Spring 2016

Coresets, complexity of shapes, and total sensitivity

Xin Xiao
University of Iowa

CORESETS, COMPLEXITY OF SHAPES, AND TOTAL SENSITIVITY

by

Xin Xiao

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Computer Science
in the Graduate College of
The University of Iowa

May 2016

Thesis Supervisor: Professor Kasturi Varadarajan

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Xin Xiao

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Computer Science at the May 2016 graduation.

Thesis Committee: ______________________
Kasturi Varadarajan, Thesis Supervisor

____
Sriram Pemmaraju

____
Sukumar Ghosh

____
Alberto Segre

____
Nariankadu Shyamalkumar

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Dr. Kasturi Varadarajan, for his patience and guidance in doing my research and writing my paper. In particular, his understanding in my having to take a break in the middle of my studies is greatly appreciated. I would also like to thank the other members of my committee, Sukumar Ghosh, Sriram Pemmaraju, Alberto Segre, Nariankadu Shyamalkumar, for their generosity in giving time to me to help clarifying my ideas and solidifying my research. All of my committee members made my experience a rewarding and satisfying one. I would also like to thank my mother and husband for their understanding and support in the difficult times finishing my paper and preparing for my defense.

ABSTRACT

In this document, we consider coreset and total sensitivity for shape fitting problems. The shape fitting problems that are of considerable interest include: (1) (j, k) projective clustering problem, and (2) circle fitting problem on the plane. In (j, k) projective clustering, we are given a finite set of points P in d-dimensional Euclidean space, and the goal is to find a shape, which is a k-tuple j-flats (affine j-subspace), that best fits P. In circle fitting problem, given an input point set $P \subset \mathbb{R}^2$, the goal is to find a circle that best fits P. In L_1-fitting, the cost of fitting P to a shape F is defined as $\sum_{p \in P} \operatorname{dist}(p, F)$, where $\operatorname{dist}(p, F)$ is the cost of assigning p to F, while in L_∞-fitting, $\max_{p \in P} \operatorname{dist}(p, F)$. We focus on L_1-fitting.

A coreset is a compact representation of the input point set. For a shape fitting problem, a coreset for a point set P is a weighted point set, with the property that the cost of fitting the coreset to a shape F approximates the cost of fitting P to F, for every shape in the family of shapes. Coreset of small (e.g., constant) cardinality is of interest, because one can afford to use off-shelf, perhaps computationally expensive algorithms to solve the geometric optimization problem for the coreset, and a good solution for the coreset is guaranteed to be also good for the original input. Depending on whether the fitting problem is L_1 fitting or L_∞ fitting, the coreset is L_1 coreset or L_∞ coreset, respectively.

One way to obtain small coreset is via non-uniform sampling, using the framework by [30]. Given a point set P, the "importance" of each point $p \in P$ is quantified by

its sensitivity $\sigma_P(p)$, and the total sensitivity of P is the summation of sensitivities at every point, $\sum_{p\in P}\sigma_P(p)$. It is shown that if one samples the point set P according to the probability distribution imposed by the sensitivities, one obtains coresets of size roughly $O(\mathfrak{S}_P^2)$.

Total sensitivity of a shape fitting problem quantifies the complexity of the shapes, which is the main object being studied in this thesis. We briefly summarize the main results below.

We establish the connection between L_∞ coreset and L_1 coreset. In particular, we show that shape fitting problems with small L_∞ coreset also have small L_1 coreset. This connection allows us to use existing work on L_∞ coreset to obtain small L_1 coreset for the aforementioned shape fitting problems (variants of (j,k) projective clustering, and circle fitting). Consequently, we obtain the first near-linear algorithm for integer (j,k) projective clustering in high dimension.

We show that the total sensitivity of shape fitting problem in $\mathbb{R}^d$ depends on the intrinsic dimension of the shapes. For many shape fitting problems, the shapes are low-dimensional: for example, in (j,k) projective clustering, each shape is a union of k j-flats, and each k-tuple of j-flats is contained in a subspace of dimension $O(jk)$. This fact allows us to get a dimension-reduction type result for the (j,k)-projective clustering problems. Specifically, for integer (j,k) projective clustering, the upper bounds of the total sensitivity is improved from $O((\log n)^{f(d,j,k)})$ to $O((\log n)^{f(j,k)})$, where $f(j,k)$ is a function depending on only j, and k, and no longer on the possibly large d.

We obtain coreset of size $O((\log n)^2)$, using the connection between L_∞ coreset and L_1 coreset. We show that circle fitting problem does not admit coreset of size $o(\log n)$. In particular, we show a construction of a point set, such that any $1/100$-coreset of P has size at least $\Omega(\log n)$.

PUBLIC ABSTRACT

In this document, we study coresets for shape fitting problems. Shape fitting problems include various optimization problems people encounter in machine learning, computer vision, image processing, computational metrology, etc. Usually for such problems, either exact algorithms are not known to exist, or are computationally expensive. The idea of coreset is to obtain a small subset – so called “succinct presentation” – of the original input, which faithfully captures all the characteristics of the input, and then solve the same optimization problem with the smaller input (coreset).

Depending on how one quantifies how well a shape approximates the input point set, there are L_∞ and L_1 shape fitting problems. Coresets for L_∞ shape fitting problems have been proven to be very successful and influential in obtaining fast approximation algorithms for a wide variety of geometric approximation problems. Inspired by that, we study coresets for L_1 shape fitting problems.

We obtain coresets for shape fitting problems such as k-clustering and subspace approximation (from machine learning), k-line fitting (from computer vision), and a more general problem known as (j, k) projective clustering. In addition, for a problem from computational metrology, circle fitting problem, we obtain both small coreset for this problem, and we also show a lower bound on the size of the coreset. These results on coresets allows us to obtain fast approximation algorithms for the corresponding shape fitting problems.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1

INTRODUCTION

In this document, we describe approximation algorithms for shape fitting problems via coresets. If the shapes we have at hand is the set of lines, for example, the shape fitting problem (which is line fitting in this case) can be stated as follows: given a point set P, how closely does the point set look like a line? Similarly, if the shapes are circles on the plane, the shape fitting problem is essentially asking how closely the input point set looks like a circle. Many geometric optimization problems can be easily stated as shape fitting problems. For instance, the goal of k-median [3]/k-means clustering [2] is to find the best k "centers", such that the overall cost of assigning points to the center is minimized; this is precisely the same as finding a k-point set such that the input point set "looks like" the k points the most (see Figure 1.1). For another example, consider the classical linear regression problem [5], where the goal is to find the best coefficients to explain data: it can be considered as a shape fitting problem, where the shapes are linear subspaces (such as lines passing through origin, hyper-planes passing through origin, etc.), as shown in Figure 1.2.

Formally, a shape fitting problem is specified by a triple $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$, where $\mathbb{R}^d$ is the d-dimensional Euclidean space, $\mathcal{F}$ is a family of shapes in $\mathbb{R}^d$, and $\mathrm{dist} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ is a continuous function that we will refer to as a *distance* function. We also assume that (a) $\mathrm{dist}(p, q) = 0$ if and only if $p = q$, and (b) $\mathrm{dist}(p, q) = \mathrm{dist}(q, p)$.

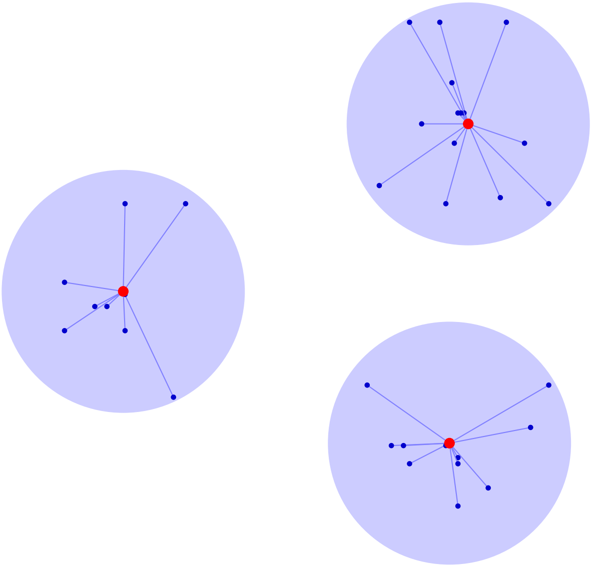


Figure 1.1. k-median/k-means clustering as a shape fitting problem (3-median/3-means clustering): how closely does the input point set “look like” a 3-point set?

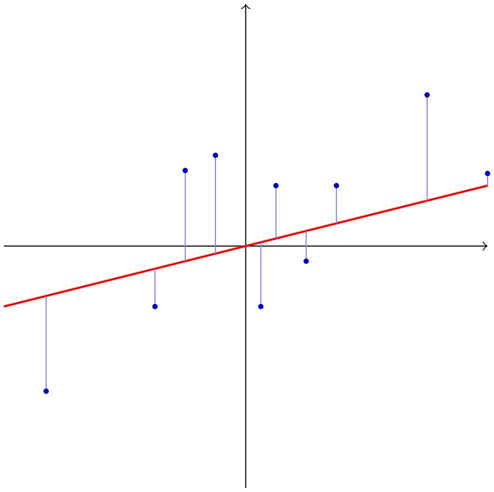


Figure 1.2. Linear regression as a shape fitting problem: how closely does the observed data “look like” a line?

It is easy to verify that these two properties are satisfied by Euclidean distance and the z^{th} power of Euclidean distance for $z \in [1, \infty)$. Euclidean distance and powers of Euclidean distance are the two distance functions we work with through out this document. We refer to each $F \in \mathcal{F}$ as a *shape* (for example, a line, a circle, a plane, a shape formed by the union of k lines/k planes, for a given parameter k, etc.), and we require each shape F to be a non-empty, closed, subset of $\mathbb{R}^d$. We define the *distance* of a point $p \in \mathbb{R}^d$ to a shape $F \in \mathcal{F}$ to be $\operatorname{dist}(p, F) = \min_{q \in F} \operatorname{dist}(p, q)$. An instance of a shape fitting problem is specified by a finite point set $P \subset \mathbb{R}^d$. There are two possible ways to quantify how well a shape F fits the point set P: one is $\sum_{p \in P} \operatorname{dist}(p, F)$, the other is $\max_{p \in P} \operatorname{dist}(p, F)$. In both cases, the goal is to find a shape which best fits P, that is, a shape minimizing $\sum_{p \in P} \operatorname{dist}(p, F)$ over all shapes $F \in \mathcal{F}$ in the first case, or a shape minimizing $\max_{p \in P} \operatorname{dist}(p, F)$ in the second case. We slightly abuse notation, and use $\operatorname{dist}(P, F)$ to denote $\sum_{p \in P} \operatorname{dist}(p, F)$ for L_1 shape fitting problem, and $\max_{p \in P} \operatorname{dist}(p, F)$ for L_∞ shape fitting problem. More explicitly, we have the following L_1 and L_∞ shape fitting problems:

Problem 1 (Shape fitting problem, L_1 fitting)**.** Given an instance P for a shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \operatorname{dist})$, find

$$F = \arg\min_{F \in \mathcal{F}} \sum_{p \in P} \operatorname{dist}(p, F).$$

Problem 2 (Shape fitting problem, L_∞ fitting)**.** Given an instance P for a shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \operatorname{dist})$, find

$$F = \arg\min_{F \in \mathcal{F}} \max_{p \in P} \operatorname{dist}(p, F).$$

These two different fitting criteria arise naturally in practice: if we consider each shape as the location of a center (or a facility) serving the clients (each point denotes the location of a client), and $\mathrm{dist}(p, F)$ is the Euclidean distance between a client p and the facility F, then the goal of L_1 shape fitting is to find the most economical placement of the facility, so that the overall distances from the clients to the facility is not too large; while the goal of L_∞ shape fitting is to find the "fairest" placement, such that each client is guaranteed to be within some reasonable distance to the facility.

We briefly explain the reason that we call the first problem L_1 shape fitting problem and the second problem L_∞ fitting: given an n-point set $P = \{p_i | 1 \leq i \leq n\}$, and a shape F, the vector

$$\begin{bmatrix} \mathrm{dist}(p_1, F) & \mathrm{dist}(p_2, F) & \cdots & \mathrm{dist}(p_n, F) \end{bmatrix}$$

encodes the distance from each point to the shape. For L_1 shape fitting problem the objective function being minimized is the L_1 norm of this vector; while for L_∞ shape fitting problem, the objective function being minimized is the L_∞ norm of this vector.

An important shape fitting problem is the (j, k) *projective clustering problem*. Each shapes considered in (j, k) projective clustering problem is a k-tuple of j-flats (affine j-subspaces). For example, if $j = 1$, 1-flats are lines, so each shape in $(1, k)$ projective clustering is formed by union of k lines (see Figure 1.3); if $j = 2$, 2-flats are planes, so each shape is formed by union of k planes.

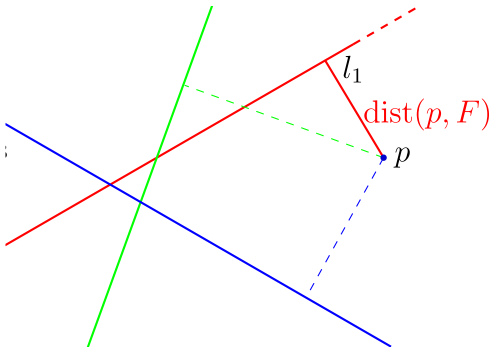


Figure 1.3. A shape in $(1, 3)$ projective clustering in $\mathbb{R}^2$, formed by the union of three lines, l_1, l_2 and l_3. The distance from p to the shape $l_1 \cup l_2 \cup l_3$ is the distance from p to the nearest line, which is l_1 in this case.

Many clustering problems are special cases of (j, k) projective clustering. We list below the special cases of (j, k) projective clustering problems that will be considered in the rest of the documents.

$(0, k)$ **projective clustering—k-clustering problems, including k-median/k-means:** When $j = 0$, $\mathcal{F}$ is the set of k-point sets of $\mathbb{R}^d$, so the $(0, k)$-projective clustering problem is the k-median clustering problem when the distance function is the Euclidean distance, and it is the k-means clustering problem when the distance function is the square of the Euclidean distance.

$(1, k)$ **projective clustering—k-line clustering:** when $j = 1$, the family of shapes is the set of k-tuples of lines in $\mathbb{R}^d$.

$(j, 1)$ **projective clustering— affine j-subspace approximation:** when $k = 1$, the family of shapes is the set of j-flats (affine j-subspaces). One particularly important problem is j-subspace approximation (which is also called low rank approximation in literature), where the family of shapes is the set of j-subspaces. We do not specifically emphasize the difference here, since constructions of coresets for these two problems are very similar, as a j-flat is contained in a $(j+1)$-subspace. Often once one has a construction for coreset for fitting j-flat, one also easily obtains a coreset construction for fitting j-subspace.

integer (j,k) **projective clustering:** Other than the above projective clustering problems where j or k is set to specific values, another variant of the (j,k) projective clustering problem is the integer (j,k)-projective clustering problem, where we assume that the input points have integer coordinates (but there is no restriction on j and k), and the magnitude of these coordinates is at most n^c, where n is the number of input points and $c > 0$ is some constant. That is, the points are in a polynomially large integer grid.

Other than (j,k) projective clustering, another shape fitting problem studied in this document is *circle fitting problem on the plane*: the family of shapes consists of all circles, $\mathcal{F} = \{F_{a,b,r} | a, b \in \mathbb{R}, r \geq 0\}$, where $F_{a,b,r}$ is the circle centered at (a,b), with radius r. The distance from a point $p = (x,y)$ to $F_{a,b,r}$ is $\left|\sqrt{(x-a)^2 + (y-b)^2} - r\right|$ (so it is distance from p to the nearest point on the circle). See Figure 1.4 for an example.

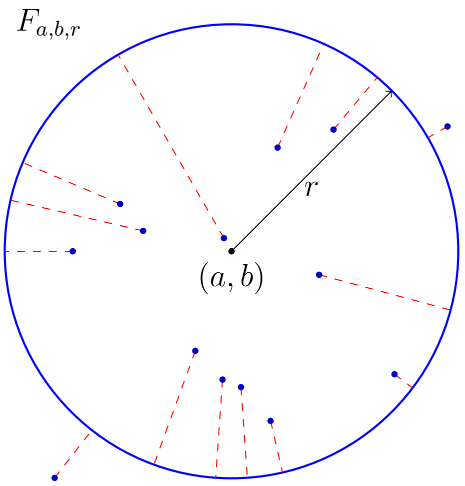


Figure 1.4. Circle fitting problem on the plane. Each point is assigned to the nearest point on the circle. The overall cost is the summation of distance from each point to the circle.

1.1 Coresets for shape fitting problems

In this section, we first introduce one of the key object in this document—coreset of a shape fitting problem; after that we describe the framework of approximation algorithms for shape fitting problems using coreset.

A coreset $S \subset \mathbb{R}^d$ of P is a weighted point set in $\mathbb{R}^d$, together with a weight function $w : S \to \mathbb{R}^+$, which assigns a weight $w(p)$ for each point $p \in S$. A coreset is a *compact representation* of the input point set P with respect to the shapes in $\mathcal{F}$. Informally, S being a "representation" (or sketch, or summary, or digest) of P (in the context of shape fitting problem) means for any shape $F \in \mathcal{F}$, the summation of the distances from weighted points in S to F approximates the summation of distances from points in P to F. So S exhibits all characteristic of P, in the sense that from the

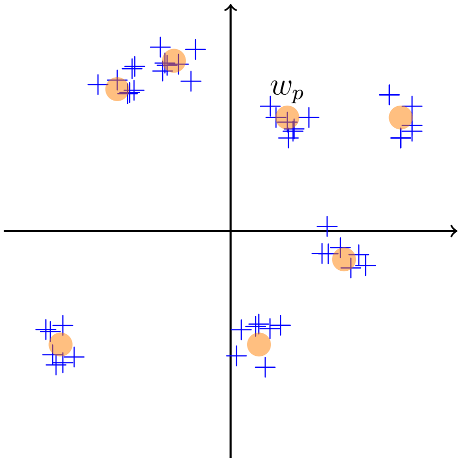


Figure 1.5. What a coreset might look like for line fitting problem. The blue crosses are points in P, and the orange points are points in S, together with weights.

view point of any shape $F \in \mathcal{F}$, S is almost indistinguishable from P, since S and P contributes roughly the same cost when points in S or P are assigned to F. The *size* of the coreset S is $|S|$ (the number of points in S). It is usually asymptotically smaller than the number of input points $|P|$, and that is the reason we call S a "compact" (or succinct, or sparse) representation of P. Figure 1.5 shows what a coreset might look like.

Formally, L_1 coreset is defined as following:

Definition 1 (L_1 ϵ-coreset of a shape fitting problem)**.** Given an instance $P \subset \mathbb{R}^d$ of a shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$, and $\epsilon \in (0, 1]$, an ϵ-coreset of P is a (weighted) set S, together with a weight function $w : S \to \mathbb{R}^+$, such that for any shape F in $\mathcal{F}$, it holds that

$$|\mathrm{cost}(P, F) - \mathrm{cost}(S, F)| \leq \epsilon \mathrm{cost}(P, F),$$

where by definition,

$$\mathrm{cost}(P, F) := \sum_{p \in P} \mathrm{dist}(p, F), \qquad \mathrm{cost}(S, F) := \sum_{p \in S} w(p)\mathrm{dist}(p, F).$$

Coreset for L_∞ shape fitting problem is defined in a similar fashion:

Definition 2 (L_∞ δ-coreset of a shape fitting problem)**.** Given an n-point set P of a shape fitting problem, and $\delta \in [0, 1)$, a subset $Q \subset P$ is an L_∞ δ-coreset of P, if for every $F \in \mathcal{F}$, it holds that

$$\max_{q \in Q} \mathrm{dist}(q, F) \geq (1 - \delta) \max_{p \in P} \mathrm{dist}(p, F).$$

(The distance from the furthest point in Q to a shape F is at least $(1 - \delta)$ times the distance from the furthest point in P to a shape F.) Also note that since $Q \subset P$,

$$\max_{q \in Q} \mathrm{dist}(q, F) \leq \max_{p \in P} \mathrm{dist}(p, F).)$$

As we will see in Section 1.2, L_∞ coreset first appeared, and it was used to solve L_∞ shape fitting problems; later it was generalized in L_1 setting.

We now explain how to use coreset to solve the shape fitting problem approximately. The coreset S has the property that can be (informally and) roughly summarized as "a good (respectively bad) fit to S is also a good (respectively bad) fit to P". In other words, if a shape fits S well, it also fits P well. Hence it usually suffices to find a $(1 + \epsilon/c)$ approximation solution for S, where c is some constant. This solution is then a $(1+\epsilon)$ approximation for P. Now the size of S is small, so one can afford to run computational expensive algorithms (for example, exact algorithms for the optimization problem) on S. This framework is presented in Algorithm 1:

Algorithm 1: Compute an approximate solution for P via coresets

```
**Input**: An input point set $P$
**Output**: A $(1+\epsilon)$ approximate solution for $P$, for $\epsilon \in (0, 1/2)$
Compute an $\epsilon/4$-coreset $S$ for $P$ ;
Compute a $(1+\epsilon/3)$ approximate solution $F$, such that
$\mathrm{dist}(S, F) \leq (1+\epsilon/3) \min_{F \in \mathcal{F}} \mathrm{dist}(S, F)$;
Output $F$.
```

We make two remarks regarding definition of coreset.

Remark 1 (Merging property of coreset)**.** Coreset can be merged. Given m point sets $P_1, \cdots, P_m$, suppose S_i is an ϵ-coreset for P_i, then $\cup_{i=1}^m S_i$ is an ϵ-coreset for $\cup_{i=1}^m P_i$ (note that the union operation here is "bag union", where multiple identical elements are counted with the multiplicity or weights). This property can be quite useful in practice: for example, suppose data is collected by m machines $M_1, \cdots, M_m$, and we are interested in performing some computation, for example, subspace approximation, or clustering, on the whole data collected by these machines. Instead of first letting each machine send their portion of data to a central machine, and then solving the optimization problem on the union of the data, one approach could be to first let each machine compute a coreset of the data collected on that machine, and then each machine only need to send its coreset to the central machine. The final computation is performed on the union of coresets. In this scenario, coreset is used as a tool to break a large computational problem to several small problems.

