$. As in Lemma 4.2, we want to check if the intersection of all the crosses in $\mathcal{S}$ is empty or not, and if not, find a common intersection (which will imply two centers, one on ℓ_2 and the other on ℓ_3, that cover $\mathcal{L}$ with radius $2r$). In fact, we will use the algorithm in Theorem 1.2 (with the implementation of Theorem 1.3) as a test procedure.

Now assume that $r^* \leq 2r(1 - \varepsilon)$. In this case we have that $\cap_{S \in \mathcal{S}} S \neq \emptyset$. Moreover, by Lemma 4.1 we have that $\cap_{S \in \mathcal{S}} S^{-\varepsilon/2} \neq \emptyset$. Thus the test procedure of Theorem 1.2 applied to $\mathcal{S}$ cannot output a set $\mathcal{S}'$ such that $\cap_{S \in \mathcal{S}'} S^{-\varepsilon/2} = \emptyset$. However, in this case (Corollary 3.1) the test procedure will find a point $p = (c_2, c_3) \in \cap_{S \in \mathcal{S}} S$. We conclude for each line $\ell_j \in \mathcal{L}$ that either $d(c_2, \ell_j) \leq 2r$ or $d(c_3, \ell_j) \leq 2r$. As the test procedure of Theorem 1.2, as described in Theorem 1.3, supports dynamic updates, we conclude our lemma. □

We would like to remark that although Lemma 4.3 does not answer the recursive decision problem exactly, it still suffices to prove Theorem 1.1. Namely, let $r \geq r_3(\mathcal{L})$. One may now run Step 1 of our algorithm on r and Step 2 on $2r(1 + \varepsilon)$

with approximation parameter $\varepsilon/2$. It is not hard to verify that this will yield an approximate 3-clustering of radius $\leq 2r(1+\varepsilon)$.

5. CONCLUSIONS AND FUTURE WORK

We have presented a new clustering algorithm for lines in Euclidean space. The main contribution is a new Helly-type theorem and its role in enabling algorithms with running time nearly linear in n and d, thus making the algorithms appealing in high dimensional applications.

Much work remains for the future: devising quasilinear-time approximation for increased (beyond 3) numbers of clusters; extending this work to flats of higher dimension (than lines); and exploring more fully the potential of our contractive Helly-type theorems in approximation algorithms.

Acknowledgments

REFERENCES

Agarwal, P., Har-Peled, S., and Varadarajan, K. R. 2005. Geometric approximation via coresets. In *Current Trends in Combinatorial and Computational Geometry*. Cambridge University Press.

Agarwal, P. K., Arge, L., and Yi, K. 2005. An optimal dynamic interval stabbing-max data structure? In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 803–812.

Agarwal, P. K. and Procopiuc, C. M. 2000. Approximation algorithms for projective clustering. In *SODA '00: Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 538–547.

Agarwal, P. K., Procopiuc, C. M., and Varadarajan, K. R. 2003. A $(1+\varepsilon)$-approximation algorithm for 2-line-center. *Comput. Geom. Theory Appl. 26,* 2, 119–128.

Allison, P. D. 2002. *Missing Data*. Sage Publications.

Alon, N. and Kalai, G. 1995. Bounding the piercing number. *Discrete and Computational Geometry 13,* 3/4, 245–256.

Amenta, N. 1996. A short proof of an interesting Helly-type theorem. *Discrete and Computational Geometry 15,* 4, 423–427.

Bădoiu, M., Har-Peled, S., and Indyk, P. 2002. Approximate clustering via core-sets. In *STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. ACM Press, New York, NY, USA, 250–257.

Demouth, J., Devillers, O., Glisse, M., and Goaoc, X. 2008. Helly-type theorems for approximate covering. *In proceedings of the twenty-fourth annual Symposium on Computational Geometry*, 120–128.

Eckhoff, J. 1993. Helly, Radon, and Carathéodory type theorems. *In Handbook of Convex Goemetry, P.M. Gruber and J.M Wills edt. A*, 389–448.

Feder, T. and Greene, D. H. 1988. Optimal algorithms for approximate clustering. In *Proc. 20th ACM Symposium on Theory of Computing*. 434–444.

Gao, J., Langberg, M., and Schulman, L. J. 2008. Analysis of incomplete data and an intrinsic-dimension helly theorem. *Discrete & Computational Geometry 40,* 4, 537–560.

González, T. 1985. Clustering to minimize the maximum intercluster distance. *Theoret. Comput. Sci. 38*, 293–306.

Grunbaum, B. and Motzkin, T. S. 1961. On components in some families of sets. *Proceedings of the American Mathematical Society 12,* 4, 607–613.
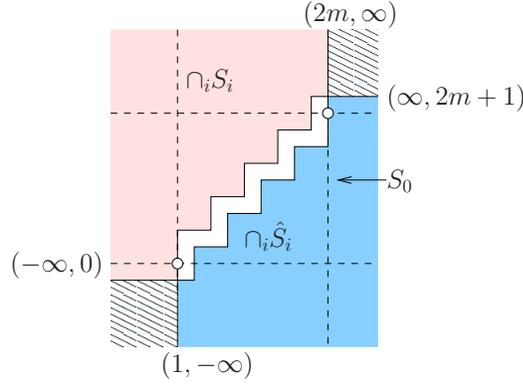
Fig. 3. The intersection of S_i and $\hat{S}_i$.

HAR-PELED, S. AND VARADARAJAN, K. 2002. Projective clustering in high dimensions using coresets. In *SCG '02: Proceedings of the eighteenth annual symposium on Computational geometry.* ACM Press, New York, NY, USA, 312–318.

HELLY, E. 1923. Über Mengen konvexer Körper mit gemeinschaftlichen Punkten. *Jahresbericht Deutsch. Math. Verein. 32*, 175–176.

HOCHBAUM, D. S. AND SHMOYS, D. B. 1985. A best possible heuristic for the k-center problem. *Mathematics of Operations Research 10,* 2, 180–184.

HOCHBAUM, D. S. AND SHMOYS, D. B. 1986. A unified approach to approximation algorithms for bottleneck problems. *J. ACM 33,* 3, 533–550.

KALAI, G. AND MESHULAM, R. 2008. Leray numbers of projections and a topological Helly-type theorem. *Journal of Topology 1,* 3, 551–556.

LANGBERG, M. AND SCHULMAN, L. J. 2007. Contraction and expansion of convex sets. In *Proceedings of the 19th Canadian Conference on Computational Geometry.* 25–28.

LARMAN, D. G. 1968. Helly type properties of unions of convex sets. *Mathematika 15*, 53–59.

LITTLE, R. J. A. AND RUBIN, D. B. 2002. *Statistical analysis with missing data*, second ed. John Wiley & Sons, Inc., Hoboken, NJ, USA.

MATOUŠEK, J. 1997. A Helly-type theorem for unions of convex sets. *Computational Geometry 18,* 1, 1–12.

MORRIS, H. 1973. Two pigeonhole principles and unions of convexly disjoint sets. *Ph. D. Thesis, Cal. Inst. of Tech., Calif..*

WENGER, R. 2004. Helly-type theorems and geometric transversals. *Handbook of discrete and computational geometry*, 73–96.

Appendix

A. PROOF OF CLAIMS

PROOF. **(of Lemma 3.1)** We define a set $\mathcal{S}$ of $n = 2m + 1$ crosses. The first m crosses are of the form $S_i = I_i \dagger J_i = [-\infty, i] \dagger [i+2, \infty]$, $i = 1, 3, 5, \ldots 2m - 1$. The second m crosses are of the form $\hat{S}_i = \hat{I}_i \dagger \hat{J}_i = [i, \infty] \dagger [-\infty, i-2]$, $i = 2, 4, \ldots 2m$. The final cross is $S_0 = I_0 \dagger J_0 = [1 + 1/2, 2m - 1/2] \dagger [1/2, 2m + 1/2]$. Please refer to Figure 3.

It is not hard to verify that the $2m + 1$ crosses have empty intersection. Or, the union of the complement of the crosses cover $\mathbb{R}^2$ completely. The boundary of the intersection of the S_i's has a sawtooth structure and consists of the line connecting the points: $(1, -\infty)$, $(1, 3)$, $(3, 3)$, $(3, 5)$, $\ldots$, $(2m - 1, 2m - 1)$, $(2m - 1, 2m + 1)$,

$(\infty, 2m + 1)$. The boundary of the intersection of the $\hat{S}_i$'s has a complementary sawtooth structure and consists of the line connecting the points: $(-\infty, 0)$, $(2, 0)$, $(2, 2)$, $(4, 2)$, ..., $(2m - 2, 2m - 2)$, $(2m, 2m - 2)$, $(2m, \infty)$. This implies that the intersection of the S_i's and $\hat{S}_i$'s consists of points (x, y) with $x \geq 2m$ and $y \geq 2m + 1$ or points (x, y) with $x \leq 1$ and $y \leq 0$. These regions do not intersect S_0. See Figure 3.

On the other hand, consider removing a single cross from $\mathcal{S}$ to obtain $\mathcal{S}'$. If we remove S_0, then by the discussion above $\mathcal{S}'$ has a non-empty intersection. If we remove S_i, it is not hard to verify that the point $(i + 1, i + 1)$ is in every set in $\mathcal{S}'$. Finally, if we remove $\hat{S}_i$, the point $(i - 1, i - 1)$ is in every set in $\mathcal{S}'$. $\square$

PROOF. **(of Lemma 3.2)** We use the construction defined above in the proof of Lemma 3.1. Let $\varepsilon > 0$ be sufficiently small and let $m = \lfloor 1/(8\varepsilon) \rfloor$. In what follows we will use the fact that $2m\varepsilon < 1/4$. We define a set $\mathcal{S}$ of $n = 2m + 1$ crosses identical to those of Lemma 3.1.