Remark 2 (Intepretability of coreset)**.** Another thing to note is that in this definition, we do not require S to be a subset of P, and we also allow the weights of points in S, $w(p)$ for $p \in S$ to be negative. Indeed, this definition reflects the fact that small coreset is a compact and sparse approximation of P with respect to the family of shapes considered in the shape fitting problem at hand. However, in this work, we will mainly focus on constructing coresets which have these additional properties.

Coreset with these two additional properties has theoretical appeal, and there are also two major practical motivations. The first reason is interpretability of the weighted point set S. In data analysis application, a coreset S being a subset of P is easier to interpret than a set of points which are sometimes linear combinations of points in the original point set P. This is in particular true in situations where each point in P has a natural meaning. For example, in text analysis, each document is represented by a point/vector in $\mathbb{R}^N$, where N is the number of indexing words (*e.g.* words in a dictionary), where each entry in the vector indicates whether a particular word appears in the document or not (so 1 in the i^{th} entry denotes the i^{th} word appeared in the document and 0 otherwise). A linear combinations of several documents is no longer a document in the corpora most of the time. The second issue is regarding the computations to be performed on the coreset. In the framework presented in Algorithm 1, after computing the coreset S of P, the next step is to solve the optimization problem on the coreset S using off-shelf algorithms. These algorithms can be (possibly computationally expensive) exact algorithms, or some heuristics. However, optimization problems with negative weights are generally harder to solve, since negative weights sometimes turn a convex optimization problem into non-convex optimization problem, and many heuristics may no longer be applicable in this case.

1.2 Related work: a brief history of coreset

In this section, we discuss the development of coresets for (j, k) projective clustering problem and circle fitting problem. Since there are abundance of results on coresets, in order to make the discussion clear, we first give a general overview of this area and remark on some extremely influential papers. After that, we provide a mosaic summarizing sizes of coresets for all the shape fitting problems studied in this document (our results are also included in Table 1.1, marked with a green color). Lastly, we mention a few shape fitting problems that do not admit small coreset.

An early influential work on coreset is by Agarwal, Har-peled and Varadarajan in the seminal work [6][7]. The problems they studied are L_∞ shape fitting problems, which are mostly referred as enclosing problems. For example, finding the best point fitting the input point set P in L_∞ sense, is the same as finding the minimum enclosing ball containing P; finding the best hyperplane fitting P in L_∞ sense, is the same as finding the minimum width slab (two parallel hyper-planes) containing P; finding the best circle fitting P, is equivalent to finding the minimum width annulus containing P. Agarwal, Har-peled and Varadarajan showed that this kind of L_∞ fitting problems, can be reduced to one particular problem: computing extent of the input point set. The extent of a point set P along a given direction is the width of the minimum slab orthogonal to the direction that encloses P. They introduced the notion of ϵ-kernel, which is a subset Q of P such that the extent of Q is at least $1-\epsilon$ times the extent of P along every direction. They also given constructions of small ϵ-kernel, which is used to compute small ϵ-coreset for a variety of L_∞ shape fitting problems.

Their work is quite influential for several reasons. The notion of ϵ-kernel is related to an approximate version of convex hulls in the strongest (containmentwise) sense. Although definitions of approximating convex hull exist in literature, they were weaker and less clean. The framework of geometric approximation via (L_∞) coreset addresses many L_∞ shape fitting problems. To some extent, the research on L_1 coreset is inspired by the success of coreset for L_∞ fitting problems.

We now turn to L_1 coresets. There are two important quantities here: the cardinality of P, which is the number of input points, and the dimension d, which is the dimension of the Euclidean space in which the shape fitting problem is considered. We will point out the dependence of the size of the coreset on n and d explicitly, as these are the two most important factors in the size of coreset.

Roughly speaking, the evolution of the size of the coreset can be summarized as following: from "polylogarithmic dependence on n, and exponential dependence on d" to "polylogarithmic dependence on n, and polynomial dependence on d", then to "independent of n, and exponential dependence on d", then to "independent of n, and polynomial dependence on d".

We now briefly overview the algorithms of computing coreset. An early prevailing framework of computing coresets is low variance sampling. A common modus operandi performs in two steps. The first step is to find a relatively good shape for the input point set (using off-shelf approximation algorithms, such as constant factor approximation algorithms), and partition the point set into multiple regions based on this shape. For each region one sets up a carefully designed probability distribution (the probability distribution is often tailored to ensure low variance of the sampling). In the second step, points from each region are randomly sampled, and each sampled point is also assigned a weight. The final output is the weighted samples S. S has the property that for an arbitrary fixed shape F, $\text{cost}(S, F)$ approximates $\text{cost}(P, F)$ with high probability. The rest of the analysis to determine the number of samples to be drawn from each region is often very problem-specific; and the $\log n$ factor appears often as a result of union bound: since the coreset has to approximate the input point set with respect to *every* shape in the family of shapes (which consists of infinitely many shapes), the routine approach is to find a set of representative shapes and apply a union bound to ensure that with constant probability, S approximates P on this set of representative shapes. This particular set of representative shapes is selected in certain way, such that if S is a good approximation of P with respect to each of them, then S approximates P with respect to every shape with at least constant probability. If the number of representative shapes is polynomial in n (the number of input points in P), there would be a $\log n$ factor in the size of the coreset S because of union bound. It worth noting that all these algorithms are randomized and they succeed with a constant probability. The dependence on d in these coreset first appeared as exponential, and later was improved to be polynomial.

Several constructions of some other weaker counterparts of coresets (sometimes referred as weak coreset) of size independent of n and d were also designed. For instance, Feldman, Monemizadeh and Sohler [20] showed a weak coreset for k-means clustering, whose size only depends on k. The weak coreset only approximates the input point set on a certain set of shapes instead all the shapes. Such work influenced later research, especially regarding the analysis of the coreset: some careful and delicate arguments using conditional probability took place of the union bound, so the number of the representative shapes is sometimes smaller. (In [20], the authors showed that one only need to consider a constant number of representative shapes.)

An important piece of work regarding the low variance sampling scheme that was ubiquitously used in the construction of coresets is the paper of Langberg and Schulman [30]. They introduced the notion of sensitivity and total total sensitivity of shape fitting problems: given a shape fitting problem, and the input point set P, the sensitivity $\sigma_P(p)$ of each point in P quantifies the importance of the point (among other points), and total sensitivity, being a summation of sensitivities of each point, $\sum_{p\in P}\sigma_P(p)$, quantifies the complexity of the shapes. There is a natural probability distribution induced by sensitivities of each points: the probability of each point being picked is proportional to its "importance", which is $\sigma_P(p)/\sum_{p\in P}\sigma_P(p)$. Compared with previous low variance sampling scheme, this "sampling-by-importance" is much simpler and natural. One of the significance of this work is that it started considering shape fitting problems in a more unified and general way. Instead of tailor-making coreset for each shape fitting problem, this paper encapsulates the complexity of shapes in total sensitivity, and provides a unified framework for coreset construction. It also connects the complexity of a shape fitting problem (total sensitivity) with the size of the coreset: the size of the coreset roughly quadratically depends on the total sensitivity. Accompanied with a careful analysis of the weighted sample, using a double-sampling argument, similar to the fundamental ϵ-net theorem [31], first coreset of size independent of n for k-median/k-means was obtained.

The next important work along the line of unification of the framework for computing coreset is by Feldman and Langberg [19]. It connects ϵ-coreset with the well-studied ϵ-approximation of range spaces. They showed that the problem of computing coreset can be reduced to the problem of computing ϵ-approximation [31] of a range space induced by the shape fitting problem. Therefore, the routine and problem-tailored analysis (to show that the weighted sample is a coreset) in most of the constructions of coreset is removed. Using the ϵ-approximation theorem as a common ground, the coreset size is also improved. This is also the first deterministic construction of coreset, since there is deterministic construction of ϵ-approximation.

The connection between coreset and ϵ-approximation of range spaces is via sensitivity and total sensitivity. The size of the coreset obtained via this framework is a function of total sensitivity and another parameter measuring the complexity of the shape fitting problem (this parameter depends on d).

We make a remark on the expression “small coreset”, since this expression will appear many times in the following chapters. As we have seen, the dependence of the size of the coreset on d can be either exponential or polynomial. In fixed dimension, d is considered a constant, therefore, the size of small coreset could depend exponentially on d; in high dimension, d is considered as an input, therefore, the size of the small coreset can at most depend polynomially on d.

Table 1.1 summarizes most of the work on coreset so far. Our results are also included in Table 1.1. The result with an “$*$” indicates that it is not exactly a coreset, instead it is some kind of succinct sketch similar to but weaker than coreset. In order to show more clearly the dependence of the coreset size on n, d, and j, the distance functions here are all Euclidean distance.

We now review some results about the lower bounds of coreset. In [26], it is shown that the L_∞ shape fitting problem of fitting an n-point set in $\mathbb{R}^3$ with two planes does not admit coreset of size $o(n)$. The point set constructed there also does not admit small L_1 coreset (the way to show this is very similar to the method used in

Shape fitting problem	Coreset size	Reference
k-median clustering	$O(2^{O(1/\epsilon^4)} \log n)^*$	[9]
k-median clustering	$O(k^3 \epsilon^{-d-1})$	[29]
k-means clustering c	$O(k \ln k \epsilon^{-2} \ln(k/\epsilon))$	[20]
k-median/k-means clustering	$O(dk^2 \epsilon^{-2} \log n \log(k/\epsilon))$	[13]
k-median/k-means clustering	$\tilde{O}(d^2 k^3 \epsilon^{-2})$	[30]
k-median clustering	$O(kd\epsilon^{-2})$	[19]
k-median/k-means clustering	$O(k^2 d\epsilon^{-2})$	[39]
k-median with outlier, k-median with weighted center	$O(d(\log n)^2 \epsilon^{-2})$	[23]
k-line clustering	$O((\log n)^{O(k)} (1/\epsilon)^{O(d \log d + k)})$	[18]
k-line clustering	$O((\log n)/\epsilon)^{O(k)} + O(dk\epsilon^{-2})$	[19]
k-line clustering	$O((\log n)^{f(k,d)} \epsilon^{-2})$	[38]
k-line clustering	$O(k^{O(k)} d(\log n)^2 \epsilon^{-2})$	[39]
j-subspace clustering	$\tilde{O}(j^4 \epsilon^{-2})^*$	[16]
j-subspace clustering	$O((1/\epsilon)^{\mathrm{poly}(j,d)} (\log n)^{O(j^2)})$	[18]
j-subspace clustering	$O(dj^{O(j^2)} \epsilon^{-2} \log n)$	[21]
j-subspace clustering	$O(dj\epsilon^{-2})$	[19]
j-subspace clustering	$O(j^4 d\epsilon^{-2})$	[39]
integer (j,k) projective clustering	$O((\log n)^{f(d,j,k)} \epsilon^{-2})$	[38]
integer (j,k) projective clustering	$O((\log n)^{f(j,k)} kjd\epsilon^{-2})$	[39]
circle fitting on the plane	$O(\mathrm{poly}(\log n, 1/\epsilon))^*$	[28]
circle fitting on the plane	$O((\log n)^2 \epsilon^{-2})$	[38]

Table 1.1. Results on coresets for the shape fitting problems.

Chapter 7, where we show the lower bound on L_1 coreset for circle fitting problem). In [27], the author shows that fitting weighted point set with 2 lines, and that fitting a point set by k-lines, where $k \geq \log n$, also do not have coreset of size independent of the cardinality of the input point set.

1.3 Our contribution so far and significance

In this section, we list the concrete contributions of this dissertation. We mainly focus on computing positively weighted coreset which is also a subset of the original input point set for variants of the (j, k)-projective clustering problem and circle fitting problem.

1.3.1 Connection between L_∞ coreset and total sensitivity

We connect L_∞ coreset and total sensitivity. In particular, we prove that shape fitting problems with small L_∞ coresets also admit L_1 coreset. Using this connection, we obtain the first small coreset for circle fitting problem. This result is quite interesting because compared with other shape fitting problems such as k-median/k-means clustering and j-subspace approximation, the amount of results on circle fitting is much fewer. We give the first near-linear algorithm for integer (j, k) projective clustering in high dimensions. Another way of stating our result is that we have a near-linear approximation for the general (not integer) (j, k) projective clustering problem, provided the optimal fit is only polynomially smaller than the diameter of the input point set. These results are in Chapter 4.

1.3.2 Total sensitivity depends on the intrinsic dimension of the shapes

We show that the total sensitivity of a shape fitting problem depends on the ambient dimension of the shape. For example, consider the (j, k) projective clustering. A shape here is the union of k j-flats, which is contained in a low-dimensional subspace of dimension $O(jk)$. For a point set in $\mathbb{R}^d$, the total sensitivity of the point set in this case depends on j and k, instead of d. Using this result, we improve the bound of the total sensitivities for variants of (j, k) projective clustering problems. In particular, we remove the exponential dependence on d of the total sensitivity for (j, k) projective clustering, and also greatly simplify earlier work on upper bounding the total sensitivities for k-clustering problems. The main idea that the total sensitivity depends on the output (the best shape) rather than on the input (the point set) reflects the deep fact that total sensitivity captures the notion of complexity of the family of shapes. This result shed light on new understanding on total sensitivity of shape fitting problems, which may inspire future work along the same lines. The coreset we derive is positively weighted, and is a subset of input point set. As we have remarked before, there are several advantages of obtaining positively weighted coreset. For example, we may run algorithms or heuristics developed for the shape fitting problem on the coreset to get an approximate solution to the shape fitting problem. These results are in Chapter 5.

1.3.3 Lower bound on the size of coreset for circle fitting problem

In Chapter 7, we show two results. The main result is that circle fitting does not admit coreset of size $o(\log n)$, where n is the cardinality of input point set. We construct a point set, whose coreset has size $\Omega(\log n)$. We also show that the total sensitivity of circle fitting problem is $\Omega(\log n)$, this implies that the upper bound of the total sensitivity of circle fitting we obtained in Chapter 4 is tight.

1.4 Organization of the document

We motivates the study of core-set and include a few application of coreset in Chapter 2. After that, in Chapter 3, we introduce the concept of sensitivity of a point in a point set for a shape fitting problem, and total sensitivity of a shape fitting problem. We briefly explain the connection between total sensitivity of a shape fitting problem, and coreset. In Chapter 4, we first show a connection between L_∞ coreset and total sensitivity, and use this connection to derive upper bounds of total sensitivity for circle fitting, k-line clustering, and integer (j, k) projective clustering problems. We obtain small coreset in fixed dimension using the upper bounds of total sensitivities for these three shape fitting problems. In Chapter 5, we show the dimension reduction argument, which shows that the total sensitivity of shape fitting problems depends on the ambient dimension of the shapes, instead of the dimension d, for which $P \subset \mathbb{R}^d$. We obtain small coreset for several variant of (j, k) projective clustering problems in high dimension. In Chapter 6, we review the results from [19], which show the connection between ϵ-approximation and ϵ-coreset, which completes the framework of using sensitivities/total sensitivity to derive coreset. In Chapter 7, we show that the lower bound of coreset for circle fitting problem is $\Omega(\log n)$ and that the total sensitivity for circle fitting we derived earlier is tight. We end this dissertation with a few open problems in Chapter 8.

CHAPTER 2

APPLICATION SCENARIO

In this chapter, we describe several applications (j, k) projective clustering. We first discuss some applications of $(j, 1)$-projective clustering, which is j-flat approximation, the we discuss the general (j, k) projective clustering. In many applications, the matrix is a natural structure to encode data: each column corresponds to an object (a gene, a document, an image, etc.), which is described by a list of features (expression levels under certain condition, frequency of words appeared in the document, grayscale of each pixel in an image, etc.). For example, in information retrieval, the information of documents in a corpora is usually encoded in a term-document matrix M, where the entry M_{ij} indicates the frequency of the i^{th} key word in the j^{th} document; in computer vision, each image is encode by an n-vector recording the grayscale of the pixels (so an 100×100 pixel image is encoded as a vector of length 10^4), and a collection of m images is represented by an $n \times m$ matrix; in genetics, a people-by-gene matrix encodes information about the response of the j^{th} gene in the i^{th} individual/condition (so each row corresponds to a person, and each column corresponds to a gene).

Subspace approximation ($(j, 1)$ projective clustering) is particularly popular in the field of computer vision, information retrieval, bioinformatics, genetics, etc. One classical use of subspace approximation is for data interpretation: in order to understand certain phenomenon, one take a large number N of measurements, each observation is thus a point in the ambient space $\mathbb{R}^N$. Low-dimensional subspace often serves as a tool to reveal the sometimes hidden, simplified structures underlying the complex data. Subspace approximation is also often used as a pre-processing step before clustering: the reason is often referred as "curse of dimensionality" [1]—essentially every point looks like an outlier in high dimensional space, therefore, cluster/outlier are no longer meaningful. In these situations, one often projects the data to a low dimensional space and analyzes the low-dimensional points, for example, see [41]. Subspace approximation is also used in information retrieval, where it is used as a tool to overcome the problem of synonymy and polysemy in information retrieval. We elaborate this application in the following section.

Clearly, there are also computational gains by using subspace approximation: after projecting the high dimensional data onto a low dimensional subspace, many computations is faster to perform in the low dimensional space (for example, distances between pairs of points, nearest neighbors of a point, etc.). Subspace approximation also helps to reduce the storage space of the data: the rank of an object-feature matrix might quite large (due to noise, or the way data is collected, etc.), however, it is often observed that the intrinsic dimension is much smaller. For instance, in pattern recognition (face recognition, or hand-written digit recognition), visual data often exhibits low-dimensional structure due to rich local regularities (objects in images often appear as color blocks), global symmetries (human face, for example, when viewed from front, is always roughly symmetric). As another example, in information retrieval, the matrix for encoding a well-structured corpora (*e,g.* an encyclopedia) is usually quite large, containing thousands of rows and thousands of columns. For example, the well-studied MEDLINE collection of medical abstracts is a 5526 × 1033 term-by-document matrix. However, it is observed that a subspace of rank in the order of hundreds captures most of the information [15].

Below we describe two applications of projective clustering: the first one is an application of subspace approximation. It is from information retrieval, which is the influential latent semantic indexing (LSI) [15][4]. The second one is an application of general (j, k)-projective clustering. It is from computer vision, in particular,motion segmentation, which classifies moving objects in video sequence [40]. There are, numerous applications of projective clustering, in particular, subspace approximation, in many other fields, such as astronomy [35][12][11], genetics [34][33][32].

2.1 Retrieving relevant documents: latent semantic indexing

A classical problem in information retrieval is automatic indexing and retrieval. Data is modeled as a matrix (the term-document matrix), and a user's query of the collection of documents is represented as a vector. Given a query, the relevance of a document to the query is measured by the angle between the vector of the query, and the vector of the document. The query matching was a method quite intensively studied before the appearance of latent semantic indexing: it essentially uses a “term-overlapping” methods, which given a query, retrieves all the documents in the collection which contains the word in user query. However, this method suffers from two problems: *synonymy* and *polysemy*. Synonymy describes the fact that there are many ways to refer to the same object (for example, “cat” and “feline”, or “heart attack” and “myocardial infarctios”). Polysemy describes the fact that most words have more than one disctinct meaning (for example, the word “cone” clearly means different things when it appears as “ice cream cone” and “pine cone”, the word “chip”, “bank” generally have many different meanings). Synonymy appears as a challenging problem in information retrieval, as two people choose the same main key word for a single well-known object less than 20% of the time [24]. This problem seems even more difficult to solve, considering that documents themselves do not contain all the terms user might try to look it up under.

The idea of latent semantic indexing (LSI) is designed to overcome the synonymy problem based on the idea of retrieving document by *conceptual content/conceptual topic*, instead of the unreliable individual words. The key idea is to treat the unreliability of the observed term-document association data as a statistical problem, since the observed term-document is only one representative of a whole family of relatively close matrices representing the same corpora. For example, people might choose different words to index the documents on the same topic (thus leading to different possible representations of the term-document matrix). However, they are all valid representations of the same concept/meaning. Low rank subspace is used to capture the major associative patterns in the data, and ignore the less important influence. As a simple example, consider the association of a pair of words “access” and “retrieval”. If “access” and “retrieval” appear together in most of the documents in the corpora, say, 95% of the time, then the absence of “retrieval” from a document containing “access” might be “erroneous”, in the sense that it should probably contain “retrieval”, but because of different wording the word“retrieval” does not appear. Based on the fact that meaningful documents naturally display correlation of the occurrence of one term and another, a subspace serves as the correct tool to captures this kind of major association and to remove the noise (introduced by synonymy) of the data.

More explicitly, LSI works as follows: given the input term-document matrix, one seeks the “sweet spot” of the dimension of the subspace. If the rank of the subspace is too small, the approximation of the original term-document might not be a sufficiently accurate approximation, and thus might not capture enough information of the term-document matrix. On the other hand, the rank of the subspace should not be too large, since if one reconstructs the original matrix too “precise”, one also begin to capture noise (or “uncertain”). The correct rank of the subspace is chosen by experiments: by varying different values of the rank, one observes how well information retrieval system using the j-subspace works, and chooses the best j, which returns most of the relevant documents and does not return too many irrelevant documents (keeping the number of returned irrelevant to a query small is important, since trivially one could always return all the documents to a query, but presenting the user all documents is meaningless). Once the rank of the subspace is determined, all the rest of computations are performed within this subspace: in particular, each document is replaced by its projection on the subspace, and each query is also first projected onto the subspace, and then the relevance of each documents to the query is computed. Usually only documents whose relevance exceeds certain threshold are returned.