It is not hard to verify that S_i^ε is included in the cross $[i - 2m - 1/4, i + 1/4] \dagger [i + 2 - 1/4, i + 2 + 1/4 + 2m]$, that $\hat{S}_i^\varepsilon$ is included in the cross $[i - 1/4, i + 1/4 + 2m] \dagger [i - 2 - 1/4 - 2m, i - 2 + 1/4]$, and that $S_0^\varepsilon \subseteq [2 - 1/4, 2m - 1 + 1/4] \dagger [2 - 1/4, 2m - 1 + 1/4]$. It holds that the ε expansion of the $2m + 1$ crosses have empty intersection. The part of the boundary of the intersection of the S_i^ε's that is close to the line "$x = y$" has a sawtooth structure and consists of the line connecting the points: $(1 + 1/4, -\infty)$, $(1 + 1/4, 3 - 1/4)$, $(3 + 1/4, 3 - 1/4)$, $(3 + 1/4, 5 - 1/4)$, ..., $(2m - 1 + 1/4, 2m - 1 - 1/4)$, $(2m - 1 + 1/4, 2m + 1 - 1/4)$, $(\infty, 2m + 1 - 1/4)$. The same part of the boundary of the intersection of the $\hat{S}_i^\varepsilon$'s has a complementary sawtooth structure and consists of the line connecting the points: $(-\infty, 1/4)$, $(2 - 1/4, 1/4)$, $(2 - 1/4, 2 + 1/4)$, $(4 - 1/4, 2 + 1/4)$, ..., $(2m - 2 - 1/4, 2m - 2 + 1/4)$, $(2m - 1/4, 2m - 2 + 1/4)$, $(2m - 1/4, \infty)$. This implies that the intersection of the S_i^ε's and $\hat{S}_i^\varepsilon$'s consists of points (x, y) with $x \geq 2m - 1/4$ and $y \geq 2m + 1 - 1/4$ or points (x, y) with $x \leq 1 + 1/4$ and $y \leq 1/4$. These regions do not intersect S_0^ε.

On the other hand, by Lemma 3.1 removing even a single cross from $\mathcal{S}$ results in non-empty intersection. As $m = \lfloor 1/(8\varepsilon) \rfloor$ and $|\mathcal{S}| = 2m + 1$ we conclude our assertion. $\square$

PROOF. **(of Lemma 4.1)** We prove $I(r(1 - \varepsilon)) \subseteq I(r)^{-\varepsilon/2}$ as the second inclusion is immediate. The lines ℓ and ℓ_1 lie in some three dimensional space. We will assume w.l.o.g. that the line ℓ lies on the x axis (namely, a parameterized description of ℓ is $(1, 0, 0)t$) and that the line ℓ_1 can be described by $(a, b, 0)t + (0, 0, h)$ for non negative a, b and h with $a^2 + b^2 = 1$. This implies that the minimum distance between ℓ and ℓ_1 is h and is obtained with the points $(0, 0, 0)$ on ℓ and $(0, 0, h)$ on ℓ_1. With this setting, it is not hard to verify that the square distance between a point $(x, 0, 0)$ on ℓ and the line ℓ_1 is $h^2 + x^2 b^2$.

Let $x(r)$ satisfy $h^2 + x(r)^2 b^2 = r^2$. Namely, $x(r) = \sqrt{\frac{r^2 - h^2}{b^2}}$. It follows that the interval $I(r)$ is equal to $[-x(r), x(r)]$. Thus it is left to show that $x(r) - \varepsilon x(r) \geq x(r(1 - \varepsilon))$ or equivalently $(1 - \varepsilon)x(r) \geq x(r(1 - \varepsilon))$ (indeed this will imply that $I(r(1 - \varepsilon)) \subset I(r)^{-\varepsilon/2}$). The inequality $(1 - \varepsilon)x(r) \geq x(r(1 - \varepsilon))$ follows directly from the definition of $x(r)$. $\square$

B.   A CONSTANT APPROXIMATION BASED ON CORESETS

We now present a 58-approximate k-center clustering algorithm for lines based on the notion of coresets. We start with the following claim:

CLAIM B.1. *Let $\mathcal{L} = \{\ell_1, \ldots, \ell_n\}$ be a set of n lines in $\mathbb{R}^d$. For a constant k, a 2-approximate k-center clustering of $\mathcal{L}$ can be found in time $O(dn^{4k+1})$.*

PROOF. We use the naive algorithm described in Section 2.1 to obtain the asserted running time. Let $c_1^*, \ldots, c_k^*$ be the cluster centers and $r_1^*, \ldots, r_k^*$ be the cluster radii in an optimal k-center clustering of $\mathcal{L}$. Let $\mathcal{L}_i$ be a partition of $\mathcal{L}$ implied by this optimal clustering. Namely, for every $\ell \in \mathcal{L}_i$ it holds that $d(c_i^*, \ell) \leq r_i^*$. Or in other words, the optimal 1-center of $\mathcal{L}_i$ is exactly c_i^* with radius r_i^*.

Consider any set $\mathcal{L}_i$. Let $r_i \leq r_i^*$ be the maximum of the 1-center radius of all triplets of lines in $\mathcal{L}_i$. In [Gao et al. 2008] it is shown, that for any line $\ell \in \mathcal{L}_i$ there exists a point c_i on ℓ such that the ball of radius $2r_i$ centered at c_i covers all lines in $\mathcal{L}_i$.

This implies the following algorithm. For $i = 1, \ldots, k$, exhaustively guess the triplet $\{\ell_1^i, \ell_2^i, \ell_3^i\}$ in $\mathcal{L}_i$ which has maximal 1-center radius out of all line triplets in $\mathcal{L}_i$. For each such triplet $\{\ell_1^i, \ell_2^i, \ell_3^i\}$, compute its optimal 1-center radius r_i. Now using the naive algorithm appearing in Section 2.1, with the radii $\{r_i\}_{i=1}^k$ and the lines $\{\ell_1^i\}_{i=1}^k$, (exhaustively) find the centers c_i. Namely, for each line ℓ_1^i we define for every line $\ell_j \in \mathcal{L}$ a corresponding interval $I_j = \{x \in \ell_1^i | d(x, \ell_j) \leq 2r_i\}$. These intervals define $O(n)$ regions on ℓ_1^i, one of which corresponds to $\mathcal{L}_i$. □

We sketch a method for finding a *small* coreset of lines $\mathcal{L}'$ in $\mathcal{L}$ with the property that the optimal k-center of $\mathcal{L}'$ is a good approximation to the optimal k-center of $\mathcal{L}$. More specifically, any k-center of $\mathcal{L}'$ of radius r with centers $c_1, \ldots, c_k$ will imply a k-center clustering of $\mathcal{L}$ of radius $29r$ centered at $c_1, \ldots, c_k$. We would like to find the coreset in time which is quasi-linear in $n = |\mathcal{L}|$ and d. Once the coreset has been found, we can find an approximate k center using Claim B.1. The proof we present holds for any constant k.

The proof has three major steps. Let r^* be the optimal k-center clustering radius of $\mathcal{L}$. Roughly speaking, in the first step we find a k'-center clustering of $\mathcal{L}$ of radius $4r^*$. This clustering partitions the lines in $\mathcal{L}$ into k' sets. As the lines in each such subset $\mathcal{P}$ are covered by a ball of *small* radius, they can be perturbed so that all lines intersect. In the second step, for each perturbed subset $\tilde{\mathcal{P}}$ of lines above we use the fact that they intersect to find a coreset $\tilde{\mathcal{P}}' \subseteq \tilde{\mathcal{P}}$. Specifically, any k center clustering of $\tilde{\mathcal{P}}'$ of radius r will imply an approximate clustering of $\tilde{\mathcal{P}}$ (with the same k centers). These coresets will be of size exponential in k. Now taking the final coreset to be the union of the (original) lines in $\mathcal{P}$ corresponding to $\tilde{\mathcal{P}}'$, we obtain the desired coreset $\mathcal{L}'$. Once we have a small coreset $\mathcal{L}'$, we can use the algorithm in Claim B.1 to find a 2-approxmate clustering of the coreset, and thus a corresponding approximate clustering of $\mathcal{L}$ (this is the third and final step of our algorithm). In what follows, we elaborate on the steps 1 and 2 described above.

B.1   Step 1

Let $c_1, \ldots, c_k$ be the cluster centers and $r_1^*, \ldots, r_k^*$ be the cluster radii in an optimal k-center clustering of $\mathcal{L}$. Let $r^* = \max_i r_i^*$. Let $\mathcal{L}_i$ be a partition of $\mathcal{L}$ implied by this optimal clustering. Namely, for every $\ell \in \mathcal{L}_i$ it holds that $d(c_i, \ell) \leq r_i^*$. For

every pair of lines ℓ_i, ℓ_j let x_i^j be the point on ℓ_i that is closest to ℓ_j. Let X denote the set $\{x_i^j\}_{ij}$. We now specify a technical lemma, the lemma is phrased for the set $\mathcal{L}_1$ but it holds analogously for any $\mathcal{L}_i$.

CLAIM B.2. *Let c_1 be the cluster center of $\mathcal{L}_1$. Let ℓ_i be any line in $\mathcal{L}_1$. Let c_1' be the projection of c_1 to ℓ_i. For any subset of lines $\mathcal{L}_1' \subseteq \mathcal{L}_1$, let x be the point in $\{x_i^j\}_{\ell_j \in \mathcal{L}_1'}$ closest to c_1'. The point x satisfies: $\forall \ell_j \in \mathcal{L}_1'$, $d(\ell_j, x) \leq 4d(\ell_j, c_1) \leq 4r^*$.*

PROOF. It is not hard to verify that for $\ell_j \in \mathcal{L}_1'$: $d(\ell_j, x) \leq 2d(\ell_j, c_1') \leq 4d(\ell_j, c_1)$. The first inequality follows from the structure of the function $d(\ell_j, \alpha)$ when α moves along the line ℓ_i. This function obtains its minimum at $\alpha = x_i^j$, and we use the fact that the distance between x_i^j and c_1' is larger than the distance between c_1' and x. □

We use the above observation to construct an efficient algorithm for finding $k' \simeq O(k^2 \log n)$ centers which cover the lines $\mathcal{L}$ within radius $4r^*$. Our algorithm will have $\simeq \log n$ iterations. In each iteration we will pick $\simeq k^2$ centers with the property that a significant fraction of the lines in $\mathcal{L}$ will be of distance $4r^*$ from these centers.