2.2 Clustering trajectories: using projective clustering in motion segmentation

We describe an application of projective clustering in computer vision: motion segmentation. It is used as a pre-processing step for surveillance, tracking, and action recognition [37]. The input of the motion segmentation problem is a set of trajectories. Each trajectory records the position (or coordinates) of a point appeared on the scene (through a camera) at F frames, so the spatiotemporal information for the point is encoded in a vector of length $2F$ (because for one position we need to record the x-coordinate and the y-coordinate). For a set of n moving points, a video sequence of F frames recording the trajectories of these points is a $2F \times n$ matrix, where each column correspond to a single point, and consecutive 2-row correspond to a snapshot of the scene. Suppose m of the trajectories correspond to the same rigid-body motion, the m columns will live on a 3-dimensional flat according to the affine model [36][42][40].

For a quick example, consider the following matrix:

$$\begin{array}{c|ccc|} & p_1 & p_2 & p_3 \\ F_1 & 1 & 1 & 3 \\ & 1 & 2 & 7 \\ F_2 & 8 & 3 & 5 \\ & 4 & 3 & 1 \\ F_3 & 9 & 1 & 6 \\ & 1 & 2 & 3 \end{array}.$$

This matrix contains three trajectories, and the video sequence contains three frames. The first two rows of the matrix encodes the position of the three points at frame F_1: the position of p_1 is $(1, 1)$, and the position of p_2 is $(1, 2)$, and the position of p_3 is $(3, 7)$; similarly, the 3rd row and 4th row records the scene at frame F_2, with p_1 at (8,4), p_2 at (3,3), and p_3 at (5,1).

Using certain model selection methods, one can determine in advance that there are k rigid-body in the video sequence. Then the goal of the the motion segmentation problem is to cluster the trajectories into k groups, such that the trajectories in each group correspond to a rigid-body motion. Since the trajectories correspond to a rigid-motion body lie in a 3-flat, the problem is the same as solving the $(3, k)$-projective clustering problem (after finding the optimum k 3-flats, one can partition the trajectories according to the 3-flats to which they are assigned).

CHAPTER 3

SENSITIVITY AND TOTAL SENSITIVITIES: QUANTIFYING THE COMPLEXITY OF SHAPE FITTING PROBLEMS

In this chapter, we introduce the notion of sensitivity and total sensitivity. These notions are from [30]. This chapter is organized as follows: we first explain some intuition and motivation for sensitivity/total sensitivity in Section 3.1. In Section 3.2 we explain the connection between total sensitivity and the size of coreset. After that we present the algorithm for computing coreset using sensitivity and total sensitivity. This section contains all the necessary definitions for the following two chapters (Chapter 4 and Chapter 5), where upper bounds of total sensitivities for several shape fitting problems are derived.

3.1 Sensitivities and total sensitivity for a shape fitting problem

In this section, we discuss the notion of *sensitivity* of a point in a point set for a shape fitting problem and *total sensitivity* of a shape fitting problem. Sensitivity and total sensitivity of a shape fitting problem is introduced by Langberg and Schulman[30] as a tool for obtaining constant size ϵ-coreset for k-median/k-means clustering problems (and other variants, generally referred as k-clustering problem). Given an instance P of a shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$, the sensitivity of a point p in P quantifies the importance (or influence or outlierness) of p among the point set. For shape fitting problems, where we are interested in approximating the overall summation of distance from each point in P to a shape F, $\sum_{p\in P}\operatorname{dist}(p,F)$, the "importance" of a point is defined in a very natural way: the "importance" of a point p among P with respect to a shape F is simply the ratio $\operatorname{dist}(p,F)/\operatorname{dist}(P,F)$, which is the fraction p contributes to the overall cost $\operatorname{dist}(P,F)$ (See Figure **??**). Since we need to approximate $\sum_{p\in P}\operatorname{dist}(p,F)$ for every $F\in\mathcal{F}$, the "importance" of p among P (considering all shapes) is taken as the largest possible fraction that p can contribute, which is $\max_{F\in\mathcal{F}}\operatorname{dist}(p,F)/\operatorname{dist}(P,F)$.

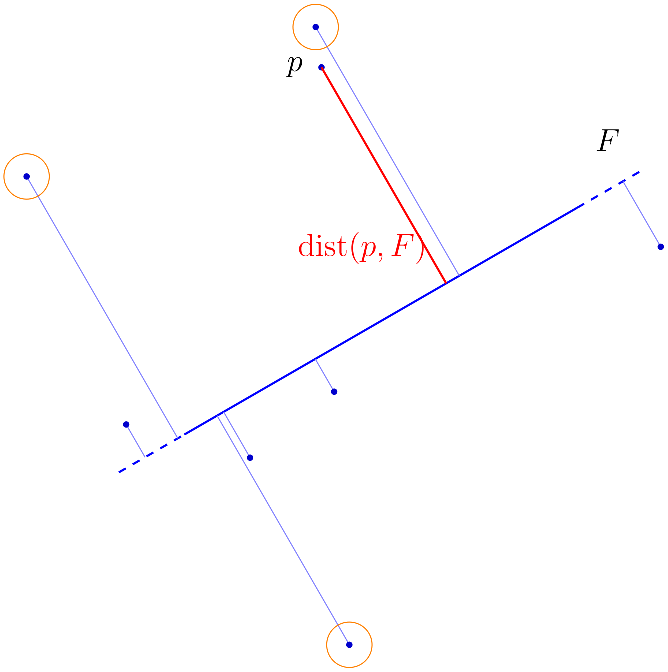


Figure 3.1. "Importance" of p among the point set with respect to a shape F is measured as $\operatorname{dist}(p,F)/\operatorname{dist}(P,F)$. For the shape F in the picture, there are at least three points which contribute more than p to $\sum_{p\in P}\operatorname{dist}(p,F)$, therefore, the "importance" of p is at most 1/4 for this particular F.

Formally, the sensitivity of a point in a point set (with respect to a family of shapes) is defined as follows:

Definition 3 (Sensitivity of a shape fitting problem)**.** Given an instance $P \subset \mathbb{R}^d$ of a shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$, the sensitivity of a point p in P is

$$\sigma_P(p) := \inf\{\beta \geq 0 | \mathrm{dist}(p, F) \leq \beta \mathrm{dist}(P, F), \forall F \in \mathcal{F}\}.$$

This definition is equivalent to let $\sigma_P(p) = \sup_{F \in \mathcal{F}} \mathrm{dist}(p, F)/\mathrm{dist}(P, F)$, with the understanding that if the denominator $\mathrm{dist}(P, F)$ is 0, then $\sigma_P(p)$ is also 0.

The total sensitivity of P is

$$\mathfrak{S}_P := \sum_{p \in P} \sigma_P(p).$$

The total sensitivity function of a shape fitting problem is

$$\mathfrak{S}_n := \sup_{P \subset \mathbb{R}^d, |P| = n} \mathfrak{S}_P.$$

We make two quick remarks regarding deriving upper bound and lower bound of total sensitivity function. (1) In order to prove that the total sensitivity function of a shape fitting problem is upper bounded by a function $f(n)$, one needs to show that for *any* point set P of cardinality n, the summation $\sum_{p \in P} \sigma_P(p)$ is upper bounded by $f(n)$. In order to prove that the total sensitivity function is lower bounded by a function $g(n)$, one only needs to find *some* point set of cardinality n, such that $\sum_{p \in P} \frac{\mathrm{dist}(p, F_p)}{\mathrm{dist}(P, F_p)}$ is at least $g(n)$, where F_p is a shape in $\mathcal{F}$, which witnesses that the sensitivity of p in P is at least $\mathrm{dist}(p, F_p)/\mathrm{dist}(P, F_p)$. (2) Total sensitivity function is always upper bounded by n, since the sensitivity of each point is at most 1 by definition (each point contributes at most 100% of the overall summation of distances from points in P to the shape).

We give a simple example to illustrate the computation of sensitivities of points in a point set. Consider the following instance of a "toy" shape fitting problem: the input point set P has three elements, $P = \{p_1, p_2, p_3\}$. The family of shapes has two members, $\mathcal{F} = \{F_1, F_2\}$. The distance from each point in P to shapes in F is given in the matrix M below, where m_{ij} is the distance from p_i to F_j, $1 \leq i \leq 3$ and $1 \leq j \leq 2$:

$$\begin{array}{cc} & \begin{matrix} p_1 & p_2 & p_3 \end{matrix} \\ \begin{matrix} F_1 \\ \\ F_2 \end{matrix} & \begin{pmatrix} 1 & 4 & 2 \\ \\ 7 & 1 & 2 \end{pmatrix} \end{array}$$

The sensitivity of p_1 in P, $\sigma_P(p_1)$ is

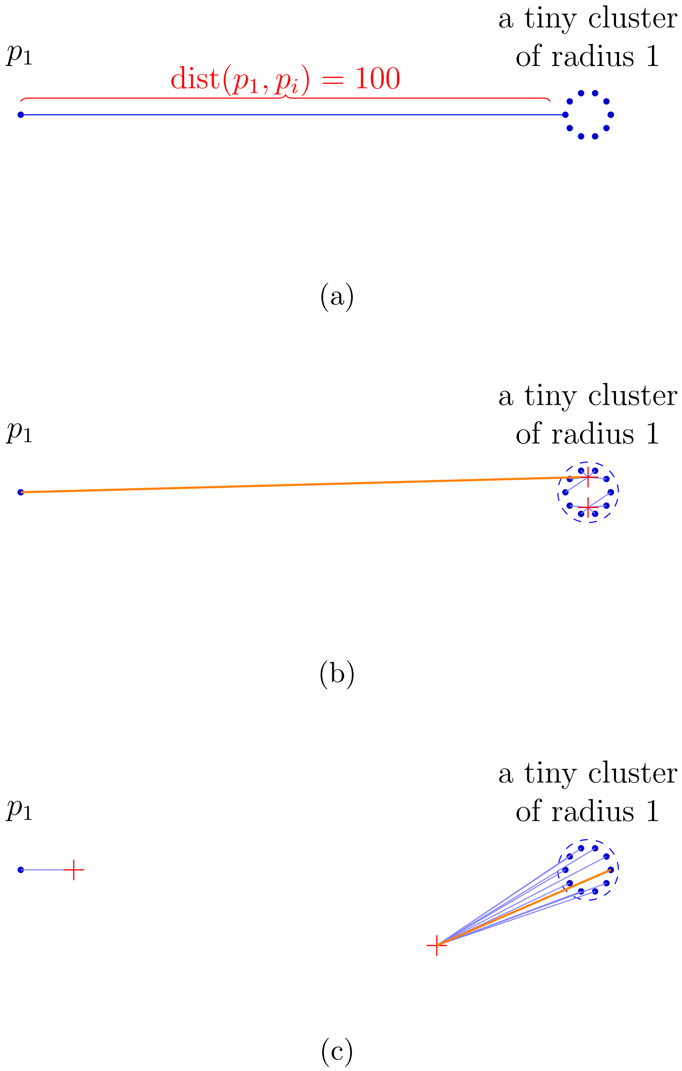
$$\begin{aligned} \sigma_P(p_1) &= \max\left\{\frac{\mathrm{dist}(p_1, F_1)}{\mathrm{dist}(P, F_1)}, \frac{\mathrm{dist}(p_1, F_2)}{\mathrm{dist}(P, F_2)}\right\} \\ &= \max\left\{\frac{\mathrm{dist}(p_1, F_1)}{\mathrm{dist}(p_1, F_1) + \mathrm{dist}(p_2, F_1) + \mathrm{dist}(p_3, F_1)},\right. \\ &\left.\frac{\mathrm{dist}(p_2, F)}{\mathrm{dist}(p_1, F_2) + \mathrm{dist}(p_2, F_2) + \mathrm{dist}(p_3, F_2)}\right\} \\ &= \max\left\{\frac{1}{1+4+2}, \frac{7}{7+1+2}\right\} = \frac{7}{10}. \end{aligned}$$

Similarly, one can easily verify that the sensitivity of p_2 in P, $\sigma_P(p_2)$, is $\max\{4/7, 1/10\} = 4/7$ and $\sigma_P(p_3) = 2/7$. The total sensitivity of P, $\mathfrak{S}_P$, is $\sigma_P(p_1) + \sigma_P(p_2) + \sigma_P(p_3) = 7/10 + 4/7 + 2/7 \approx 1.5$.

Sensitivity of a point in a point set P naturally captures the importance of a point p in P (with respect to the shapes at hand). For example, consider the following instance (Figure 3.2) of 2-means clustering. The point set is formed by a cluster of points, and an "outlier" which is far away from the majority of the points.

Sensitivities of points in a point set reflect the importance of points. The point p_1, which has a high sensitivity, is a very important points in the point set P of Figure 3.2, and each of the points in the small cluster is relatively less important (which is also consistent with their small sensitivities). This statement can be made quite precise in the context of succinct (or compact) representation of P—if we are to pick "representative" points from P to form a sketch of P, it looks like that p_1 is indispensable, and for the rest of the points, it seems that if we pick an arbitrary point from the small cluster with proper weight, we get a fairly accurate sketch of P, as shown in Figure 3.3.

So far, one perhaps has already vaguely felt that sensitivities of points in each point set seem to provide some clue on the construction of coreset: in order to form a sketch of a point set, points with high sensitivities look quite indispensable, while points with low sensitivities seem less so. Recall that our goal is to find a *small* sketch, in the next section, we will show that there is a close connection between total sensitivity of a point set (for a shape fitting problem) and the size of the coreset.

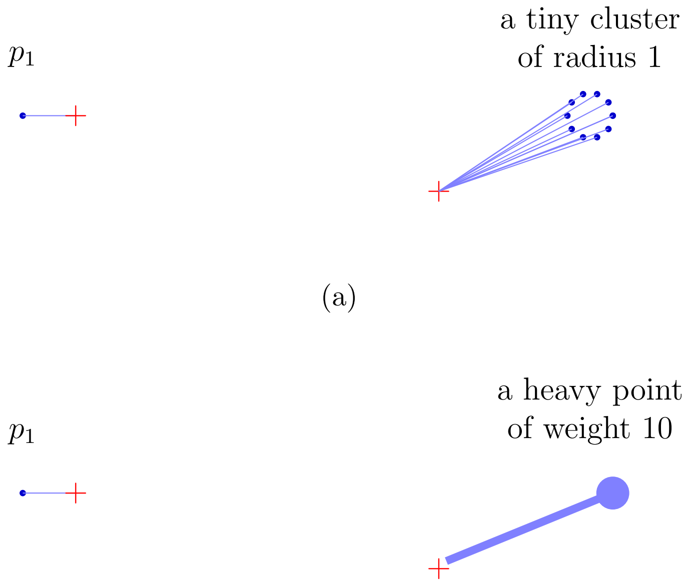


(a)

(b)

(c)

Figure 3.2. Illustration that sensitivity of a point reflects the “ourlierness”/“importance” of the point among the point set (with respect to a family of shapes). (a) shows the point set, where p_1 is far away from the rest of points $p_2, \cdots, p_{11}$; (b) shows that when the two centers (red crosses) are very near to the tiny cluster, the overall cost $\sum_{p\in P} \text{dist}(p, F)$ is almost solely contributed by p_1, hence p_1 is a very “important” point and has sensitivity almost 1; (c) shows that the sensitivity of each point in the tiny clustering is small, since for every point in the tiny cluster, the distance from this point to the center (red crosses) are almost the same as the rest of points in the cluster. Hence each point in the tiny cluster contributes roughly the same fraction to the overall cost, $\sum_{p\in P} \text{dist}(p, F)$. Thus each point has a small sensitivity roughly between 1/9 and 1/10.



(a)

(b)

Figure 3.3. Picking representative points from P to get a sketch of P.

3.2 Total sensitivity and the size of coreset

In this section, we show that point set with small total sensitivity admits small coreset. More explicitly, the smaller total sensitivity is, the smaller the size of coreset is.

To get a feel on the relation between total sensitivity and the size of coreset, we give a crude (and possibly oversimplified) reasoning. The space we consider here is the 2-dimensional Euclidean space $\mathbb{R}^2$. Consider two family of shapes, $\mathcal{F}_1$ and $\mathcal{F}_2$, where $\mathcal{F}_1$ is the family of very complex shapes—which consists of all subsets of $\mathbb{R}^2$; while $\mathcal{F}_2$ consists of extremely simple shapes—only horizontal lines. The distance here is Euclidean distance. The instance for these two shape fitting problems is an n-point set P on a line. First consider the complex family of shapes $\mathcal{F}_1$. The total sensitivity of P with respect to $\mathcal{F}_1$ is n, since for any p_i in P, the shape formed by the union of the points in $P \setminus \{p_i\}$ witnesses that $\sigma_P(p_i) = 1$, for $1 \leq i \leq n$ (see Figure 3.4). The coreset for $\mathcal{F}_1$ is P: indeed, every point in P is indispensable, because if we omit any p in P, the cost of fitting P with the shape F which contains all the points in P except p, is strictly positive (as p is not contained in this shape), while on the other hand, the cost of fitting $P \setminus \{p\}$ with F is 0 as all other points contributes 0 to the overall cost (and also note that increasing the weights to other points in $P \setminus \{p\}$ would not work.) Therefore, the complexity of shapes in $\mathcal{F}_1$ forces the coreset to include every member of P in this case.

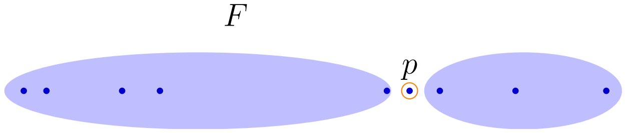


Figure 3.4. An illustration that the complex family of shapes $\mathcal{F}_1$ forces the coreset to include every point in P. F witnesses that the sensitivity of p in P is 1, as only p contributes a strictly positive quantity of $\sum_{p \in P} \mathrm{dist}(p, F)$; if p is not included in the sketch S of P and S solely consists of points from $P \setminus \{p\}$, $\mathrm{cost}(S, F)$ would be 0, while $\mathrm{cost}(P, F)$ is strictly positive.

Now consider the simple family of horizontal lines $\mathcal{F}_2$. The total sensitivity of P with respect to $\mathcal{F}_2$ is 1: each point has sensitivity $1/n$ for any shape F, every points contributes exactly the same amount to the overall summation $\sum_{p\in P}\operatorname{dist}(p,F)$, as can be easily seen from Figure 3.5. The coreset is also quite simple: any point in P with multiplicity n is a coreset; for example, the set $S=\{p\}$, where the weight of p is n, is a coreset. Indeed, from the view of a shape, the points in P are completely indistinguishable: they all contribute the same amount to $\operatorname{dist}(P,F)$, for any $F\in\mathcal{F}_2$. Therefore, we can summarize the point set with a single weighted point. From the analysis of these two situations, it should be clear that total sensitivity quantifies the complexity of a shape fitting problem, in the sense that small total sensitivity implies small coreset.

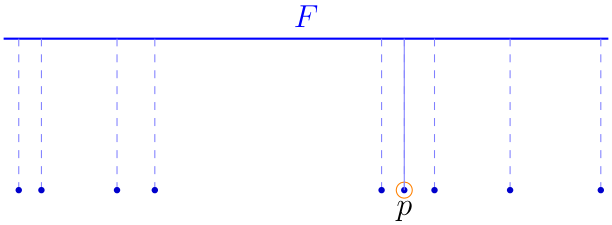


Figure 3.5. An illustration that the simple family of shapes $\mathcal{F}_2$ has a small coreset of $\{p\}$, where the weight of p is n. The sensitivity of p in P is $1/n$, as for any F, each point contributes the same quantity to $\sum_{p\in P}\operatorname{dist}(p,F)$; so $\sum_{p\in P}\operatorname{cost}(p,F)=n\operatorname{dist}(p,F)$, which is exactly $\operatorname{cost}(S,F)=w(p)\operatorname{dist}(p,F)$.

We give a more non-trivial example, to show that total sensitivity increases as the complexity of shapes increases. Consider the family $\mathcal{F}$ of lines, which is a very simple shape, and its more complex variant, the family $\mathcal{F}'$ of rays. The family of rays is more complex than the family of lines, since one needs to specify a starting point for ray, after specifying the line on which the ray lies. A deeper reason (in the context of coresets for shape fitting problem) is that a ray can always behave like a line, so there is indeed more shapes with respect to which a sketch S of P has to approximate. It is probably the best to look at the Figure 3.6 to see this. As we will shown, on the plane, the total sensitivity for any point set P for a family of lines is a constant, independent of the cardinality of P; while the total sensitivity of P for a family of rays is at least $O(\log n)$. The increase of total sensitivity from constant to $O(\log n)$ reflects that total sensitivity captures the complexity of shapes.

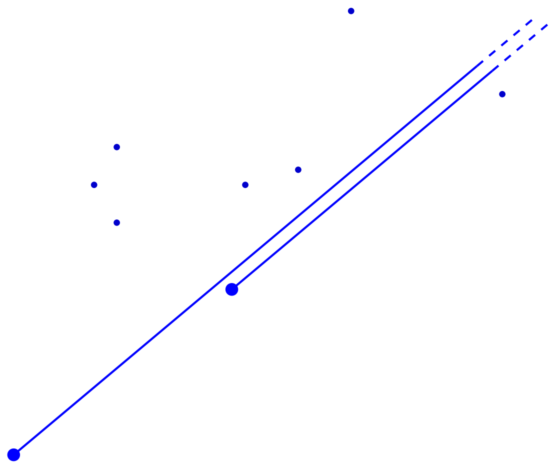


Figure 3.6. For a point set P, by moving the ray far left enough, the ray is indistinguishable from a line.

The remaining question is: how to compute a coreset using sensitivities and total sensitivity? One fairly natural approach is “sampling by importance”: since we already have a method to measure the importance of each point in a point set (with respect to the underlying shape fitting problem) via the notion of sensitivity, $\sigma_P(\cdot)$, the most intuitive method is to sample the points according to their sensitivities: that is, the probability that a point p in P is picked is $\sigma_P(p)/\sum_{q\in P}\sigma_P(q)$, which is $\sigma_P(p)/\mathfrak{S}_P$. Note that this precisely reflects the interpretation that the more outlier/important a point is, the greater the chance that the point is selected. In order to get an accurate sketch of P, one only needs to repeat the sampling (without replacement) for a sufficient number of iterations and to assign weights properly. Since we do not want to keep the reader in suspense, we will present the algorithm (Algorithm 2) of computing coresets using sensitivities/total sensitivity at this time point. The number of samples that need to be draw, depends quadratically on the total sensitivity function, and another parameter characterizing the complexity of shapes, $\dim(P)$, which will be explained in detail in Chapter 6. We remark that $\dim(P)$ is related to a notion in range space (Vapnik-Chervonenkis dimension), and it is generally not hard to compute. $\dim(P)$ for (j,k) projective clustering problems is $O(jkd)$, and constant for circle fitting.

The following theorem which connects total sensitivity with coreset is from [19], and it will be explained in Chapter 6 for the sake of completeness. For the time being, it can be considered as a black box, which produces a coreset as long as one computes (an upper bound of) the sensitivities $\sigma_P(p)$ for each p in the input point set P, and total sensitivity $\mathfrak{S}_P$.

Theorem 1 (Connection between total sensitivity and ϵ-coreset [19])**.** Given any n-point instance $P \subset \mathbb{R}^d$ of a shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$, and any $\epsilon \in (0,1]$,

Algorithm 2: Low variance sampling algorithm using $\sigma_P(\cdot)$ and $\mathfrak{S}_P$

```
Input: A point set $P \subset \mathbb{R}^d$ for a shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$, $\epsilon \in (0,1)$,
       $\delta \in (0,1)$, the number of samples $m = O(\epsilon^{-2}(\mathfrak{S}_P)^2(\dim(P) + \log 1/\delta))$
Output: A (multi-)point set $S$ of size $m$, together with weights $w(p)$ for each
        $p \in S$, such that $S$ is an $\epsilon$-coreset of $P$ with probability at least $1 - \delta$.
for $p \in P$ do
    compute $\sigma_P(p)$;
end
$\mathfrak{S}_P \leftarrow \sum_{p \in P} \sigma_P(p)$;
$S \leftarrow \emptyset$;
for $i \leftarrow 1$ to $m$ do
    Randomly pick a point from $P$, where the probability of $p \in P$ being picked
    is $\sigma_P(p)/\mathfrak{S}_P$. Suppose the selected point is $p \in P$;
    $S \leftarrow S \cup \{p\}$;
    $w(p) \leftarrow \mathfrak{S}_P/(m\sigma_P(p))$;
end
```

there exists an ϵ-coreset for P of size

$$O\left(\left(\frac{\mathfrak{S}_n}{\epsilon}\right)^2 \dim(P)\right).$$

The algorithmic result is that Algorithm 2 outputs an ϵ-coreset of P of size

$$O\left(\left(\frac{\mathfrak{S}_n}{\epsilon}\right)^2 \left(\dim(P) + \log\frac{1}{\delta}\right)\right)$$

with probability at least $1 - \delta$.

In the next two chapters, we focus on deriving upper bounds of total sensitivities for a family of projective clustering problems and the circle fitting problem on the plane.

CHAPTER 4

FROM L_∞ CORESET TO L_1 CORESET

In this chapter, we derive upper bounds for several projective clustering problems, and the circle fitting problem. The central problem considered in this chapter is the following:

Problem 3 (Upper bound the total sensitivity of the shape fitting problem with $o(n)$)**.** Given a shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$,

- is there any connection between L_∞ coreset and total sensitivity of $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$?
- For the shape fitting problem, where the shape fitting problem is either (a) circle fitting, (b) k-line clustering problem, or (c) integer (j, k) projective clustering, are the total sensitivities for these problems $o(n)$?

We answer affirmatively the first question in Problem 3, the answer of which is stated in Theorem 2. The answer to the first question immediately helps us to answer the second question in Problem 3, the answer of which is stated in Theorem 3.

Organization of this chapter: We first show that there is a connection between L_∞ coreset and L_1 coreset in Section 4.1. In Section 4.2, we apply this connection to several shape fitting problems (circle fitting, k-line clustering, and integer (j, k) projective clustering) to derive $o(n)$ upper bounds of total sensitivities. Using the black box of computing ϵ-coreset via sensitivities/total sensitivity (Theorem 1), we obtain small coreset for these shape fitting problems. In Section 4.3, we apply our result of small coreset for (j, k) projective clustering to get the first near linear algorithm for integer (j, k) projective clustering.

We remind the reader that this chapter studies shape fitting problem in fixed dimension (the dimension d is considered as a constant). So the upper bounds of the sizes of the coreset in this chapter depend exponentially on d. The dependence on the number of input points is polylogarithmic.

4.1 Total sensitivity and L_∞ coreset

In this section, we show a connection between L_∞ coreset and L_1 coreset. In particular, this result can be summarized as a shape fitting problem which admits small L_∞ coreset also has small total sensitivity, thus also has small L_1 coreset. Before the technical details, we first give a high-level overview, and some intuition on this connection. Let us consider the line fitting problem on the plane, and consider the point set shown in Figure 4.1. Recall the L_∞ δ-coreset of P is a subset $Q_1 \subset P$, such that for any line, the furthest distance from points in Q_1 is at least $1 - \delta$ times the furthest distance from points in P to the line. In other words, L_∞ coreset Q_1 roughly approximates the outline (or boundary) of the region where the points in P are from. A simple exact L_∞ coreset is the set of points on the convex hull of P, as shown in Figure 4.1.

Each point in Q_1 is a witness that each remaining point in $P \backslash Q_1$, has sensitivity

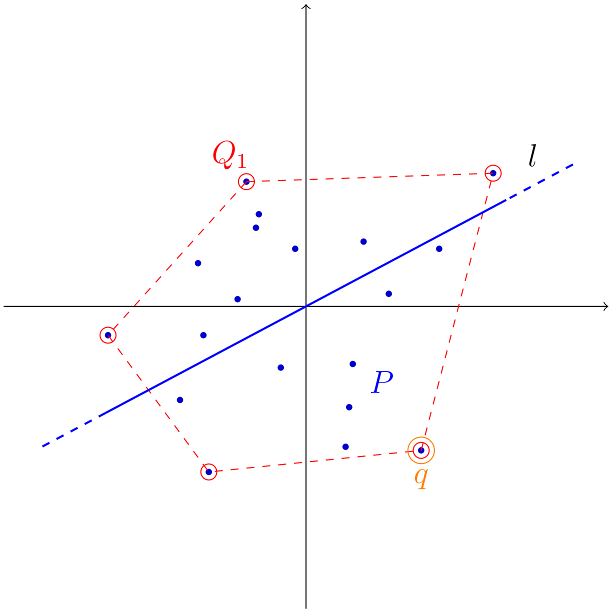


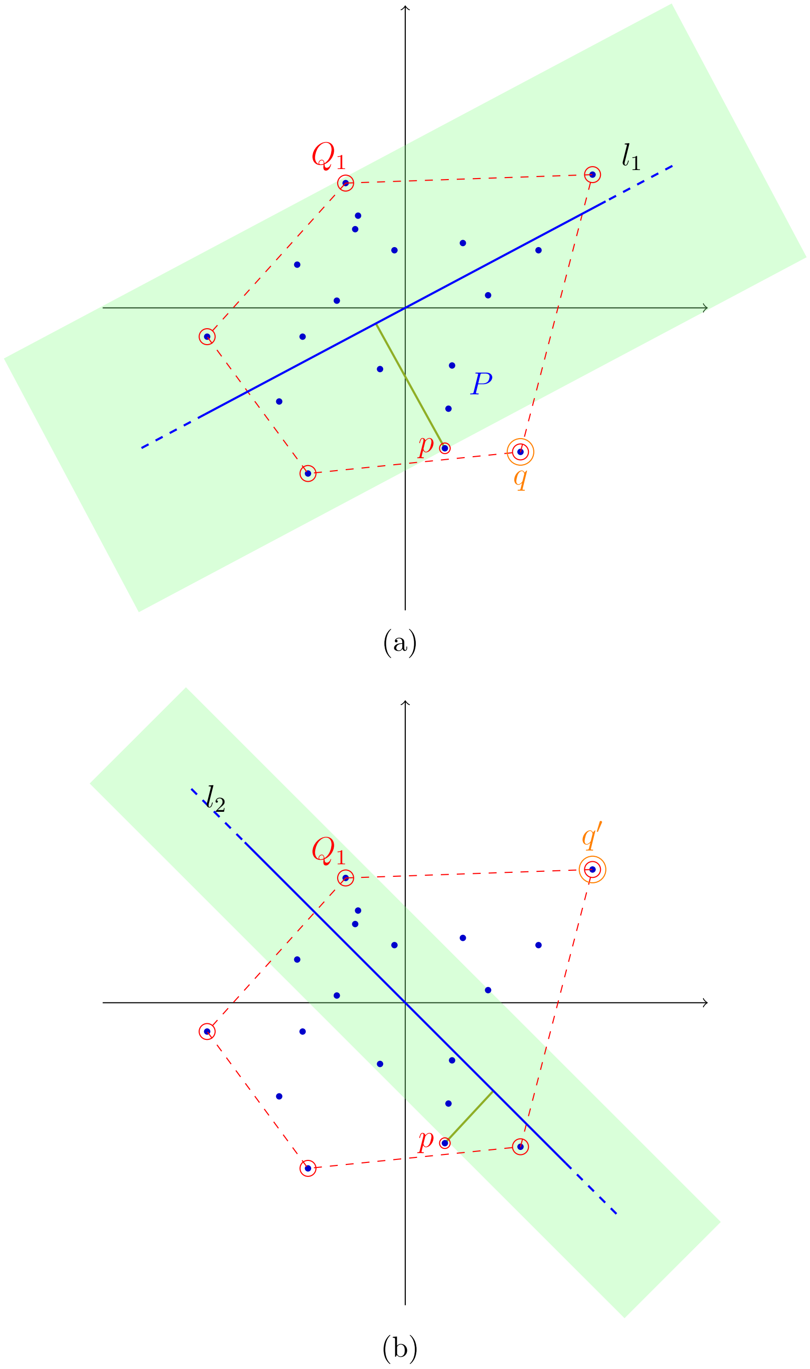
Figure 4.1. The set of points on the convex hull of a point set P (blue points) is an exact L_∞ coreset. For example, the furthest point from P to the line l is q, which is in Q_1.

at most 1/2. Consider a point $p \in P \setminus Q_1$. For any line, there is always exists some point q from Q_1 that is further than p, as illustrated in Figure 4.2(a). Therefore, for line l, we have

$$\frac{\text{dist}(p,l)}{\text{dist}(P,l)} \leq \frac{\text{dist}(p,l)}{\text{dist}(p,l)+\text{dist}(q,l)} \leq \frac{1}{2}. \tag{4.1}$$

As can be seen from Figure 4.2, this is true for any line l, therefore, the sensitivity of p is at most 1/2.

This observation suggests a “peeling” argument to bound the sensitivities: we first compute an L_∞ coreset Q_1 of P, and assign each point in Q_1 an upper bound of sensitivity 1 (since sensitivity by definition cannot exceed 1); peel off this layer



(a)

(b)

Figure 4.2. Illustration that the sensitivity of any point in $P \setminus Q_1$ is at most $1/2$. (a) shows that for the line l_1, the point $q \in Q_1$ is further than p to line l_1. (b) shows that for the line l_2, the point $q' \in Q_1$ is further than p to line l_2.

from P, we get $P \setminus Q_1$. We now compute an L_∞ coreset Q_2 of $P \setminus Q_1$, and assign each point in Q_2 an upper bound of $1/2$. This is because Q_2 is a subset of $P \setminus Q_1$ by construction, and each point in $P \setminus Q_1$ has sensitivity at most $1/2$, as reasoned above (Eq (4.1)). We repeat this peeling process: in the i^{th} iteration, the remaining points is in $P \setminus \cup_{j=1}^{i-1} Q_j$. We compute an L_∞ coreset Q_i of this point set, and assign every point in Q_i an upper bound of sensitivity $1/i$. The reasoning is similar to the case when $i = 2$: for a point q_i in Q_i, for any arbitrary shape F, each layer Q_1, $Q_2, \cdots, Q_{i-1}$ contains a distinct "witness" point, which is further away from F than q_i. This peeling process continues until every point is peeled off. If there are n points in P, and each time we peel off c points, we get at most $\frac{n}{c}$ layers of P, therefore, this argument shows that the total sensitivity of P is at most

$$c \times 1 + c \times \frac{1}{2} + \cdots + c \times \frac{1}{n/c} \le c \times \left(1 + \frac{1}{2} + \cdots \frac{1}{n}\right) \le c \log n.$$

Figure 4.3 shows this peeling process on P (each time an L_∞ coreset of size at most 5 is peeled off). Figure 4.3(b) shows that a point in the third layer, Q_3, indeed has sensitivity at most $1/3$.

We now show the rigorous proof of the above peeling argument for bounding total sensitivities. Recall the definition of L_∞ coreset of an instance P of a shape fitting problem:

Definition 2 (L_∞ coreset for a shape fitting problem)**.** Given an n-point set P of a shape fitting problem, and $\delta \in [0, 1]$, a subset $Q \subset P$ is an L_∞ δ-coreset of P, if for every $F \in \mathcal{F}$, it holds that

$$\max_{q \in Q} \operatorname{dist}(q, F) \geq (1-\delta) \max_{p \in P} \operatorname{dist}(p, F).$$

(the distance from the furthest point in Q to a shape F is at least $(1-\delta)$ times the distance from the furthest point in P to a shape F.) Also note that since $Q \subset P$,

$$\max_{q \in Q} \operatorname{dist}(q, F) \leq \max_{p \in P} \operatorname{dist}(p, F).)$$

The following theorem shows that if a shape fitting problem admits a small L_∞ coreset, then its total sensitivity is also small (that is, $o(n)$).

Theorem 2 (Small L_∞ coreset and small total sensitivity)**.** Given a shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \operatorname{dist})$. Suppose that for some $0 \leq \delta < 1$, there is a non-decreasing function $f_\delta(n)$ so that any point set $P' \subset \mathbb{R}^d$ of size n admits an L_∞ δ-coreset of size at most $f_\delta(n)$, then for any $P \subset \mathbb{R}^d$ of size n, we can compute an upper bound $s_P(p)$ of the sensitivity $\sigma_P(p)$, for every $p \in P$, so that

$$\sum_{p \in P} s_P(p) \leq \frac{f_\delta(n)}{1-\delta} \log n.$$

Proof. We construct a sequence of subsets $P = P_1 \supseteq P_2 \supseteq P_3 \cdots P_m$, where $m \leq n$ and $|P_m| \leq f_\delta(n)$. P_{i+1} is constructed from P_i as follows. If $|P_i| \leq f_\delta(n)$, the sequence ends. Otherwise, we compute an L_∞ δ-coreset Q_i of P_i whose size is at most $f_\delta(n)$, and let $P_{i+1} = P_i \setminus Q_i$. This finishes the description of the construction.

Let Q_m denote the set P_m. Now, the sets $Q_1, Q_2, \ldots, Q_m$ partition P. We claim that for any $q \in Q_i$, its sensitivity $\sigma_P(q)$ can be upper bounded by $s(q) = \frac{1}{(1-\delta)i}$. To show this, consider an arbitrary shape $F \in \mathcal{F}$. Consider any $1 \leq j \leq i$.

Observe that $q \in P_j$; let $q_j \in Q_j$ be the point in the δ-coreset Q_j of P_j such that $\mathrm{dist}((,q)_j, F) = \max_{p \in Q_j} \mathrm{dist}(p, F)$. We have

$$\mathrm{dist}(q_j, F) = \max_{p \in Q_j} \mathrm{dist}(p, F) \geq (1-\delta) \cdot \max_{p \in P_j} \mathrm{dist}(p, F)$$
$$\geq (1-\delta) \cdot \mathrm{dist}(q, F).$$

Thus $\frac{\mathrm{dist}(q,F)}{\sum_{p \in P} \mathrm{dist}(p,F)} \leq \frac{\mathrm{dist}((,q),F)}{\sum_{1 \leq j \leq i} \mathrm{dist}((,q)_j,F)} \leq \frac{1}{(1-\delta)i}$. Therefore, $\sigma_P(q) \leq s(q) = \frac{1}{(1-\delta)i}$.

Finally, $\sum_{p \in P} s(p) = \sum_{i=1}^{m} \frac{|Q_i|}{(1-\delta)i} \leq f_\delta(n) \sum_{i=1}^{m} \frac{1}{(1-\delta)i} \leq \frac{f_\delta(n) \log n}{1-\delta}$. □

The upper bound in Theorem 2 is tight: in order to get an asymptotically smaller upper bound of total sensitivity (*i.e.* $o(\log n)$), some other properties of the family $\mathcal{F}$ of shapes, other than the assumption that it has small L_∞ coreset, must be used. We show two families of shapes, each family of shapes admits a constant size L_∞ coreset, and the total sensitivity function $\mathfrak{S}_n$ is $\Omega(\log n)$.

Observation 1 (Rays extending to infinity on the right side)**.** Consider the shape fitting problem $(\mathbb{R}^2, \mathcal{F}, \mathrm{dist})$, where $\mathcal{F}$ consists of rays: $\mathcal{F} = \{F_a | a \in \mathbb{R}\}$, where $F_a = \{(x, 0) \in \mathbb{R}^2 | x \geq a\}$ (a ray starting from $(a, 0)$). The distance from a point p to a shape F_a is $a - p$ if $a \geq p$ and 0 otherwise (the point is contained in the open ray in this case). It is easy to see that any point set P' admits an exact L_∞ coreset (the δ is 1 in this case), which is the left-most point in P'. However, the total sensitivity function $\mathfrak{S}_n$ is $\Omega(\log n)$. We show a point set whose total sensitivity is $\Omega(\log n)$. Let P be a set of n points, where $p_0 = (0, 0)$, and $p_i = (\sum_{j=0}^{i-1} 2^j, 0)$ for $i = 1, \cdots, n-1$. Consider p_i and a shape $F_{\sum_{j=0}^{i} 2^j}$, for $3 \leq i \leq n$:

$$\sigma_P(p_i) \geq \frac{\mathrm{dist}(p_i, F_{\sum_{j=0}^{i} 2^j})}{\mathrm{dist}(P, F_{\sum_{j=0}^{i} 2^j})} = \frac{2^{i-1}}{1 + \sum_{j=1}^{i} j 2^{j-1}} \geq \frac{2^{i-1}}{(i-2)2^{i-1}} = \frac{1}{i-2}.$$

Hence $\mathfrak{S}_P \geq \sum_{i=3}^{n} \frac{1}{i-2}$, which is $\Omega(\log n)$.

Observation 2 (2-line clustering)**.** Consider the shape fitting problem $(\mathbb{R}^2, \mathcal{F}, \text{dist})$, where $\mathcal{F}$ consists of pairs of lines (each shape is a union of two lines). The distance from a point to a shape formed by two lines l_1 and l_2 is the minimum Euclidean distance from the point to the nearest line. This shape fitting problem admits a constant size L_∞ coreset. We now show an n-point set P whose total sensitivity is $\Omega(\log n)$. Let P be the following point set in $\mathbb{R}^2$: $p_i = (1/2^{-1}, 0)$, for $i = 1, \cdots, n$. Let F_i be a pair of lines: one vertical line and one horizontal line, where the vertical line is y-axis, and the horizontal line is $\{(x, 1/2^i) | x \in \mathbb{R}\}$.

Consider the point p_i, where $i = 1, \cdots, n$. We show that $\text{dist}(p_i, F_i)/\text{dist}(P, F_i)$ is at least $1/(2+i)$ for $i = 1, \cdots, n$. For $j \leq i$, note that $\text{dist}(p_j, F_i) = 1/2^i$: since the distance from p_j to the horizontal line in F_i is $1/2^i$ and the distance to the vertical line is $1/2^{j-1}$, $\text{dist}(p_j, F_i) = \min\{1/2^{j-1}, 1/2^i\} = 1/2^i$. For $i+1 \leq j \leq n$, on the other hand, $\text{dist}(p_j, F_i) = 1/2^{j-1}$. Therefore, $\sum_{j=i+1}^{n} \text{dist}(p_j, F_i) = \sum_{j=i+1}^{n} 1/2^{j-1} = (1/2^{i-1}) \cdot (1 - (1/2)^{n-i})$. Thus, we have

$$\sigma_P(p_i) \geq \frac{\text{dist}(p_i, F_i)}{\text{dist}(P, F_i)} = \frac{1/2^i}{(1/2^{i-1} - 1/2^{n-1}) + i \cdot (1/2^i)} > \frac{1}{2+i}.$$

Therefore, $\mathfrak{S}_P \geq \sum_{i=1}^{n} \frac{1}{2+i}$, which is $\Omega(\log n)$.

The proof in Theorem 2 is constructive. Algorithm 3 computes the sensitivities for each $p \in P$.

One subtle thing to note in Theorem 2 and also Algorithm 3 is that δ is not part of the input: as long as for *some* (instead of every) $\delta \in [0,1)$ there is a procedure `LInfinityCoreset` to compute a small L_∞ coreset of size $f_\delta(|P|)$ for input point set P, we can get an upper bound of total sensitivity using Theorem 2 (and also use Algorithm 3). We give a pedagogical example here. Consider the k-median clustering problem in $\mathbb{R}^d$. Any point set $P \subset \mathbb{R}^d$ admits an L_∞ (2/3)-coreset of size $k+1$: such a coreset can be obtained by starting with an arbitrary point in P, denoted p_1. For $1 \le i \le k$, letting p_{i+1} be the point in P furthest from $\{p_1, \cdots, p_i\}$. Using this procedure in the place of `LInfinityCoreset` in Algorithm 3, one obtains an upper bound of $(k+1)/(1-2/3)\log n$ (which is $O(k \log n)$) on the total sensitivity of P. Although for all the problems where we are going to apply Theorem 2, we already have the L_∞ δ-coreset construction for any $\delta \in (0,1)$, the requirement in the Theorem 2 is indeed

Algorithm 3: Compute the upper bound $s_P(\cdot)$ of $\sigma_P(\cdot)$ using L_∞ δ-coreset

```
Input: A point set $P$
Output: An upper bound of the sensitivity of $p$ in $P$, $s_P(p)$, for each $p \in P$
$i \leftarrow 1$;
while $P \neq \emptyset$ do
    $Q \leftarrow$ LInfinityCoreset($P$) ;    // LInfinityCoreset($P$) computes an $L_\infty$ $\delta$-coreset of $P$ of size at most $f_\delta(|P|)$.
    for $q \in Q$ do
        $s_P(q) \leftarrow \frac{1}{(1-\delta)i}$;
    end
    $P \leftarrow P \setminus Q$;
    $i \leftarrow i + 1$;
end
```

weaker.