We describe the first iteration in detail, the remaining iterations have a similar structure. We start with some notation. A point x_i^j is said to be *good* if the following conditions hold: (a) Both ℓ_i and ℓ_j are in $\mathcal{L}_a$ for some $a \in [k]$. (b) For this value of a, consider the projection of c_a onto ℓ_i: c_a'. We request that the point x_i^j is relatively close to c_a'. Namely that at least half of the points in the set $\{x_i^j\}_{\ell_j \in \mathcal{L}_a}$ have distance to c_a' which is greater or equal than $d(x_i^j, c_a')$. For $a \in [k]$ let n_a be the size of $\mathcal{L}_a$. It holds that $\sum_a n_a = n$ and it is not hard to verify that there are at least $\sum_a n_a^2/2 \geq n^2/2k$ good points x_i^j in X. Moreover,

CLAIM B.3. *Let x_i^j be a good point in which ℓ_i and ℓ_j are in $\mathcal{L}_a$. Then for at least half the lines $\ell \in \mathcal{L}_a$ it holds that $d(\ell, x_i^j) \leq 4r^*$.*

PROOF. Consider the projection of c_a onto ℓ_i. To obtain our assertion, we use Claim B.2 with $\mathcal{L}_a'$ defined to be the lines $\ell_{j'} \in \mathcal{L}_a$ for which $d(x_i^{j'}, c_a') \geq d(x_i^j, c_a')$. □

By the above, for a good point x_i^j with ℓ_i and ℓ_j in $\mathcal{L}_a$ there are at least $n_a/2$ lines ℓ for which $d(\ell, x_i^j) \leq 4r^*$. Thus, the $n_a/2$ closest lines to x_i^j in $\mathcal{L}$ satisfy $d(\ell, x_i^j) \leq 4r^*$. Consider the set Γ of values of a for which $n_a \geq n/(2k)$. It holds that for a good point x_i^j with corresponding a in Γ, there are at least $n/(4k)$ lines ℓ for which $d(\ell, x_i^j) \leq 4r^*$. In the current iteration of our algorithm, we would like to pick at least one point x_i^j as above (namely, with corresponding a in Γ). A basic calculation shows that there are at least $n^2/(4k)$ good points x_i^j with corresponding a in Γ. Thus, if we pick $\alpha k \log(k \log n)$ points x_i^j uniformly at random, for a sufficiently large constant α, with probability $1 - 1/(8k \log n)$ we will choose at least one point x_i^j as desired. Denoting this set of $\alpha k \log(k \log n)$ points x_i^j by C, it now follows that there are at least $n/(4k)$ lines $\ell \in \mathcal{L}$ such that $d(\ell, C) \leq 4r^*$. Removing the closest $n/(4k)$ lines in $\mathcal{L}$ to the centers C, we move to the next iteration where we have only $n(1 - 1/(4k))$ lines that are not covered by C.

For the remaining iterations, we consider only the set of lines not yet covered by our set of centers, and preform the exact same random iteration and analysis. All in all, we will need $O(k \log n)$ iterations to cover all lines (we stop once the set of uncovered lines is of size $O(\log n)$). The result is a set of $O(k^2 \log n \log(k \log n))$ centers which cover $\mathcal{L}$ within radius $4r^*$. The running time of the algorithm is $\tilde{O}(nd)$. The algorithm will succeed with probability at least $1/2$, however this can be boosted using standard methods.

B.2   Step 2

In the second step of the suggested algorithm, we consider a ball $\mathcal{B}$ of radius $r \leq 4r^*$ from the k'-center clustering of $\mathcal{L}$ above. Let $\mathcal{P}$ be the set of lines covered by $\mathcal{B}$. We find a subset of lines $\mathcal{P}' \subseteq \mathcal{P}$ with the property that any k-center clustering of $\mathcal{P}'$ of radius r' implies a k-center clustering of $\mathcal{P}$ with the same centers and radius $5(r' + r) + r$. As this can be done for every ball $\mathcal{B}$ as above, taking the union of the corresponding subsets $\mathcal{P}'$ will yield the desired coreset $\mathcal{L}'$.

We start by *shifting* the lines in $\mathcal{P}$ so that they all intersect. Each line preserves its orientation and is additively translated by a distance of at most r. This is easily done as the lines in $\mathcal{P}$ are all of distance at most r from a common point.