4.2 Total sensitivities of k-line clustering, integer (j,k) projective clustering and circle fitting

In this section, we derive upper bounds of total sensitivities for three shape fitting problems using Theorem 2: k-line clustering, integer projective (j,k) clustering, and circle fitting problems. We use previous results of L_∞ coreset for these shape fitting problems.

Theorem 3 (Upper bound of total sensitivities for circle fitting, k-line clustering, and integer (j,k) projective clustering)**.** Let $P \subset \mathbb{R}^d$ be an n-point set of a shape fitting problem, where the shape fitting problem is either (a) circle fitting, (b) k-line clustering problem, or (c) integer (j,k) projective clustering. We can compute in $O(n(\log n)^{O(1)})$ time an upper bound $s_P(p)$ on the sensitivity $\sigma_P(p)$ for each $p \in P$ so that $\sum_{p\in P} s_P(p) \leq (\log n)^{O(1)}$. For the k-line clustering problem, the constant in the exponent of the logarithm depends on k and d, and for the integer (j,k) projective clustering problem, it depends on j, k and d.

Proof. **Circle Fitting**: An L_∞ 1/2-coreset of size $O(1)$ can be computed for any n-point set can be computed in time $O(n)$, see for example [6] and [7]. Using the dynamization technique described in these papers, such a 1/2-coreset can be maintained in $(\log n)^{O(1)}$ time per insert or delete. The result follows using Theorem 2 and the remarks following its proof on the implied algorithm and its dynamization.

k-line clustering: An L_∞ 1/2-coreset of size $O(1)$ (with the constant depending on j) exists for any n-point set [8], but the construction in that paper does not describe an efficient enough algorithm for constructing such a coreset. Nevertheless, using techniques that are now standard, a 1/2-coreset of size $(\log n)^{O(1)}$ can be computed in $O(n(\log n)^{O(1)})$ time. The dynamization technique described in [7] allows us to maintain a 1/2-coreset in $(\log n)^{O(1)}$ time per insertion and deletion.

Integer (j, k) projective clustering: An L_∞ 1/2-coreset of size $(\log \Delta \cdot \log n)^{O(1)}$ can be computed in time $n(\log \Delta \cdot \log n)^{O(1)}$ for any n-point set with integer coordinates and diameter Δ [17]. The dynamization technique in [7] allows us to maintain a 1/2-coreset in $(\log \Delta \log n)^{O(1)}$ time per insertion and deletion. The result follows by recalling that Δ is $(nd)^{O(1)}$ for any input to the integer projective clustreing problem with n points. □

Plugging the upper bounds of total sensitivities stated in Theorem 3 in Theorem 1, we get small coresets in fixed dimension.

Theorem 4 (Small coresets for circle fitting, k-line clustering and integer (j, k) projective clustering)**.** Let $P \subset \mathbb{R}^d$ be an n-point set of circle fitting problem $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$, where the shape fitting problem is either (a) circle fitting, (b) k-line clustering problem, or (c) integer (j, k) projective clustering. Given $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$. With probability at least $1-\delta$ we can compute in (a) $O(n(\log n)^{O(1)} + (\log n)^{O(1)}\epsilon^{-2}(O(1) + \log(1/\delta)))$, (b) $O(n(\log n)^{O(1)} + (\log n)^{O(1)}\epsilon^{-2}(O(kd) + \log(1/\delta)))$, (c) $O(n(\log n)^{O(1)} + (\log n)^{O(1)}\epsilon^{-2}(O(kdj) + \log(1/\delta)))$ time an ϵ-coreset of size (a)$O((\log n/\epsilon)^2(O(1) + \log(1/\delta)))$ for circle fitting problem; (b)$O(\epsilon^{-2}(\log n)^{f(k,d)}(O(kd) + \log(1/\delta)))$ for k-line clustering problem; (c) $O(\epsilon^{-2}(\log n)^{f(k,j,d)}(O(kdj) + \log(1/\delta)))$ for integer (j, k)-projective clustering problem, where $f(k, d)$ is a function of k and d, and $f(k, d, j)$ is a function of k, d, and j.

Proof. The sizes of the coresets follows directly from Theorem 1, by substituting the upper bounds of total sensitivity $O((\log n)^{O(1)})$ into the formula. The running time is the the summation of the time for computing the upper bound of sensitivity, $s_P(p)$ for each $p \in P$, and the time for sampling. The time for computing the upper bound of sensitivity is $O(n(\log n)^{O(1)})$, as proved in Theorem 3. The time for sampling is linear to the the size of coresets, which is $O((\mathfrak{S}_n/\epsilon)^{-2}(\dim(P, \mathcal{R}(P)) + \log(1/\delta)))$. The factor, $\dim(P, \mathcal{R}(P))$ is constant for circle fitting, and $O(kdj)$ for (j, k) projective clustering [19]. Hence the theorem follows. $\square$

4.3 Near linear algorithm for integer (j, k) projective clustering problem

In this section, we describe the near linear algorithm for integer (j, k) projective clustering in $\mathbb{R}^m$ using the coreset we obtained in the last section. Let $P \subseteq \mathbb{R}^m$ be an input instance of n points with integer coordinates of magnitude at most $\Delta = (mn)^{10}$, and $0 < \epsilon < 1$ be a parameter. We describe an algorithm that runs in $O(mn(\log mn)^{O(1)})$ and returns a shape $F \in \mathcal{F}$ (a union of j k-flats) that with probability at least a constant is nearly optimal: $\text{cost}(P, F) \leq (1 + \epsilon)\text{cost}(P, F')$ for any $F' \in \mathcal{F}$. Note that we consider j and k constants but the dimension m as part of the input. We have used m rather than d to denote the dimension of the host space to emphasize that here, unlike in the last two sections, it is not a constant. For simplicity, we assume that the shape we are trying to fit is a union of k linear j-subspaces in $\mathbb{R}^m$, as opposed to a union of affine subspaces.

The result is obtained in three steps. First, we use a known dimension reduction result to reduce the problem to a (j,k) projective clustering in constant dimension. To solve the projective clustering problem in constant dimension, we compute a small coreset using essentially Theorem 4. In the third step, we solve the projective clustering problem on the coreset nearly optimally in time polynomial in the size of the coreset.

4.3.1 Dimension reduction

Using the algorithm of Deshpande and Varadarajan [16], we compute in time $nm\left(\frac{kj}{\epsilon}\right)^{O(1)}$ a linearly independent subset $\{a_1, a_2, \ldots, a_{d'}\} \subseteq P$ whose span contains (with probability at least 0.9) a shape $F \in \mathcal{F}$ such that $\text{cost}(P,F) \leq (1+\epsilon)\text{cost}(P,F')$ for any $F' \in \mathcal{F}$. Here, $d' = \left(\frac{kj}{\epsilon}\right)^{O(1)}$ is a constant. Let V denote the subspace spanned by $\{a_1, a_2, \ldots, a_{d'}\}$. It now suffices to solve the following problem nearly optimally: among the shapes in $\mathcal{F}$ that are *contained* in V, find the one that minimizes $\text{cost}(P, \cdot)$.

Fix $b \in \mathbb{R}^m$ orthogonal to V. For $p \in P$, let $\bar{p}$ denote the orthogonal projection of p onto V and $p^{\perp}$ the projection of p onto the orthogonal complement of V. For $p \in P$, let $p' = \bar{p} + ||p^{\perp}||_2 b$, and let $P' = \{p' \mid p \in P\}$. Observe that $\text{cost}(P,F) = \text{cost}(P',F)$ for any $F \in \mathcal{F}$ that is contained in V. It therefore suffices to solve the following problem nearly optimally: among the shapes in $\mathcal{F}$ that are *contained* in V, find the one that minimizes $\text{cost}(P', \cdot)$. This is a (j,k) projective clustering problem in $d'+1$ dimensions, except for the additional constraint that the shape must lie in the d'-dimensional subspace V.

4.3.2 Computing a coreset

Our next step is to compute an L_1 ϵ-coreset Q for P' using Theorem 4, treating P' as a point set in $d'+1$ dimensions. For any $p' \in P'$, we have $||p'||_2 = ||p||_2 \leq \sqrt{m}\Delta$; however, the coordinates of p' when expressed in terms of an orthonormal basis for the subspace spanned by V and b are not necessarily integers. So we have to address this technicality before applying Theorem 4. This is not hard to do given the following lemma.

Lemma 1. Let F be an optimal solution for the (j, k) projective clustering problem on the point set P. If $\text{cost}(P, F) > 0$, then $\text{cost}(P, F) > \frac{1}{(m\Delta)^c}$, for some constant c that depends only on j.

Proof. We first need the following observation.

Claim 1. Let $\{p_1, p_2, \ldots, p_{j+1}\}$ be any linearly independent subset of P. The $(j+1)$-dimensional volume of the simplex spanned by this subset is at least $\frac{1}{((j+1)!)^2}$.

Proof. Let A be the $(j + 1) \times m$ matrix whose rows are the vectors p_i. Then the volume of the simplex in question is $\frac{1}{((j+1)!)^2} \det(AA^T)$. The matrix AA^T has entries that are all integers. □

Suppose that F, the optimal solution is a union of the k j-subspaces $f_1, f_2, \ldots, f_k$. Let $P_1, \ldots, P_k$ be the partition of P obtained by assigning each point in P to the nearest of these k subspaces. Assuming $\text{cost}(P, F) > 0$, at least one of the sets, say P_i, contains $(j + 1)$ linearly independent points $\{q_1, \ldots, q_{j+1}\}$. Let f_i' be a j-subspace in the span of $\{q_1, \ldots, q_{j+1}\}$ that contains the projection of f_i on this span. Then the set $\{q_1, \ldots, q_{j+1}\}$ is contained in a $(j+1)$-dimensional box, j of whose sides have length $2\max_{t=1}^{j+1} ||q_t||_2$ and whose $(j+1)$-th side has length $2\max_{t=1}^{j+1} \mathrm{dist}(q_t, f_i')$. This box must contain the simplex spanned by $\{q_1, \ldots, q_{j+1}\}$, so we have:

$$\frac{1}{((j+1)!)^2} \le 2^{j+1}(\max_{t=1}^{j+1} ||q_t||_2)^j * \max_{t=1}^{j+1} \mathrm{dist}(q_t, f_i') \le (2\Delta m)^{j+1} \max_{t=1}^{j+1} \mathrm{dist}(q_t, f_i').$$

The lemma follows from the above inequality by observing that $\mathrm{cost}(P, F) \ge \max_{t=1}^{j+1} \mathrm{dist}(q_t, f_i') \ge \max_{t=1}^{j+1} \mathrm{dist}(q_t, f_i)$. □

If $\mathrm{cost}(P, F) = 0$ for the optimal $F \in \mathcal{F}'$, then this must be true for some $F' \in \mathcal{F}$ that is contained in V as well. This means that P itself must be contained in V. In this case, such an F' can be found by applying the method of [17] for shape fitting in the L_∞ sense.

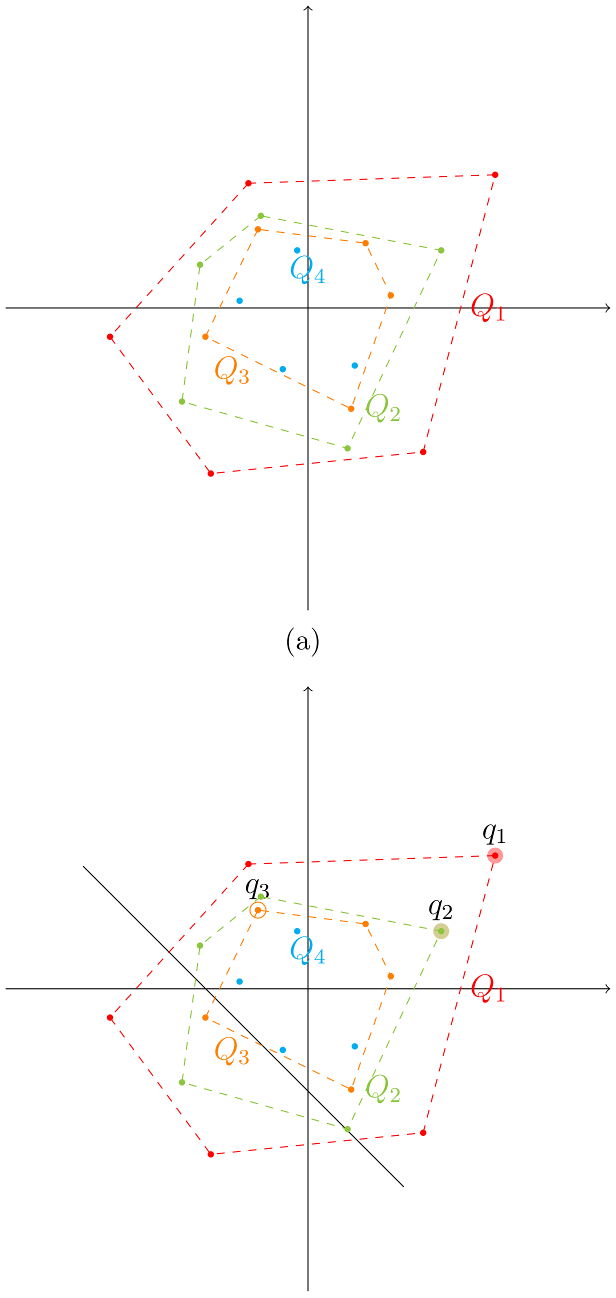
Let us therefore consider the case where $\mathrm{cost}(P, F) > \frac{1}{(m\Delta)^c}$ for the optimal $F \in \mathcal{F}'$. In this case, we express the points in P' in terms of an orthogonal basis for the span of V and b, but round the coordinates of each point in P' to the nearest multiple of $\frac{1}{(mn\Delta)^{c_1}}$ where $c_1 > c$ is a sufficiently large integer. We now scale so that the coordinates of points in P' are integers. Note that the magnitude of the largest coordinate is $(mn\Delta)^{O(1)}$.

Now, treating P' as an input to the integer (j, k) projective clustering problem in $\mathbb{R}^{d'+1}$ we compute a coreset Q using Theorem 4. The running time for this step is $n(\log mn)^{O(1)}$.

4.3.3 Solving the problem on the coreset

We need to find a shape F that is contained in V such that $\text{cost}(Q, F) \leq (1+\epsilon)\text{cost}(Q, F')$ for any shape F' contained in V. Since the size of Q is $(\log mn)^{O(1)}$, we can afford to use a generic polynomial time algorithm for this. We omit the details from this version, and conclude with our main result:

Theorem 5. Let P be an n-point instance of the integer (j, k)-projective clustering problem $(\mathbb{R}^m, \mathcal{F})$ (the largest magnitude of any coordinate for a point in P is at most $(mn)^{10}$), and $\epsilon > 0$ be a parameter. There is a randomized algorithm that runs in time $mn(\log mn)^{O(1)}$ and returns a shape $F \in \mathcal{F}$ such that with constant probability, $\text{cost}(P, F) \leq (1 + \epsilon)\text{cost}(P, F')$ for any $F' \in \mathcal{F}$. Here, j and k are constants but m is not.



(a)

(b)

Figure 4.3. (a) shows each layer of P in the peeling process. Each layer is colored with different colors. Each layer has at most 5 points, so when there are less than 5 points left, the remaining 4 blue points, is considered as a single layer, and the peeling stops. (b) shows that for a point q_3 in the third layer Q_3, its sensitivity cannot exceed 1/3, as there always exist two points, one from Q_1 and one from Q_2, which are further away to the shape than q_3.

CHAPTER 5
FROM LOW DIMENSION TO HIGH DIMENSION

We have obtained an upper bound of total sensitivities for several shape fitting problems via the connection between L_∞ coreset in fixed dimension. In this section, we focus on the high dimensional setting. The dimension d is no longer a constant, and we would like to derive upper bounds of total sensitivities of shape fitting problems which polynomially depend on d, instead of exponentially. The central question in this section is the following:

Problem 4 (Total sensitivity and the dimension d)**.** For projective clustering problems in $\mathbb{R}^d$, such as k-median/k-means, k-line clustering, integer (j,k) clustering, what is the dependence of the total sensitivity function $\mathfrak{S}_n$ on the dimension d? Can we remove the exponential dependence on d in the upper bounds of total sensitivities in Theorem 3 and get upper bounds of $\mathfrak{S}_n$ polynomially depending on d?

The reduction argument in Section 5.1 shows that the factor d can be removed from shape fitting problems where each shape is a low-dimensional object; using this fact, we answer the second question affirmatively in Section 5.2.

This chapter is organized as follows: we first show a dimension reduction argument, which essentially allows us to upper bound the total sensitivity of a high dimensional shape fitting problem with the total sensitivity of a low dimensional problem times a constant. Then we apply this result to several variants of (j,k) projective clustering problem, where j or k are set to specific values, and integer (j,k) projective clustering problem, to obtain small coreset in high dimension.

Problem 4 can be considered as part of a more general question: since total sensitivity quantifies the complexity of shapes, what are the “right” factors to appear in total sensitivity? Is the dependence on d essential or can it be completely removed? The results in this section can be considered as a fairly important step towards the final solution of the more general question. At first glance, it might look like that the occurrence of d in total sensitivity is unavoidable: consider the hyperplane fitting problem, where each shape $F \in \mathcal{F}$ is a hyperplane in $\mathbb{R}^d$. Let P be a point set of size d in general position. Then clearly $\sigma_P(p) = 1$ (since there always exists a hyperplane containing all $d-1$ points other than p), so $\mathfrak{S}_P = d$. However, inspecting the question more carefully, one would notice that there is a difference between the hyperplane fitting problem and (j,k) projective clustering prolems. A hyperplane is a high- dimensional object—each hyperplane is an affine subspace of dimension $d-1$, while for the (j,k) projective clustering problems, the shapes are intrinsically low dimensional: a k-tuple of j-flats is contained in a subspace of dimension at most $k(j+1)$. Hence it seems that $k(j+1)$, instead d, is a more ”correct” factor to appear in the upper bounds of total sensitivities for the variants of (j,k) projective clustering problems.

Indeed this is the case. The fact that the shapes in (j,k) projective clustering are low-dimensional is exploited through two observations. The first observation is that if the *instance* of the shape fitting problem, P, is also low-dimensional, then one can replace the exponential factor of d in the upper bound of the total sensitivity $\mathfrak{S}_P$ with the intrinsic dimension of P and F. For example, consider the k-line clustering problem. Each shape F is a union of k lines, therefore, it is contained in a subspace of dimension at most $2k$ (each line is contained in a 2-subspace, which is a plane). Suppose P is also contained in a line, in particular this means that P is contained in the a subspace of dimension at most 2. Then we can bound the total sensitivity of P by $O(k^{f_1(k)} \log |P|)$ ($f_1(k)$ is a function of k), instead of $O(k^{f_2(d,k)} \log |P|)$ ($f_2(d, k)$ is a function of d and k), as will be shown in later sections. The second observation is that even if P is not low-dimensional, we can project P onto a low dimensional subspace to get P'. The total sensitivity of P is upper bounded by a constant times the total sensitivity of P', which is low-dimensional now. Then we only need to use the first observation on P'.

This reduction immediately produce small coresets in high dimension: the dependence of the size of the coreset on d is only polynomial (which appears in the size of the coreset through the factor $\dim(P)$, which is $O(djk)$ for (j, k) projective clustering problems).

5.1 Dimension reduction

We start with a dimension reduction argument, which shows that for shape fitting problems where each shape is a low-dimensional object, we can upper bound the total sensitivity of an arbitrary point set P with the total sensitivity of a low-dimensional point set. We first define formally the projection of a point set P onto a shape F.

Definition 3 (projection of points on a shape)**.** Define $\mathrm{proj} : \mathbb{R}^d \times \mathcal{F} \to \mathbb{R}^d$, where $\mathrm{proj}\,(p)\,F$ is the projection of p on a shape F, that is, $\mathrm{proj}\,(p, F)$ is a point in F which is nearest to p, $\mathrm{dist}(p, \mathrm{proj}\,(p, F)) = \min_{q \in F} \mathrm{dist}(p, q)$ (ties are broken arbitrarily). We abuse the notation to denote the multi-set $\{\mathrm{proj}\,(p, F)\,|p \in P\}$ by $\mathrm{proj}\,(P, F)$ for $P \subset \mathbb{R}^d$.

See Figure 5.1 for example of projecting a point set on $\mathbb{R}^2$ to a line.

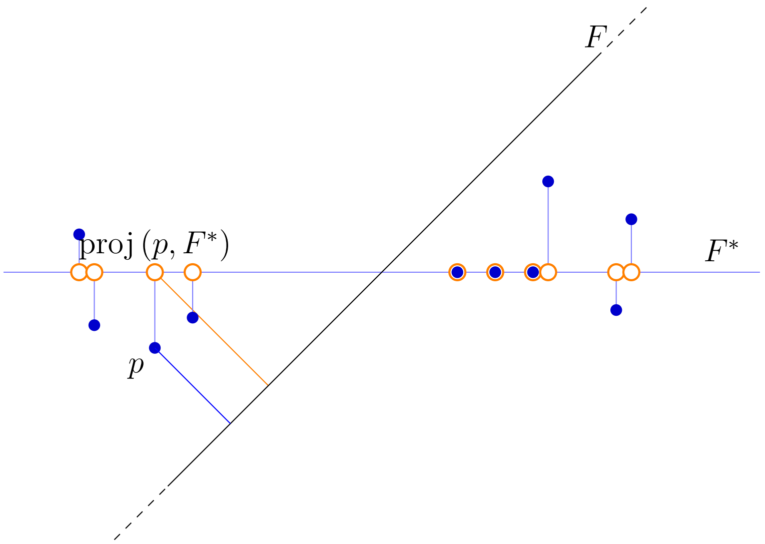


Figure 5.1. The total sensitivity of P is upper bounded by the total sensitivity of $P' = \mathrm{proj}\,(P, \mathcal{F}^*)$ times a constant factor.

Theorem 6 (Dimension reduction, computing the total sensitivity of a point set in high dimensional space with the projected lower dimensional point set)**.** Given an instance P of a shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$. Let F^* denote a shape that minimize $\mathrm{dist}(P, F)$ over all $F \in \mathcal{F}$. Let p' denote $\mathrm{proj}\,(p, F^*)$ and let P' denote $\mathrm{proj}\,(P, F^*)$. Assume that the distance function satisfies the relaxed triangle inequality: $\mathrm{dist}(p, q) \le \alpha(\mathrm{dist}(p, r) + \mathrm{dist}(r, q))$ for any $p, q, r \in \mathbb{R}^d$ for some constant $\alpha \ge 1$. Then

1. the following inequality holds: $\mathfrak{S}_P \le 2\alpha^2 \mathfrak{S}_{P'} + \alpha$.

2. if $\mathrm{dist}(P, F^*) = 0$, then $\sigma_P(p) = \sigma_{P'}(p')$ for each $p \in P$. If $\mathrm{dist}(P, F^*) > 0$, then

$$\sigma_P(p) \le \left(\alpha \frac{\mathrm{dist}(p, p')}{\mathrm{dist}(P, F^*)} + 2\alpha^2 \sigma_{P'}(p')\right). \tag{5.1}$$

Proof. If $\mathrm{dist}(P, F^*) = 0$, then $P = P'$, and clearly both parts of the theorem hold.

Let us consider the case where $\mathrm{dist}(P, F^*) > 0$. By definition,

$$\sigma_P(p) = \inf\{\beta \ge 0 \mid \mathrm{dist}(p, F) \le \beta \mathrm{dist}(P, F), \forall F \in \mathcal{F}\},$$

$$\sigma_{P'}(p') = \inf\{\beta' \ge 0 \mid \mathrm{dist}(p', F) \le \beta' \mathrm{dist}(P', F), \forall F \in \mathcal{F}\}.$$

Let F be an arbitrary shape in $\mathcal{F}$. Then we have

$$\begin{aligned}
\mathrm{dist}(p, F) &\le \alpha \mathrm{dist}(p, p') + \alpha \mathrm{dist}(p', F) \\
&\le \alpha \mathrm{dist}(p, p') + \alpha \sigma_{P'}(p') \mathrm{dist}(P', F) \\
&\le \alpha \mathrm{dist}(p, p') + 2\alpha^2 \sigma_{P'}(p') \mathrm{dist}(P, F) \\
&= \alpha \frac{\mathrm{dist}(p, p')}{\mathrm{dist}(P, F)} \cdot \mathrm{dist}(P, F) + 2\alpha^2 \sigma_{P'}(p') \mathrm{dist}(P, F) \\
&\le \alpha \frac{\mathrm{dist}(p, p')}{\mathrm{dist}(P, F^*)} \cdot \mathrm{dist}(P, F) + 2\alpha^2 \sigma_{P'}(p') \mathrm{dist}(P, F) \\
&= \left(\alpha \frac{\mathrm{dist}(p, p')}{\mathrm{dist}(P, F^*)} + 2\alpha^2 \sigma_{P'}(p')\right) \mathrm{dist}(P, F).
\end{aligned}$$

The first inequality follows from the relaxed triangle inequality, the second inequality follows from the definition of sensitivity of p' in P', and third inequality follows from the fact that

$$\begin{aligned}\mathrm{dist}(P', F) &= \sum_{p' \in P'} \mathrm{dist}(p', F) \leq \sum_{p \in P} \alpha \left(\mathrm{dist}(p, F) + \mathrm{dist}(p, p')\right) \\ &= \alpha(\mathrm{dist}(P, F) + \mathrm{dist}(P, F^*)) \leq 2\alpha \mathrm{dist}(P, F),\end{aligned}$$

since $\mathrm{dist}(P, F^*) \leq \mathrm{dist}(P, F)$.

Thus the second part of the theorem holds. Now,

$$\begin{aligned}\mathfrak{S}_P &= \sum_{p \in P} \sigma_P(p) \\ &\leq \sum_{p \in P} \left(\alpha \frac{\mathrm{dist}(p, p')}{\mathrm{dist}(P, F^*)} + 2\alpha^2 \sigma_{P'}(p') \right) \\ &= \alpha + 2\alpha^2 \mathfrak{S}_{P'}.\end{aligned}$$

□

Note that although the above theorem uses the optimum shape F^* to a point set P (for the shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$), one could use a constant factor approximate solution $\tilde{F}$ instead. Therefore, the result can be easily turned into an algorithm to compute the upper bound of the sensitivity $\sigma_P(p)$ for each $p \in P$. The upper bound of $\mathfrak{S}_P$ is only slightly larger in this case (depending on the constant factor approximation solution):

Corollary 1. Given an instance P of a shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$. Let $\tilde{F}$ denote an c-approximation for fitting P (c is some constant), *i.e.*

$$\mathrm{dist}(P, \tilde{F}) \leq c \min_{F \in \mathcal{F}} \mathrm{dist}(P, F).$$

Let p' denote proj $\left(p, \tilde{F}\right)$ and let P' denote proj $\left(P, \tilde{F}\right)$. Assume that the distance function satisfies the relaxed triangle inequality: $\operatorname{dist}(p,q) \le \alpha(\operatorname{dist}(p,r) + \operatorname{dist}(r,q))$ for any $p, q, r \in \mathbb{R}^d$ for some constant $\alpha \ge 1$. Then

1. the following inequality holds: $\mathfrak{S}_P \le (\alpha^2 + c\alpha)\mathfrak{S}_{\tilde{P}} + \alpha$.

2. if $\operatorname{dist}(P, \tilde{F}) = 0$, then $\sigma_P(p) = \sigma_{P'}(p')$ for each $p \in P$. If $\operatorname{dist}(P, \tilde{P}) > 0$, then

$$\sigma_P(p) \le \left(\alpha \frac{\operatorname{dist}(p, p')}{\operatorname{dist}(P, \tilde{F})} + (\alpha^2 + c\alpha)\sigma_{P'}(p')\right). \tag{5.2}$$

The problem of computing the total sensitivity an instance P of the shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \operatorname{dist})$, where each shape $F \in \mathcal{F}$ is contained in a subspace of dimension m_2, is reduced the problem of computing the total sensitivity of an instance P', where P' is also contained in a subspace of dimension m_2 (by using the the dimension reduction in Theorem 6 or Corollary 1). We now use an important property of the Euclidean distance, which is the rotation-invariant property. This property essentially guarantees that we only need to consider a shape fitting problem $(\mathbb{R}^{2m_2}, \mathcal{F}, \operatorname{dist})$. In particular, consider the (j,k) projective clustering problem. $m_2 = (j+1)k$ since each shape is a union of j-dimensional affine subspace. Fix an arbitrary subspace G of dimension $\min\{d, 2m_2\}$ that contains P'. Then for for any $F \in \mathcal{F}$, there is an $F' \in \mathcal{F}$ such that (a) F' is contained in G, and (b) $\operatorname{dist}(p', F') = \operatorname{dist}(p', F)$ for all $p' \in P'$. We show a "toy" example to further illustrate this. Since it is difficult to show it pictorially when the dimension is larger than 3, we show an toy example, which should convince the reader that it is the case.

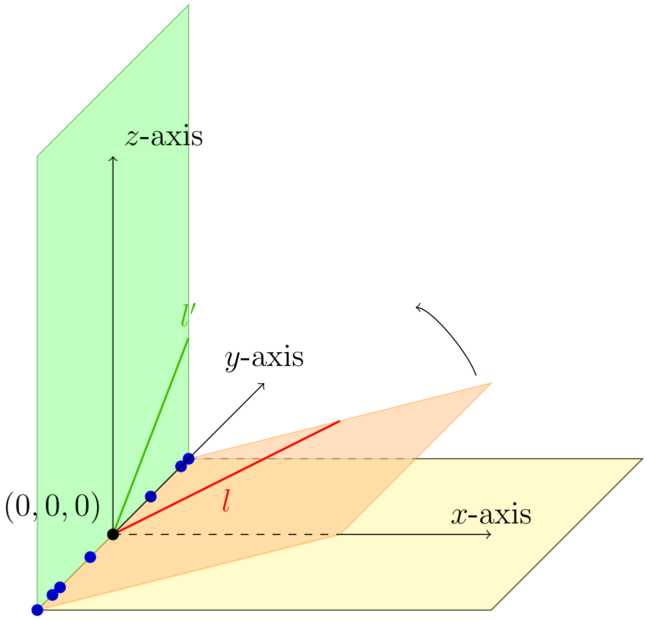


Figure 5.2. An illustration of Theorem 7. The ambient space is $\mathbb{R}^3$. The point set is contained in a line passing through origin, and the family of shapes is the set of lines passing through origin. Hence both the point set and each shape is a low-dimensional object (both are contained in some 1-subspace). The picture shows that the total sensitivity of $(P, \mathcal{F}, \mathbb{R}^d)$ is the same as $(P, \mathcal{F}', \mathbb{R}^2)$, where $\mathcal{F}'$ is the set of lines passing through origin and lie in the plane determined by y-axis and z-axis. For any l that does not lie in the yz-plane, we can always rotate the plane determined by the line and the subspace the point set is from, so that it completely coincides with the yz-plane. The transformation does not change the distances from each point in the point to the shape because of the rotation invariant property of Euclidean space.

Theorem 7 (Sensitivity of a lower dimensional point set in a high dimensional space)**.** Let P' be an n-point instance of the (j,k)-projective clustering problem $(\mathbb{R}^d, \mathcal{F}, \text{dist})$, where dist is the z^{th} power of the Euclidean distance, for some $z \in (0,\infty)$. Assume that P' is contained in a subspace of dimension m_1. (Note that for each shape $F \in \mathcal{F}$, there is a subspace of dimension $m_2 = k(j+1)$ containing it.) Let G be any subspace of dimension $m = \min\{m_1 + m_2, d\}$ containing P'; fix an orthonormal basis for G, and for each $p' \in P'$, let $p'' \in \mathbb{R}^m$ be the coordinates of p' in terms of this basis. Let $P'' = \{p'' \mid p' \in P'\}$, and view P'' as an instance of the (j,k)-projective clustering problem $(\mathbb{R}^m, \mathcal{F}', \text{dist})$, where $\mathcal{F}'$ is the set of all k-tuples of j-subspaces in $\mathbb{R}^m$, and dist is the z^{th} power of the Eucldiean distance. Then, $\sigma_{P'}(p') = \sigma_{P''}(p'')$ for each $p' \in P'$, and $\mathfrak{S}_{P'} = \mathfrak{S}_{P''}$.

Using Corollary 1 and Theorem 7, we get an algorithm to compute $\sigma_P(\cdot)$ (Algorithm 4).

5.2 Coresets in high dimension for a family of (j,k) projective clustering problems

We now describe the result on the total sensitivities of several projective clustering problems in high dimension. Using these upper bounds on total sensitivities and Theorem 1 we also obtain small coresets (in high dimension) for these shape fitting problems.

Algorithm 4: Compute the sensitivities using the dimension reduction technique

```
Input: A point set $P$
Output: Upper bound $s_P(\cdot)$ of the sensitivity of each point in $P$, $\sigma_P(p)$,
        $\forall p \in P$
$\tilde{F} \leftarrow$ ConstantApproximate($P$) ; // ConstantApproximate($P$) computes a
$c$-approximate shape fitting $P$
$P' \leftarrow \text{proj}\left(P, \tilde{F}\right)$;
for $p' \in P'$ do
    $s'_{P'}(p') \leftarrow$ Sensitivity($P'$,$p'$) ;    // Sensitivity($P'$,$p'$) computes an
    upper bound of the sensitivities of each $p' \in P'$ for a shape
    fitting problem $(\mathbb{R}^{2m_2}, \mathcal{F}, \text{dist})$
end
if $dist(P, \tilde{F}) == 0$ then
    $s_P(p) \leftarrow s'_{P'}(p')$;
end
else
    $s_P(p) \leftarrow \left(\alpha\frac{\text{dist}(p,p')}{\text{dist}(P,\tilde{F})} + (\alpha^2 + c\alpha)s_{P'}(p')\right)$;
end
```

5.2.1 k-median/k-means clustering ($(0, k)$ projective clustering)

In this section, we derive upper bound of the total sensitivity function for the k-median/k-means problems, and its generalizations, where the distance function is z^{th} power of Euclidean distance. These bounds are similar to the ones derived by Langberg and Schulman [30], but the proof is much simplified. For the rest of the document, dist is assumed to be the z^{th} power of the Euclidean distance.

Theorem 8 (Total sensitivity of $(0, k)$-projective clustering)**.** Consider the shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \text{dist})$, where $\mathcal{F}$ is the set of all k-point subsets of $\mathbb{R}^d$. We have the following upper bound on the total sensitivity:

$$\begin{aligned} \mathfrak{S}_n &\leq 2^{2z-1}k + 2^{z-1}, & z \geq 1, \\ \mathfrak{S}_n &\leq 2k+1, & z \in (0,1). \end{aligned}$$

In particular, the total sensitivity of the k-median problem (which corresponds to the case when $z = 1$) is at most $2k + 1$, and the total sensitivity of the k-means problem (which corresponds to the case when $z = 2$) is $8k + 2$.

Proof. Let P be an arbitrary n-point set. Apply Theorem 6, and note that $\operatorname{proj}(P, C^*)$, where C^* is an optimum set of k centers, contains at most k distinct points. Assume that $C^* = \{c_1^*, c_2^*, \cdots, c_k^*\}$. Let P_i be the set of points in P whose projection is c_i^*, that is, $P_i = \{p \in P | \operatorname{proj}(p, C^*) = c_i^*\}$. It is easy to see that the summation of sensitivities of the $|P_i|$ copies of c_i^* is at most 1: for any k-point set C in $\mathbb{R}^d$,

$$|P_i| \cdot \frac{\operatorname{dist}(c_i^*, C)}{\operatorname{dist}(C^*, C)} = \frac{|P_i| \operatorname{dist}(c_i^*, C)}{\sum_{j=1}^{k} |P_j| \operatorname{dist}(c_j^*, C)} \leq 1.$$

Therefore, the total sensitivity of $\operatorname{proj}(P, C^*)$ is at most k. Substituting α from the remark after Theorem 6, we get the above result. □

Theorem 9 (ϵ-coreset for $(0, k)$-projective clustering)**.** Consider the shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \operatorname{dist})$, where $\mathcal{F}$ is the set of all k-point subsets of $\mathbb{R}^d$. For any n-point instance P, there is an ϵ-coreset of size $O(k^3 d \epsilon^{-2})$.

Proof. Observe that the $\dim(P)$ is $O(kd)$. Using Theorem 1, and Theorem 8, we obtain the above result. □

5.2.2 k-line clustering ($(1, k)$ projective clustering)

In this section, we derive upper bounds on the total sensitivity function for the k-line clustering problem, that is, the $(1, k)$-projective clustering problem.

Theorem 10 (Total sensitivity for k-line clustering problem)**.** Consider the shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$, where $\mathcal{F}$ is the set of k-tuple of lines. The total sensitivity function, $\mathfrak{S}_n$, is $O(k^{f(k)} \log n)$, where $f(k)$ is a function the depends only on k.

Proof. Let P be an arbitrary n-point set. Let K^* denote an optimum set of k lines fitting P. Using Theorems 6 and 7, it suffices to bound the sensitivity of an n-point instance of a k-line clustering problem housed in $\mathbb{R}^{4k}$. By Theorem 3, the total sensitivity of this latter shape fitting problem is $O(k^{f(k)} \log n)$, where $f(k)$ is a function depending only on k. Therefore, $\mathfrak{S}_n$ is $O(k^{f(k)} \log n)$.

(Alternatively, one could use a recent result in [23]. Let P' denote the projection of P into K^*. Since K^* is a union of k lines, we can upper bound the sensitivity of P' by k times the sensitivity of an n-point set that lies on a *single line.* The sensitivity of an n-point set that lies on a single line can be upper bounded by the sensitivity of an n-point set for the *weighted* $(0, k)$-projective clustering problem, for which the sensitivity bound is $k^{f(k)} \log n$ as shown in [23].) □

Notice that for k-line clustering problem, the bound on the total sensitivity depends logarithmically on n. We give below a construction of a point set that shows that this is necessary, even for $d = 2$.

Theorem 11 (The upper bound of total sensitivity for k-line clustering problem is tight)**.** For every $n \geq 2$, there exists an n-point instance of the k-line clustering problem $(\mathbb{R}^2, \mathcal{F}, \mathrm{dist})$, where dist is the Euclidean distance, such that the total sensitivity of P is $\Omega(\log n)$.

Proof. We construct a point set P of size n, together with n shapes $F_i \in \mathcal{F}$, $i = 1, \cdots, n$, such that $\sum_{i=1}^{n} \mathrm{dist}(p_i, F_i)/\mathrm{dist}(P, F_i)$ is $\Omega(\log n)$. Note that this implies that $\mathfrak{S}_P$ is at least $\Omega(\log n)$. Let P be the following point set in $\mathbb{R}^2$: $p_i = (1/2^{i-1}, 0)$, for $i = 1, \cdots, n$. Let F_i be a pair of lines: one vertical line and one horizontal line, where the vertical line is the y-axis, and the horizontal line is $\{(x, 1/2^i) | x \in \mathbb{R}\}$.

Consider the point p_i, where $i = 1, \cdots, n$. We show that $\mathrm{dist}(p_i, F_i)/\mathrm{dist}(P, F_i)$ is at least $1/(2+i)$, for $i = 1, \cdots, n$. For $j \leq i$, note that $\mathrm{dist}(p_j, F_i) = 1/2^i$: since the distance from p_j to the horizontal line in F_i is $1/2^i$ and the distance to the vertical line is $1/2^{j-1}$, $\mathrm{dist}(p_j, F_i) = \min\{1/2^{j-1}, 1/2^i\} = 1/2^i$. For $i + 1 \leq j \leq n$, on the other hand, $\mathrm{dist}(p_j, F_i) = 1/2^{j-1}$. Therefore, $\sum_{j=i+1}^{n} \mathrm{dist}(p_j, F_i) = \sum_{j=i+1}^{n} 1/2^{j-1} = (1/2^{i-1}) \cdot (1 - (1/2)^{n-i})$. Thus, we have

$$\sigma_P(p_i) = \sup_{F \in \mathcal{F}} \frac{\mathrm{dist}(p_i, F)}{\mathrm{dist}(P, F)} \geq \frac{\mathrm{dist}(p_i, F_i)}{\mathrm{dist}(P, F_i)} = \frac{1/2^i}{(1/2^{i-1} - 1/2^{n-1}) + i \cdot (1/2^i)} > \frac{1}{2+i}$$

Therefore, $\mathfrak{S}_P \geq \sum_{i=1}^{n} \sigma_P(p_i) > \sum_{i=1}^{n} \frac{1}{2+i}$, which is $\Omega(\log n)$. □

Theorem 12 (ϵ-coreset for k-line clustering problem)**.** Consider the shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$, where $\mathcal{F}$ is the set of all k-tuples of lines in $\mathbb{R}^d$. For any n-point instance P, there is an ϵ-coreset with size $O(k^{f(k)} d (\log n)^2/\epsilon^2)$.

Proof. This result follows from Theorem 10, Theorem 1, and the fact that $\dim(P)$ in this case is $O(kd)$. □

5.2.3 j-subspace approximation (($j, 1$) projective clustering)

In this section, we derive upper bounds on the sensitivity of the subspace approximation problem, that is, the $(j, 1)$-projective clustering problem. For the applications of Theorems 6 and 7 in the other sections, we use existing bounds on the sensitivity that have a dependence on the dimension d. For the subspace approximation problem, however, we derive here the dimension-dependent bounds on sensitivity by generalizing an argument from [30] for the case $j = d - 1$ and $z = 2$. This derivation is somewhat technical. With these bounds in hand, the derivation of the dimension-independent bounds is readily accomplished in a manner similar to the other sections.

Dimension-dependent bounds on Sensitivity

We first recall the notion of an (α, β, z)-*conditioned basis* from [14], and state one of its properties (Lemma 2). We will use standard matrix terminology: m_{ij} denotes the entry in the i-th row and j-th column of M, and $M_{i\cdot}$ is the i-th row of M.

Definition 4. Let M be an $n \times m$ matrix of rank ρ. Let $z \in [1, \infty)$, and $\alpha, \beta \geq 1$. An $n \times \rho$ matrix A is an (α, β, z)-conditioned basis for M if the column vectors of A span the column space of M, and additionally A satisfies that: (1) $\sum_{i,j} |a_{ij}|^z \leq \alpha^z$, (2) for all $u \in \mathbb{R}^\rho$, $\| u \|_{z'} \leq \beta \| Au \|_z$, where $\| \cdot \|_{z'}$ is the dual norm for $\| \cdot \|_z$ (*i.e.* $1/z + 1/z' = 1$).

Lemma 2. Let M be an $n \times m$ matrix of rank ρ. Let $z \in [1, \infty)$. Let A be an (α, β, z)-conditioned basis for M. For every vector $u \in \mathbb{R}^m$, the following inequality holds:

$$|M_{i\cdot} u|^z \leq (\| A_{i\cdot} \|_z^z \cdot \beta^z) \| Mu \|_z^z$$

Proof. We have $M = A\tau$ for some $\rho \times m$ matrix τ. Then,

$$|M_{i\cdot} u|^z = |A_{i\cdot} \tau u|^z \leq \| A_{i\cdot} \|_z^z \cdot \| \tau u \|_{z'}^z \leq \| A_{i\cdot} \|_z^z \cdot \beta^z \| A\tau u \|_z^z = \| A_{i\cdot} \|_z^z \cdot \beta^z \| Mu \|_z^z.$$

The second step is Holder's inequality, and the third uses the fact that A is (α, β, z)-conditioned. □

Using Lemma 2, we derive an upper bound on the total sensitivity when each shape is a hyperplane.