Now consider the intersection of the set $\tilde{\mathcal{P}}$ (of the shifted lines in $\mathcal{P}$) and the boundary of $\mathcal{B}$. This intersection consists of $2|\tilde{\mathcal{P}}|$ points (2 points for each line in $\tilde{\mathcal{P}}$). Denote the set of points on the boundary of $\mathcal{B}$ by $\mathcal{Q}$. Roughly speaking, we now show that a 5-*multiplicative coreset* $\mathcal{Q}'$ for the $2k$-center problem on $\mathcal{Q}$ implies the coreset $\mathcal{P}'$ we are looking for. For $\delta > 0$ and a set of points $\mathcal{Q}$, a $(1 + \delta)$-multiplicative coreset $\mathcal{Q}'$ w.r.t. the k-center problem, is a set of points for which any cover of $\mathcal{Q}'$ with any k balls of radius $r_1, \ldots, r_k$ implies a cover of $\mathcal{Q}$ with k balls of radius $(1 + \delta)r_1, \ldots, (1 + \delta)r_k$ (the latter balls being a blown up version of the former).

We start by considering the following mapping between balls in $\mathbb{R}^d$ (that do not contain the origin) and pairs of balls in $\mathbb{R}^d$. For simplicity, assume that $\mathcal{B}$ is centered at the origin and has unit radius. Let β be a ball of radius r_β centered at c_β which does not contain the origin. Let C_β be all points p on the boundary of $\mathcal{B}$ for which the line passing through p and the origin intersect β. The set C_β consists of two caps on the boundary of $\mathcal{B}$. Notice that a line in $\tilde{\mathcal{P}}$ is covered by β if and only if its corresponding points in $\mathcal{Q}$ are covered by C_β. For any $\delta > 0$, let β^δ be the $(1 + \delta)$ expansion of the ball β, namely β^δ is the ball centered at c_β of radius $(1 + \delta)r_\beta$ (assume also that β^δ does not contain the origin). Let C_{β^δ} be the set corresponding to β^δ. We now define the two balls corresponding to β. The balls are defined to satisfy two properties: first the intersection of the balls with the boundary of $\mathcal{B}$ is exactly the caps in C_β. Secondly, the intersection of the $(1 + \delta)$ expansion of the balls with the boundary of $\mathcal{B}$ is exactly the caps in C_{β^δ}. It is not hard to verify that such a pair of balls exist. We denote these balls by B_β and $-B_\beta$ (as one is the symmetric image of the other with respect to the origin).

Let $\mathcal{Q}'$ be a $(1 + \delta)$ multiplicative coreset for the $2k$ center problem on $\mathcal{Q}$, and let $\tilde{\mathcal{P}}'$ be the set of (shifted) lines corresponding to points in $\mathcal{Q}'$. We now show that $\tilde{\mathcal{P}}'$ is a coreset for the k-center problem on $\tilde{\mathcal{P}}$. Afterwards we will show that the original lines (before they were shifted) corresponding to $\tilde{\mathcal{P}}'$ are a coreset for $\mathcal{P}$.

Let $\beta_1, \ldots, \beta_k$ be a k-center clustering for $\tilde{\mathcal{P}}'$ of radius $r_1, \ldots, r_k$ accordingly, we

show that the balls obtained by expanding β_i to radius $(1+\delta)r_i$ cover all of $\tilde{\mathcal{P}}$. If any one of the expanded balls β_i covers the origin, then we are done. Assume otherwise. For $i \in \{1, 2, \ldots, k\}$ let B_{β_i} and $-B_{\beta_i}$ be the balls defined above corresponding to β_i. By the discussion above, this set of $2k$ balls covers $\mathcal{Q}'$. Thus their expansion by a factor of $(1 + \delta)$ cover all $\mathcal{Q}$. Moreover, these expanded balls correspond to the $(1 + \delta)$ expansion of $\beta_1, \ldots, \beta_k$. We conclude that the $(1 + \delta)$ expansion of the balls β_i cover $\tilde{\mathcal{P}}$.

Now consider the original lines (before shifting) $\mathcal{P}$ and the coreset $\mathcal{P}'$ corresponding to $\tilde{\mathcal{P}}'$. Consider a k-center clustering for $\mathcal{P}'$ with centers $c_1, \ldots, c_k$ and radii $r_1, \ldots, r_k$ accordingly. Then the balls of radius $r_i + 4r^*$ centered at c_i cover all shifted lines $\tilde{\mathcal{P}}'$. From the above analysis, the balls of radius $(r_i + 4r^*)(1 + \delta)$ centered at c_i cover all shifted lines $\tilde{\mathcal{P}}$. Finally, the balls of radius $(r_i+4r^*)(1+\delta)+4r^*$ centered at c_i cover the original lines $\mathcal{P}$. This gives an overall ratio of $9+5\delta$. Fixing $\delta = 4$ we get the an approximation ratio of 29.

To conclude our proof, it is left to show how to efficiently find a small 5-multiplicative (*i.e.*, $\delta = 4$) coreset for a given set of points $\mathcal{P}$ of size n.