Lemma 3 (total sensitivity for fitting a hyperplane)**.** Consider the shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \text{dist})$ where $\mathcal{F}$ is the set of all $(d-1)$-flats, that is, hyperplanes. The total sensitivity of any n-point set is $O(d^{1+z/2})$ for $1 \leq z < 2$, $O(d)$ for $z = 2$, and $O(d^z)$ for $z > 2$.

Proof. We can parameterize a hyperplane with a vector in $\mathbb{R}^{d+1}$, $u = \begin{bmatrix} u_1 & \cdots & u_{d+1} \end{bmatrix}^T$: the hyperplane determined by u is $h_u = \{x \in \mathbb{R}^d | \sum_{i=1}^d u_i x_i + u_{d+1} = 0\}$, where x_i denotes the i^{th} entry of the vector x. Without loss of generality, we may assume that $\sum_{i=1}^d u_i^2 = 1$. The Euclidean distance to h_u from a point $q \in \mathbb{R}^d$ is

$$\text{dist}(q, h_u) = \frac{\left| \sum_{i=1}^d u_i q_i + u_{d+1} \right|}{\sqrt{\sum_{i=1}^d u_i^2}} = \left| \sum_{i=1}^d u_i q_i + u_{d+1} \right|.$$

(the second equality follows from the assumption that $\sum_{i=1}^{d} u_i^2 = 1$.)

Let $P = \{p_1, p_2, \ldots, p_n\} \subseteq \mathbb{R}^d$ be any set of n points. Let $\tilde{p_i}$ denote the row vector $\begin{bmatrix} p_i^T & 1 \end{bmatrix}$, and let M be the $n \times (d+1)$ matrix whose i^{th} row is $\tilde{p_i}$. Then, $\operatorname{dist}(p_i, h_u) = |M_{i\cdot} u|^z$, and $\operatorname{dist}(P, h_u) = \sum_{i=1}^{n} |M_{i\cdot} u|^z = \| Mu \|_z^z$. Then using Lemma 2, we have

$$\sigma_P(p_i) = \sup_u \frac{|M_{i\cdot} u|^z}{\| Mu \|_z^z} \le \| A_{i\cdot} \|_z^z \cdot \beta^z,$$

where A is an (α, β, z)-conditioned basis for M. Thus,

$$\mathfrak{S}_P = \sum_{i=1}^{n} \sigma_P(p_i) \le \beta^z \sum_{i=1}^{n} \| A_{i\cdot} \|_z^z = \beta^z \sum_{i,j} |a_{ij}|^z = (\alpha\beta)^z.$$

For $1 \le z < 2$, M has $((d+1)^{1/z+1/2}, 1, z)$-conditioned basis; for $z = 2$, M has $((d+1)^{1/2}, 1, z)$-conditioned basis; for $z > 2$, M has $((d+1)^{1/z+1/2}, (d+1)^{1/z'-1/2}, z)$-conditioned basis [14]. Thus the total sensitivity for the three cases are $(d+1)^{1+z/2}$, $d+1$, and $(d+1)^z$, respectively. □

It is now easy to derive dimension-dependent bounds on the sensitivity when each shape is a j-subspace.

Corollary 2 (Total sensitivity for fitting a j-subspace)**.** Consider the shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \operatorname{dist})$ where $\mathcal{F}$ is the set of all j-flats. The total sensitivity of any n-point set is $O(d^{1+z/2})$ for $1 \le z < 2$, $O(d)$ for $z = 2$, and $O(d^z)$ for $z > 2$.

Proof. Denote $\mathcal{F}'$ the set of hyperplanes in $\mathbb{R}^d$. Let $P \subseteq \mathbb{R}^d$ be an arbitrary n-point set. We first show that $\sigma_{P,\mathcal{F}}(p) \le \sigma_{P,\mathcal{F}'}(p)$, where the additional subscript is being used to indicate which shape fitting problem we are talking about (hyperplanes or

j-flats). Let p be an arbitrary point in P. Let $F_p \in \mathcal{F}$ denote the j-subspace such that $\sigma_{P,\mathcal{F}}(p) = \mathrm{dist}(p, F_p)/\mathrm{dist}(P, F_p)$. Let proj (p) F_p denote the projection of p on F_p. Consider the hyperplane F' containing F_p and orthogonal to the vector $p -$proj (p) F_p. We have $\mathrm{dist}(p, F') = \mathrm{dist}(p, F_p)$, whereas $\mathrm{dist}(q, F') \leq \mathrm{dist}(q, F_p)$ for each $q \in P$. Therefore,

$$\sigma_{P,\mathcal{F}'}(p) \geq \mathrm{dist}(p, F')/\mathrm{dist}(P, F') \geq \mathrm{dist}(p, F_p)/\mathrm{dist}(P, F_p) = \sigma_{P,\mathcal{F}}(p).$$

It follows that $\mathfrak{S}_{P,\mathcal{F}} \leq \mathfrak{S}_{P,\mathcal{F}'}$. The statement in the corollary now follows from Lemma 3. □

Dimension-independent Bounds on the Sensitivity

We now derive dimension-independent upper bounds for the total sensitivity the for j-subspace fitting problem.

Theorem 13 (Total sensitivity for j-subspace fitting problem)**.** Consider the shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$ where $\mathcal{F}$ is the set of all j-flats. The total sensitivity of any n-point set is $O(j^{1+z/2})$ for $1 \leq z < 2$, $O(j)$ for $z = 2$, and $O(j^z)$ for $z > 2$.

Proof. Use Theorem 6, note that the projected point set P' is contained in a j-subspace. Further, each shape is a j-subspace. So, applying Theorem 7 and Corollary 2, the total sensitivity is $O(j^{2+z/2})$ or $z \in [1, 2)$, $O(j)$ for $z = 2$ and $O(j^z)$ for $z > 2$. □

Using Theorem 13 and the fact that dim (P) for the j-subspace fitting problem is $O(jd)$, we obtain small ϵ-coresets:

Theorem 14 (ϵ-coreset for j-subspace fitting problem)**.** Consider the shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$ where $\mathcal{F}$ is the set of all j-flats. For any n-point set, there exists an ϵ-coreset whose size is $O(j^{3+z}d\epsilon^{-2})$ for $z \in [1, 2)$, $O(j^3 d\epsilon^{-2})$ for $z = 2$ and $O(j^{2z+1}d\epsilon^{-2})$ for $z \geq 2$.

Proof. The result follows from Theorem 13, and Theorem 1. □

We note that for the case $j = d-1$ and $z = 2$, a linear algebraic result from [10] yields a coreset whose size is an improved $O(d\epsilon^{-2})$.

5.2.4 integer (j, k) projective clustering

Theorem 15. Consider the shape fitting problem $(\mathbb{R}^d, \mathcal{F}, \mathrm{dist})$, where $\mathcal{F}$ is the set of k-tuples of j-flats. Let $P \subset \mathbb{R}^d$ be any n-point instance with integer coordinates, the magnitude of each coordinate being at most n^c, for some constant c. The total sensitivity $\mathfrak{S}_P$ of P is $O((\log n)^{f(k,j)})$, where $f(k, j)$ is a function of only k and j. There exists an ϵ-coreset for P of size $O((\log n)^{2f(k,j)}kjd\epsilon^{-2})$.

Proof. Observe that the projected point set $P' = \mathrm{proj}\,(P)\,\{J_1^*, \cdots, J_k^*\}$, where $\{J_1^*, \cdots, J_k^*\}$ is an optimum k-tuple of j-flats fitting P, is contained in a subspace of dimension $O(jk)$. Using Theorem 3, Theorem 7, and Theorem 6, the total sensitivity $\mathfrak{S}_P$ is upper bouned by $O((\log n)^{f(k,j)})$, where $f(k, j)$ is a function of k and j. (A technical complication is that the coordinates of P', in the appropriate orthonormal basis, may not be integers. This is addressed in Section 3.3.2.)

Using Theorem 1 and the fact that $\dim(P)$ is $O(djk)$, we obtain the bound on the coreset. □

CHAPTER 6

FROM ϵ-APPROXIMATION OF RANGE SPACES TO ϵ-CORESET OF SHAPE FITTING PROBLEMS

In this chapter, we describe a result from [19], which connects ϵ-coreset with ϵ-approximation of range spaces. Using a fundamental result in range spaces (Theorem 16), the construction of ϵ-coreset boils down to a construction of ϵ-approximation of a special range space, induced by the shape fitting problem at hand. Not only the routine analysis to determine the number of samples in the construction of coresets is removed, one also get smaller ϵ-coreset.

We first provide some necessary notions and results in ϵ-approximation of range spaces. We start by defining range spaces:

Definition 5 (Range spaces and the dimension of a range space)**.** A range space is a pair $(U, \mathcal{R})$, where U is a finite set, and $\mathcal{R}$ is a family of subsets of U. The dimension of a range space $(U, \mathcal{R})$, $\dim(U, \mathcal{R})$, is the smallest positive integer m, such that for any $U' \subset U$ of cardinality at least 2, it holds that

$$|\{U' \cap R | R \in \mathcal{R}\}| \leq |U'|^m.$$

$\dim(U, \mathcal{R})$ is closely related to the Vapnik-Chervonenkis dimension (VC-dimension) of the range space $(U, \mathcal{R})$; in fact, one can show that m is roughly no greater than the VC-dimension of $(U, \mathcal{R})$.

For a toy example of a range space, let $U = e_1, e_2, e_3$, and let $R_1 = \{e_1\}, R_2 = \{e_2, e_3\}, R_3 = e_1, e_3$, the pair $(U, \{R_1, R_2, R_3\})$ is a range space.

Definition 6 (ϵ-approximation of a range space)**.** An ϵ-approximation of a range space $(U, \mathcal{R})$ is a subset of U, such that

$$\left| \frac{|R|}{|U|} - \frac{|S \cap R|}{|S|} \right| \leq \epsilon, \forall R \in \mathcal{R}. \tag{6.1}$$

A central result regarding ϵ-approximation of range spaces is the following:

Theorem 16 (ϵ-approximation of range spaces)**.** Let $(U, \mathcal{R})$ be a range space. The dimension of $(U, \mathcal{R})$ is denoted $\dim(U, \mathcal{R})$. Let ϵ, δ be real numbers in $(0, 1)$. Let S be a sample of

$$|S| = \frac{c}{\epsilon^2} \left(\dim(U, \mathcal{R}) + \log \frac{1}{\delta} \right)$$

i.i.d. elements from U, where c is a sufficiently large constant. Then with probability at least $1 - \delta$, S is an ϵ-approximation of $(U, \mathcal{R})$ (that is, S satisfies the inequality (6.1)).

A crucial component in the connection between ϵ-coresets of shape fitting problems and ϵ-approximation of range spaces is a special range space defined in the following way:

Definition 7 (range space of a shape fitting problem)**.** Given an instance P of a shape fitting problem. For each $p \in P$, let A_p be a multi-set of $m_p = \lceil n\sigma_P(p) \rceil$ copies of the the point p, each copies has weight $1/m_p$ (one can image that each light-weight copy of p is obtained by cutting a "whole" point p into m_p small pieces).

Let $\tilde{P} = \cup_{p \in P} A_p$, which is a multi-set consisting of m_p copies of (light-weight) $p \in P$. Let $\mathcal{R}(P) := \{R_{F,r} | F \in \mathcal{F}, r \geq 0\}$, where each $R_{F,r}$ is

$$R_{F,r} := \{p \in P | (1/m_p) \mathrm{dist}(p, F) \leq r\}.$$

Note that although $R_{F,r}$ is a subset of P, we can also viewed it as a subset of the weighted point set $\tilde{P}$: if p is in $R_{F,r}$, then replace p with m_p copies of the light-weight copies of p, where each copy has weight $1/m_p$. We denote the obtained subsets of $\tilde{P}$ by $\tilde{R}_{F,r}$. Therefore, the range space $(P, \{R_{F,r} | F \in \mathcal{F}, r \geq 0\})$ can be easily transformed into a range space $(\tilde{P}, \{\tilde{R}_{F,r} | F \in \mathcal{F}, r \geq 0\})$.

Feldman and Langberg [19] proved the following result:

Theorem 17. Let P be a weighted point set, where each point p has weight w_p. Let $\mathcal{F}$ denote a family of shapes. Let dist denote the distance function. If a subset $S \subset P$ satisfies that

$$\left| \frac{|\{p \in P | w_p \mathrm{dist}(p, F) \leq r\}|}{|P|} - \frac{|\{p \in S | w_p \mathrm{dist}(p, F) \leq r\}|}{|S|} \right| \leq \frac{\epsilon}{5}, \forall F \in \mathcal{F}, r \geq 0, \tag{6.2}$$

then the (weighted) set S also satisfies that

$$\left| \frac{\sum_{p \in P : w_p \mathrm{dist}(p,F) \leq r} w_p \mathrm{dist}(p, F)}{|P|} - \frac{\sum_{p \in S : w_p \mathrm{dist}(p,F) \leq r} w_p \mathrm{dist}(p, F)}{|S|} \right| \leq \epsilon r, \forall F \in \mathcal{F}, r \geq 0. \tag{6.3}$$

If we apply the above theorem to the weighted point set $\tilde{P}$, then it turns out that by assigning weights to the point in S properly, we have already obtained

ϵ-coreset: let $r = (1/n)\sum_{p\in P}\operatorname{dist}(p,F)$. Notice that each point in $\tilde{P}$ satisfies that

$$\frac{1}{m_p}\operatorname{dist}(p,F) \le \frac{1}{n}\left(\frac{1}{\sigma_P(p)}\operatorname{dist}(p,F)\right) \le \frac{1}{n}\sum_{p\in P}\operatorname{dist}(p,F).$$

Therefore, the set $\{p \in \tilde{P} | w_p \operatorname{dist}(p,F) \le r\}$ is exactly $\tilde{P}$. Further, notice that for any shape F, the quantity $\sum_{p\in P}\operatorname{dist}(p,F)$ is exactly $\sum_{p\in\tilde{P}} w_p \operatorname{dist}(p,F)$, since $\sum_{p\in\tilde{P}} w_p\operatorname{dist}(p,F) = \sum_{p\in P} m_p \cdot (1/m_p)\operatorname{dist}(p,F)$. Therefore, multiplying Eq 6.3 by $\left|\tilde{P}\right| = \sum_{p\in P} m_p$, we get

$$\left|\sum_{p\in P}\operatorname{dist}(p,F) - \frac{\sum_{p\in P} m_p}{|S|}\sum_{p\in S}\frac{1}{m_p}\operatorname{dist}(p,F)\right| \le \epsilon\cdot\left(\frac{1}{n}\sum_{p\in P}\operatorname{dist}(p,F)\right)\cdot\left(\sum_{p\in P} m_p\right),$$

which is

$$\left|\sum_{p\in P}\operatorname{dist}(p,F) - \frac{1}{|S|}\sum_{p\in S}\frac{\sum_{p\in P} m_p}{m_p}\operatorname{dist}(p,F)\right| \le \epsilon(\mathfrak{S}_P+1)\sum_{p\in P}\operatorname{dist}(p,F).$$

Therefore, the weighted set S, where each point p in S has weight $(\sum_{p\in P} m_p)/(m_p|S|)$ (which is roughly $\mathfrak{S}_P/(|S|\sigma_P(p))$) is an $(\mathfrak{S}_P+1)\epsilon$ coreset of P with respect to the family $\mathcal{F}$ of shapes. However, the S is in fact much easier to compute, as it is nothing but an ϵ-approximation of the range space $(\tilde{P}, \{\tilde{R}_{F,r} | F \in \mathcal{F}, r \ge 0\})$, and we already have Theorem 16 to compute it.

ϵ-approximation of a range space can be computed via uniform random sampling 16. Notice that if one performs a uniform sampling on $\tilde{R}$, it is the same as sampling points in P, where the probability that $p \in P$ is picked is $m_p/\sum_{p\in P} m_p$, as $\tilde{P}$ contains m_p copies of p, for each $p \in P$. Therefore, the connection between the uniform sampling of $\tilde{R}$ and the sampling scheme in Algorithm 2 is clear:

$$\frac{m_p}{\sum_{p\in P} m_p} = \frac{\lceil n\sigma_P(p)\rceil}{\sum_{p\in P}\lceil n\sigma_P(p)\rceil} \approx \frac{\sigma_P(p)}{\sum_{p\in P}\sigma_P(p)}.$$

The size of the the coreset can also be derived from Theorem 16: in order to get an ϵ-coreset, we need to compute an $\epsilon/(5\mathfrak{S}_P + 5)$-approximation of the range space $(\tilde{R}, \{\tilde{R}_{F,r} | F \in \mathcal{F}, r \geq 0\})$. Using Theorem 16, in order to get an $\epsilon/(5\mathfrak{S}_P + 5)$-approximation of the range space with probability at least $1 - \delta$, we need to draw $O((\mathfrak{S}_P/\epsilon)^2(\dim(P) + \log(1/\delta)))$ samples, where $\dim(R)$ is the dimension of the range space $(\tilde{R}, \{\tilde{R}_{F,r} | F \in \mathcal{F}, r \geq 0\})$. This is precisely Theorem 1.

CHAPTER 7

L_1 CIRCLE FITTING

In this chapter, we focus on the problem of L_1 circle fitting:

Problem 5 (L_1 circle fitting on the plane)**.** Given a finite set of points $P \subset \mathbb{R}^2$, compute an optimal circle, $C(x, y, r)$, which denotes a circle with center (x, y) and radius r, minimizing the summation of distances from points in P to the circle:

$$\sum_{p \in P} \left| \sqrt{(p_x - x)^2 + (p_y - y)^2} - r \right|$$

This question is motivated by the problem of measuring circularity (or roundness) in computational metrology[43][25]: there one needs to verify that a manufactured object is ”close enough” to the ideal shape, which is a disk, in this case. This problem is referred as circularity test in computational metrology: if the object meets the tolerance specification, accept it, otherwise reject. The common approach is to sample some points from the surface (or boundary) of the object, and the circularity/roundness of the object is quantified by the minimum width of an annulus enclosing all the sampled points. Figure 7.1 and Figure 7.2 shows an example.

As mentioned earlier, for any shape fitting problem, we can consider L_∞ and L_1 fitting (or generally L_p fitting) depending on how we define the objective function that we want to optimize. For circle fitting problem, for any circle C, for a given point set P, we have the following vector, encoding the distance from each point in

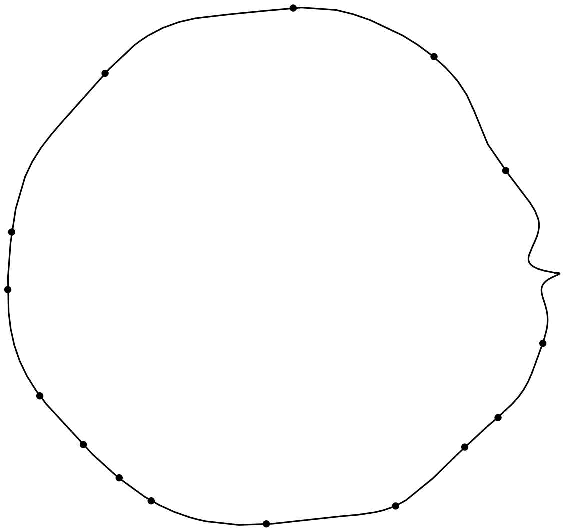


Figure 7.1. Sample points from a manufactured object

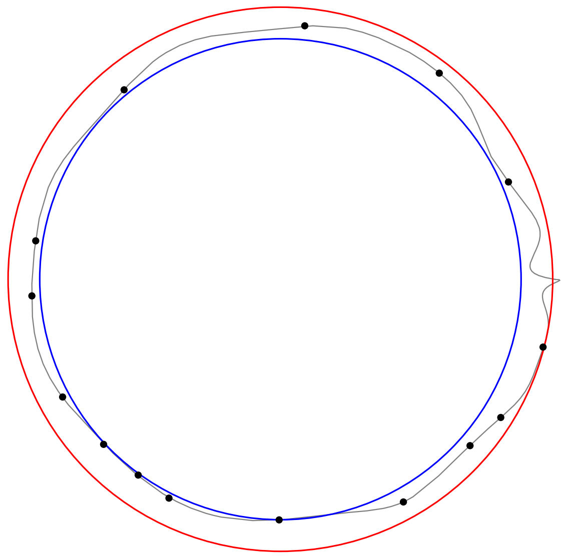


Figure 7.2. Finding the minimum width annulus enclosing all the points

P to the circle:

$$\begin{bmatrix}\text{dist}(p_1, C) & \text{dist}(p_2, C) & \cdots & \text{dist}(p_n, C)\end{bmatrix}$$

If we define the circularity, that is, the measure that how closely a set of points approximates a circle, as the width of a narrowest annulus enclosing P, then we are trying to optimize the L_∞ norm of the vector above, that is,

$$\| \begin{bmatrix}\text{dist}(p_1, C) & \text{dist}(p_2, C) & \cdots & \text{dist}(p_n, C)\end{bmatrix} \|_\infty = \max_{p_i \in P} \text{dist}(p_i, C)$$

If we define the circularity as the summation of all the distances from the points to the circle, then we are trying to optimize the L_1 norm:

$$\| \begin{bmatrix}\text{dist}(p_1, C) & \text{dist}(p_2, C) & \cdots & \text{dist}(p_n, C)\end{bmatrix} \|_1 = \sum_{p_i \in P} \text{dist}(p_i, C)$$

See Figure 7.3 for an example. Compared with L_∞ norm, L_1 norm is less sensitive to noise/outliers and hence more robust. Also, this problem is interesting in the sense that the shape we consider here, is quite different from families of shapes that appeared in problems such as subspace approximation, k-median/k-means clustering, projective (j, k) clustering, etc.