CLAIM B.4. *Given a set of points $\mathcal{Q}$ of size n, a 5-multiplicative coreset $\mathcal{Q}'$ for the k center problem of size $O(k^k)$ can be found in time $O(ndk^k \log k)$.*

PROOF. Our proof will be inductive (on k) and follows the line of proof given in [Agarwal et al. 2005]. Consider the simple greedy 2-approximation algorithm of [González 1985; Hochbaum and Shmoys 1985; 1986] for the k center problem on $\mathcal{Q}$. Let $\mathcal{Q}'$ be the set of points of size $k + 1$, and $\mathcal{B}_1, \ldots \mathcal{B}_k$ be the k-clustering of $\mathcal{Q}$ implied by [González 1985; Hochbaum and Shmoys 1985; 1986]. Let r^* be the optimal k-center radius of $\mathcal{Q}$, and let $r = r(\mathcal{B}_i)$ (all balls $\mathcal{B}_i$ have the same radius). In [González 1985; Hochbaum and Shmoys 1985; 1986] it is shown that $2r^* \geq r \geq r^*$, and that any pair of points $q_1, q_2 \in \mathcal{Q}'$ satisfy $d(q_1, q_2) \geq r$. Note that $\mathcal{Q}'$ is not a multiplicative coreset for $\mathcal{Q}$. The set $\mathcal{Q}'$ and the balls $\mathcal{B}_i$ can be found in time $O(nd \log k)$ [Feder and Greene 1988].

For each ball $\mathcal{B}_i$ let $\mathcal{Q}_i$ be the points of $\mathcal{Q}$ included in $\mathcal{B}_i$. Let $\mathcal{Q}'_i$ be a 5-multiplicative coreset for the $(k - 1)$-center problem on $\mathcal{Q}_i$. We now show that $\Gamma = \mathcal{Q}' \cup \cup_i \mathcal{Q}'_i$ is a 5-multiplicative coreset for the k-center problem on $\mathcal{Q}$.

Indeed, let $\mathcal{C}_1, \ldots, \mathcal{C}_k$ by any k balls covering Γ of radius $r_1, \ldots, r_k$. Consider a ball $\mathcal{B}_i$. We will show that a multiplicative blowup of the balls $\mathcal{C}_i$ will include all points of $\mathcal{Q}$ in $\mathcal{B}_i$. As $\mathcal{C}_1, \ldots, \mathcal{C}_k$ cover the $k + 1$ points in $\mathcal{Q}'$, the analysis of [González 1985; Hochbaum and Shmoys 1985; 1986] implies that at least one ball $\mathcal{C}_i$ has radius $\geq r^*/2$. If $\mathcal{B}_i$ intersects all balls $\mathcal{C}_1, \ldots, \mathcal{C}_k$ (and namely the ball $\mathcal{C}_i$), then blowing up the balls $\mathcal{C}_i$ by a factor of 5 will include all of $\mathcal{B}_i$. Otherwise, the points of Γ in the ball $\mathcal{B}_i$ are covered by $(k - 1)$ balls in $\mathcal{C}_1, \ldots, \mathcal{C}_k$. These $(k - 1)$ balls cover $\mathcal{Q}'_i$, and thus by induction we have that blowing them up by a factor of 5 will cover all points in $\mathcal{B}_i$.

All in all, the size of the coreset Γ satisfies $S(k) = kS(k-1)+(k+1)$, which implies that $|\Gamma| = O(k^k)$. The construction takes time $T(k) = kT(k-1)+O(nd \log k)$ which implies that $T(k) = O(ndk^k \log k)$. □

We remark that the running time of this algorithm is $O(dn \log n \log \log n)$ (for constant k). In the first step, we find an $O(\log n \log \log n)$-center clustering of $\mathcal{L}$ in time $O(dn \log n \log \log n)$. In the second step, we use this clustering to find a coreset $\mathcal{L}'$ of size $O(\log n \log \log n)$ in time $O(dn \log n \log \log n)$. Finally, in the third step,

we apply Claim B.1 to obtain an approximate k-clustering for our coreset. The final step is done in time $O(d(\log n \log \log n)^{4k+1})$.

C. $(1 + \varepsilon)$-APPROXIMATE CLUSTERING

We can extend Theorem 1.1, which provides a constant-factor approximation, to $(1 + \varepsilon)$-approximate clustering, but only in a runtime exponential in d (and $\log 1/\varepsilon$). This is impractical for the high-dimensional applications explained in the introduction—hence relegation of this material to the Appendix. (Additionally, the methods involved are less novel than the material in the body of the paper.) Namely, we show

COROLLARY C.1. *For a set of n lines $\mathcal{L}$ in $\mathbb{R}^d$, one can solve the $(1 + \varepsilon)$-approximate k-center problem in time $O\left((nd + n\log n)\left(\frac{c}{\varepsilon}\right)^{d-1}\log 1/\varepsilon\right)$ and $O\left(\left(nd + n\frac{\log^2 n}{\varepsilon}\right)\left(\frac{c}{\varepsilon}\right)^{d-1}\log 1/\varepsilon\right)$ for $k = 2$ and $k = 3$ respectively. Here $c > 0$ is a constant independent of ε, n and d.*

Once one allows an exponential (in d and $\log 1/\varepsilon$) running time, a result similar to that of Corollary C.1 can also be obtained directly using standard "coreset" based techniques. However, existing coreset based techniques do not imply Theorem 1.1 which states a quasilinear running time in d and n (for our $(2+\varepsilon)$-approximation). Achieving a $(1 + \varepsilon)$-approximation with running time polynomial in d and $1/\varepsilon$ remains open.

We now present both proofs for the $(1 + \varepsilon)$-approximate clustering of lines. The first proof is based on Theorem 1.1. The second proof is based on the algorithm presented in Section B, and does not use the result of Theorem 1.1.

C.1   Improving the approximation ratio of Theorem 1.1 using ε-nets

We now consider improving the approximation ratio of the naive algorithm from Section 2.1 from 2 to $(1 + \varepsilon)$. Namely, given a set of lines $\mathcal{L}$ and a radius r, we would like to output a k-clustering of $\mathcal{L}$ of radius at most $(1 + \varepsilon)r$ or report that $r < r_k(\mathcal{L})$. Improving the approximation ratio of our $2, 3$-clustering algorithms is done in an analogous manner (and will yield Corollary C.1).

To improve the approximation ratio, we find in Step 1 a set of k lines (not necessarily in $\mathcal{L}$), say $\hat{\ell}_1, \ldots, \hat{\ell}_k$, with the property that $d(\hat{\ell}_i, c_i) \leq \varepsilon r$, for all $1 \leq i \leq k$. It is not hard to verify that this suffices to imply a $(1 + \varepsilon)$-approximation factor. Finding a line $\hat{\ell}_1$ such that $d(\hat{\ell}_1, c_1) \leq \varepsilon r$ is done exhaustively, using an ε-net for the unit ball $\mathcal{B}^{d-1}$ in $\mathbb{R}^{d-1}$. Consider a set of points Γ of size $\left(\frac{\sqrt{4e\pi}}{\varepsilon}\right)^{d-1}$ with the property that for any vector $v \in \mathbb{R}^{d-1}$ of length ≤ 1 it holds that $d(v, \Gamma) \leq \varepsilon$. Such a set can be obtained by the intersection of a standard $\varepsilon/\sqrt{d}$ grid with $\mathcal{B}^{d-1}$. For a line ℓ_i, let $\mathcal{B}_i$ be the set of vectors $v \in \mathbb{R}^d$ of length ≤ 1 that are orthogonal to (the direction of) ℓ_i. $\mathcal{B}_i$ is a ball in $\mathbb{R}^{d-1}$. Let Γ_i be an ε-net for $\mathcal{B}_i$. Step 1 now searches over all k tuples of lines $\ell_1, \ldots, \ell_k$ in $\mathcal{L}$ and over all k-tuples of points $\gamma_1, \ldots, \gamma_k$ in the corresponding sets $\Gamma_1, \ldots, \Gamma_k$ (respectively). The resulting subset of lines considered is $\hat{\ell}_i = \ell_i + r\gamma_i = \{p | p - r\gamma_i \in \ell_i\}$. It is not hard to verify that for some k tuple of lines $\hat{\ell}_1, \ldots, \hat{\ell}_k$ it will hold that $d(\hat{\ell}_i, c_i) \leq \varepsilon r$ for $i = 1, \ldots, k$.

C.2   A proof based on coresets

The proof follows the line of proof given in Section B with two main differences
that are possible due to the fact that we allow exponential running time (in d and
$\log 1/\varepsilon$). In Step 1 of the scheme described in Section B. we do not cover the given
lines with $O(k^2)$ ball of radius $4r^*$. Rather each such ball $\mathcal{B}$ is covered by balls of
radius εr^* (using standard covering techniques). Namely, as done in Section C, we
efficiently construct a set of points $\Gamma_{\mathcal{B}} \in \mathcal{B}$ of size $\left(\frac{4\sqrt{4e\pi}}{\varepsilon}\right)^{d-1}$ with the property
that for any line ℓ covered by $\mathcal{B}$ it holds that $d(\ell, \Gamma_{\mathcal{B}}) \leq \varepsilon r^*$. Once this is done
for all the $4k^2$ balls that cover $\mathcal{L}$, we obtain a set of points $\Gamma = \cup_{\mathcal{B}} \Gamma_{\mathcal{B}}$ with the
property that Γ is a $k' = 4k^2 \left(\frac{4\sqrt{4e\pi}}{\varepsilon}\right)^{d-1}$ - center clustering of $\mathcal{L}$ with radius εr^*.

For the second step of the scheme presented in Section B, we use a $(1 + \varepsilon)$-
multiplicative coreset for $\mathcal{Q}$ instead of a 5-multiplicative one. Such a coreset can
be found in the desired running time using ideas stated in [Agarwal et al. 2005].