Since exact algorithm for L_1 circle fitting is expensive, we are interested in approximation algorithms that are sublinear. In [28], a linear approximation algorithm for L_1 circle fitting is proposed. The method there has a similar flavor to the core-set approach, however, it was left as an open problem whether it is possible to obtain small coreset for circle fitting problem. We answer this question affirmatively: by upper bounding the total sensitivity of the circle fitting problem, we are able to get

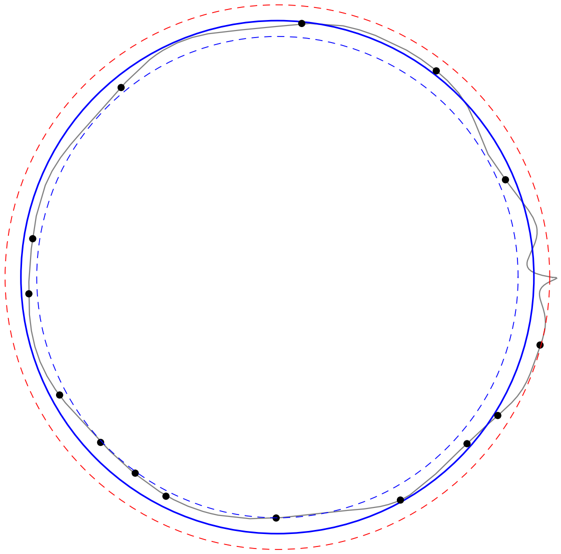


Figure 7.3. L_1 fitting of points sampled from the boundary of an object

coreset of size roughly $O((\log n/\epsilon)^2)$. Moreover, the coreset has the desirable properties that the points in the coreset are from the input point set, and the weights of points in the coreset are non-negative.

7.1 Circle fitting problem has small coreset

We have mentioned the results regarding circle fitting in Theorem 3 in Chapter 4. We include a separate statement regarding the coreset size of circle fitting problem in this section for the sake of completeness.

Theorem 18 (small coreset for circle fitting)**.** Given a point set of size n on the plane, for the circle fitting problem there exists an ϵ-coreset of size $O((\log n/\epsilon)^2)$. With probability at least $1-\delta$, we can compute in $O(n(\log n)^{O(1)} + (\log n/\epsilon)^2(O(1) + \log(1/\delta))$ time an ϵ-coreset of size $O((\log n/\epsilon)^2(O(1) + \log(1/\delta))$.

7.2 $\Omega(\log n)$ lower bound of the size of coreset

In this section, we show a lower bound on the size of coreset for circle fitting problems.

Theorem 19. There exists a point set $P \subset \mathbb{R}^2$ of size n, such that any 1/100-coreset of P for the circle fitting problem has size at least $\Omega(\log n)$.

This theorem implies that for the point set P in the theorem above, for any $\epsilon \in (0, \frac{1}{100})$, the ϵ-coreset of P has size at least $\Omega(\log n)$, since any ϵ-coreset is also an ϵ'-coreset of P, if $\epsilon \leq \epsilon'$, by the definition of coreset.

Construction of the point set P**:** For simplicity, assume that $n = (3^{N+1} - 1)/2$, for some N. (If n is not such a number, the result would only change by a constant factor.) The point set P consists of $N+1$ groups: the i^{th} group, denoted by P_i, contains 3^{N-i} copies of the point $p_i = (2^i, 0)$, $i = 0, \cdots, N$. We prove Theorem 19 by establishing the following three lemmas:

Lemma 7.2.1 (Each P_i contributes a significant portion to the total fitting cost)**.** For each P_i, $i = 0, \cdots, N$, there exists a circle C_i, such that

$$\frac{\text{dist}(P_i, C_i)}{\text{dist}(P, C_i)} \geq \frac{1}{18}. \tag{7.1}$$

Lemma 7.2.2 (Each p_i cannot be too heavy in an 1/100-coreset)**.** Let S be an 1/100-coreset of P. For each p_i, in the coreset S,

$$w(p_i) \leq 19|P_i|. \tag{7.2}$$

Lemma 7.2.3 (Any consecutive chunk cannot be totally omitted in an 1/100-coreset)**.** Let l be a sufficiently large number (we will determine the value of l in the proof). Partition the sequence $p_0, \cdots, p_N$, which are points in P, into chunks of length $2l+1$:

$$\underbrace{p_0, \cdots, p_{2l}}_{\text{first chunk } CH_1}, \underbrace{p_{2l+1}, \cdots, p_{4l+1}}_{\text{second chunk } CH_2}, \cdots, \underbrace{\cdots, p_N}_{\approx (N/2l)^{\text{th}} chunk}$$

For S to be an 1/100-coreset of P,

$$S \cap CH_i \neq \emptyset, \text{for every chunk.}$$

Theorem 19 follows from Lemma 7.2.3 immediately. We describe the proof for Theoerem 19 below:

Proof. There are $N/(2l) = \Theta(\log n)$ chunks, and the point sets of the chunks are disjoint. For S to be an 1/100-coreset of P, S needs to include at least one point from each chunk, hence it has $\Omega(\log n)$ distinct points. □

From Lemma 7.2.1, we also get a corollary, which says that the upper bound of the total sensitivity for n-point sets, $O(\log n)$, is tight:

Corollary 3. The total sensitivity of circle fitting problem, $\mathfrak{S}_n$, is $\Theta(\log n)$.

The proof of the corollary above follows from the definition of sensitivity Definition 3, Lemma 7.2.1 and Theorem 3.

Proof. The summation of sensitivities of point in P_i in P, by definition, is

$$\sum_{p_i \in P_i} \sigma_{p_i}(P) = \sum_{p_i \in P_i} \sup_{C(x,y,r)} \frac{\text{dist}(p_i, C_{x,y,r})}{\text{dist}(P, C_{x,y,r})} \geq \sum_{p_i \in P_i} \frac{\text{dist}(p_i, C_i)}{\text{dist}(P, C_i)} = \frac{\text{dist}(P_i, C_i)}{\text{dist}(P, C_i)} \geq \frac{1}{18}$$

In other words, the summation of sensitivities of each group of points P_i is at least $1/18$. Since there are $N+1$ groups, the total sensitivity of P, is $(N+1)/18 = \Omega(\log n)$. Theorem 3 says that $\mathfrak{S}_n$ is $O(\log n)$, therefore, $\mathfrak{S}_n = \Theta(\log n)$. □

In the rest of this section, we prove lemma 7.2.1, Lemma 7.2.2 and Lemma 7.2.3.

7.2.1 Each P_i contributes significant portion to the fitting cost

We recall lemma 7.2.1:

Lemma 7.2.1 (Each P_i contributes a significant portion to the total fitting cost)**.** For each P_i, $i = 0, \cdots, N$, there exists a circle C_i, such that

$$\frac{\text{dist}(P_i, C_i)}{\text{dist}(P, C_i)} \geq \frac{1}{18}. \tag{7.1}$$

We first show that we can approximate the distance function, dist$(\cdot, \cdot)$, with another function.

Lemma 4. Given a point p on the x-axis, $(x, 0)$, and a circle $C(0, r, r)$ of radius r, with center on y-axis, define dist'(p, C) as below:

$$\text{dist'}(p, C) = \begin{cases} x^2/r & x \leq r \\ x & x > r \end{cases}$$

We have

$$\frac{1}{3}\text{dist'}(p, C) \leq \text{dist}(p, C) \leq \text{dist'}(p, C), \tag{7.3}$$

(dist(p, C) is the distance from the point to the circle, which is $\left|\sqrt{x^2 + r^2} - r\right|$.)

The proof of Lemma 4 is relatively straight forward, and we omit the proof and show Figure 7.4 to demonstrate that the lemma is correct intuitively.

We now prove Lemma 7.2.1.

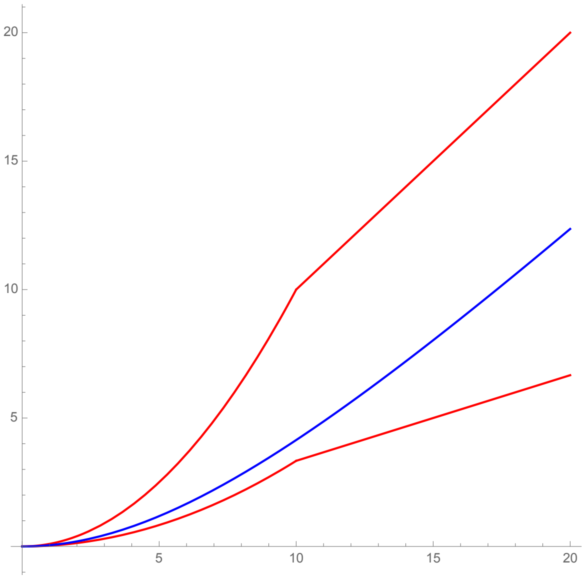


Figure 7.4. Illustration that the distance function dist$(\cdot,\cdot)$ is sandwiched between (1/3)dist'$(\cdot,\cdot)$ and dist'$(\cdot,\cdot)$ in Lemma 4.

Proof. Consider the i^{th} group of points P_i. Let C_i be a circle with center $(0, 2^i)$, tangent with x-axis (in other words, it has radius 2^i). We have

$$\begin{aligned}
\text{dist}(P_i, C_i) &= |P_i| \cdot \text{dist}(p_i, C_i) && (P_i \text{ contains } |P_i| \text{ copies of } p_i)\\
&\geq |P_i| \cdot \frac{1}{3} \cdot \text{dist'}(p_i, C_i) && (\text{Lemma 4, Inequality 7.3})\\
&= \frac{1}{3} \cdot 3^{N-i} \cdot 2^i && (\text{Definition of dist'}(\cdot,\cdot) \text{ in Lemma 4})
\end{aligned}$$

The contribution to the fitting cost of C_i from all the points to the left of and including p_i, that is, points in $P_0, \cdots, P_i$, is at most $4 \cdot (3^{N-i} 2^i)$:

$$\begin{aligned}
\sum_{j=0}^{i} \text{dist}(P_j, C_i) &= \sum_{j=0}^{i} |P_j| \text{dist}(p_j, C_i)\\
&\leq \sum_{j=0}^{i} |P_j| \text{dist'}(p_j, C_i) \quad = \sum_{j=0}^{i} 3^{N-j} \frac{2^{2j}}{2^i} = \frac{3^N}{2^i} \sum_{j=0}^{i} \left(\frac{4}{3}\right)^j \leq 4 \cdot 3^{N-i} \cdot 2^i
\end{aligned}$$

The contribution to the fitting cost of C_i from all the points to the right of P_i is at most $2 \cdot 3^{N-i} \cdot 2^i$:

$$\begin{aligned}\sum_{j=i+1}^{N} \text{dist}(P_j, C_i) &= \sum_{j=i+1}^{N} |P_j|\text{dist}(p_j, C_i) \\ &\leq \sum_{j=i+1}^{N} |P_j|\text{dist'}(p_j, C_i) \qquad = \sum_{j=i+1}^{N} 3^{N-j} \cdot 2^j \leq 2 \cdot 3^{N-i} \cdot 2^i\end{aligned}$$

Therefore, we can lower bound the contribution of P_i to the overall cost as following:

$$\frac{\text{dist}(P_i, C_i)}{\text{dist}(P, C_i)} \geq \frac{(1/3) \cdot 3^{N-i} \cdot 2^i}{4 \cdot 3^{N-i} \cdot 2^i + 2 \cdot 3^{N-i} \cdot 2^i} = \frac{1}{18}$$

□

7.2.2 Each p_i cannot be too heavy in the 1/100-coreset

We recall lemma 7.2.2:

Lemma 7.2.2 (Each p_i cannot be too heavy in an 1/100-coreset)**.** Let S be an 1/100-coreset of P. For each p_i, in the coreset S,

$$w(p_i) \leq 19|P_i|. \tag{7.2}$$

This lemma follows from the previous lemma, Lemma 7.2.1.

Proof. Let p_i in a point in P, and $w(p_i)$ is the weight of p_i in the 1/100-coreset S.

$$\begin{aligned}w(p_i)\text{dist}(p_i, C_i) &\leq \text{dist}(S, C_i) && \text{(non-negative weights)} \\ &\leq \big(1 + \frac{1}{100}\big)\text{dist}(P, C_i) && \text{(S is an 1/100-coreset)} \\ &\leq \frac{101}{100} \cdot 18 \cdot \text{dist}(P_i, C_i) && \text{(Lemma 7.2.1, Inequality 7.1)} \\ &= \frac{101}{100} \cdot 18 \cdot |P_i|\text{dist}(p_i, C_i) \\ &< 19|P_i| \cdot \text{dist}(p_i, C_i)\end{aligned}$$

Therefore,

$$w(p_i) \leq 19|P_i|.$$

□

7.2.3 Any consecutive chunk cannot be totally discarded

We recall lemma 7.2.3:

Lemma 7.2.3 (Any consecutive chunk cannot be totally omitted in an 1/100-coreset)**.** Let l be a sufficiently large number (we will determine the value of l in the proof). Partition the sequence $p_0, \cdots, p_N$, which are points in P, into chunks of length $2l+1$:

$$\underbrace{p_0, \cdots, p_{2l}}_{\text{first chunk } CH_1}, \underbrace{p_{2l+1}, \cdots, p_{4l+1}}_{\text{second chunk } CH_2}, \cdots, \underbrace{\cdots, p_N}_{\approx (N/2l)^{\text{th}} chunk}$$

For S to be an 1/100-coreset of P,

$$S \cap CH_i \neq \emptyset, \text{for every chunk.}$$

Proof. We give some intuition before the formal proof. The idea is to show that if a chunk, consisting of a set of consecutive points in P, is omitted completely in the coreset S, then there exists a circle which witnesses that $\text{dist}(S, C)$ does not approximate $\text{dist}(P, C)$. In particular, the points in the chunk contributes a large portion to the overall fitting cost. The points to the left of the chunk would be "too close" to the circle, and right "too lightweight", so $\text{dist}(P_j, C)$ would be small because either $\text{dist}(p_j, C)$ or $|P_j|$ is small. However, in the coreset S, the weights of p_j's cannot be inflated too much, by Lemma 7.2.2. Hence they would not be able to make up the large missing cost due to the omission of the chunk. Figure 7.5 and Figure 7.6

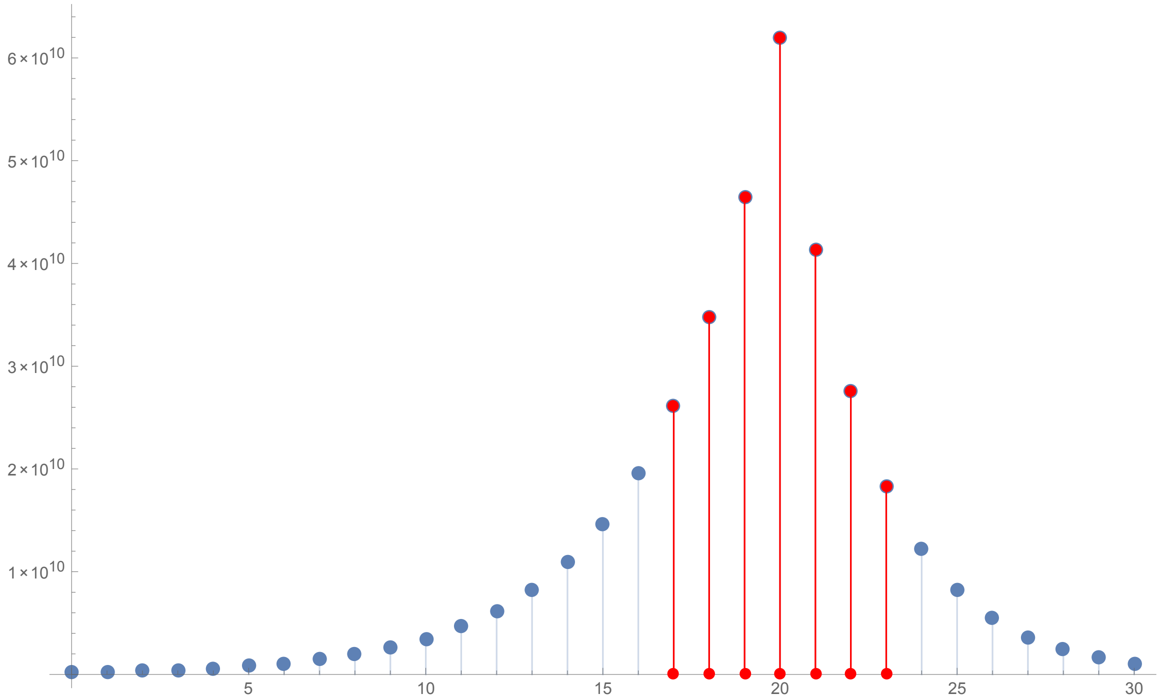


Figure 7.5. The points in P are show on the x-axis. (In reality the points are exponentially distributed on the x-axis.) The vertical line on the point $(p_j, 0)$ denotes roughly the value of $\mathrm{dist}(p_j, C_{20})$. Chunk $\{p_{17}, \cdots, p_{23}\}$ cannot be omitted because of circle C_{20}.

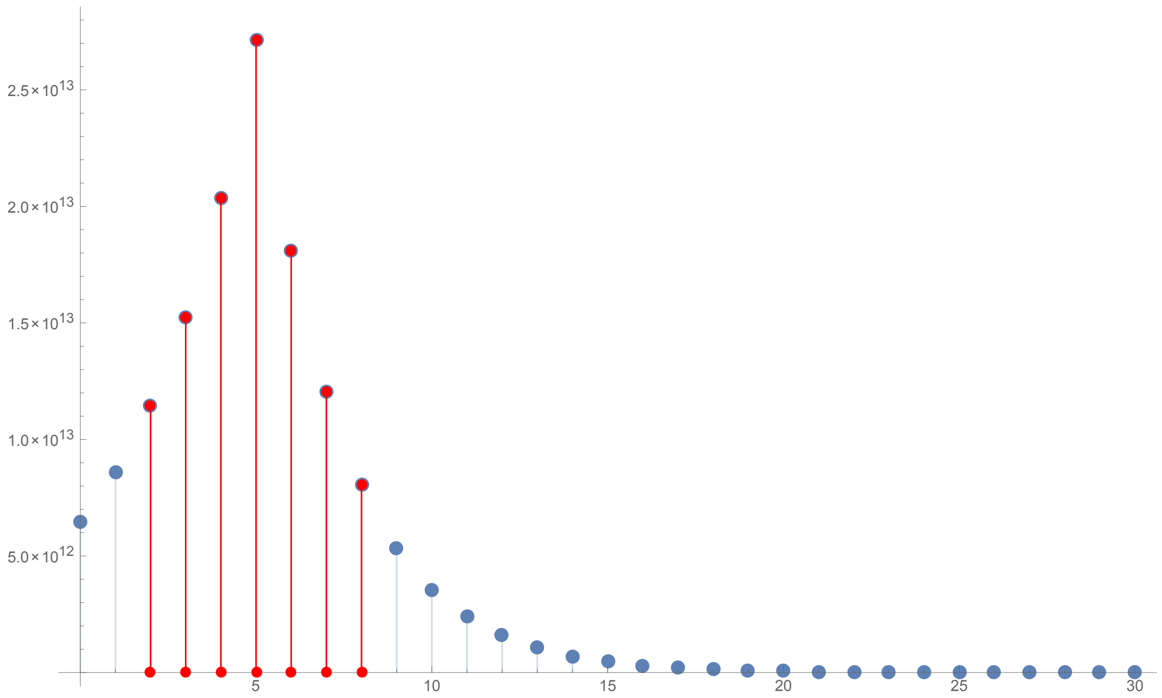


Figure 7.6. The points in P are show on the x-axis. (In reality the points are exponentially distributed on the x-axis.) The vertical line on the point $(p_j, 0)$ denotes roughly the value of $\mathrm{dist}(p_j, C_5)$. Chunk $\{p_2, \cdots, p_8\}$ cannot be omitted because of circle C_5.

illustrate this situation.

We now prove the above statements formally. Let l be a large constant (the value of which would be determined during the proof). Consider the chunk $\{p_{i-l}, \cdots, p_i, \cdots, p_{i+l}\}$. Consider the circle C_i, which has radius 2^i, and center at $(2^i, 0)$.

The contribution from the points to the left of the chunk, that is, $P_0, \cdots, P_{i-l-1}$, in the 1/100-coreset S, can be upper bounded as following:

$$\begin{aligned}
\sum_{j=0}^{i-l-1} w(p_j)\mathrm{dist}(p_j, C_j) &\le 19 \sum_{j=0}^{i-l-1} \mathrm{dist}(P_j, C_i) \qquad \text{(Lemma 7.2.1, Inequality 7.2)} \\
&\le 19 \sum_{j=0}^{i-l-1} \mathrm{dist'}(P_j, C_i) \qquad \text{(Lemma 4, Inequality 7.3)} \\
&\le 19 \sum_{j=0}^{i-l-1} 3^{N-j} \frac{(2^j)^2}{2^i} \\
&= 19 \frac{3^N}{2^i} \sum_{j=0}^{i-l-1} (\frac{4}{3})^j \\
&\le 19 \cdot 3 \cdot (\frac{3}{4})^l \cdot 3^{N-i} 2^i
\end{aligned}$$

The contribution from the right chunks, that is, $P_{i+l+1}, \cdots, P_N$, can be upper bounded as following:

$$\begin{aligned}
\sum_{j=i+l+1}^{N} w(p_j)\mathrm{dist}(p_j, C_i) &\le 19 \sum_{j=i+l+1}^{N} \mathrm{dist}(P_j, C_i) \qquad \text{(Lemma 7.2.1, Inequality 7.2)} \\
&\le 19 \sum_{j=i+l+1}^{N} \mathrm{dist'}(P_j, C_i) \qquad \text{(Lemma 4, Inequality 7.3)} \\
&= 19 \sum_{j=i+l+1}^{N} 3^{N-j} 2^j \\
&= 19 \cdot 3^N \sum_{j=i+l+1}^{N} (\frac{2}{3})^j \\
&\le 19 \cdot \frac{2}{3} \cdot (\frac{2}{3})^l 3^{N-i} 2^i
\end{aligned}$$

Now we can choose l, such that

$$19 \cdot 3 \cdot (\frac{3}{4})^l < \frac{1}{100},$$
$$19 \cdot \frac{2}{3} \cdot (\frac{2}{3})^l < \frac{1}{100},$$

If S does not include any points from the chunk $\{p_{i-l}, \cdots, p_i, \cdots, p_{i+l}\}$, then the fitting cost of S to C_i is at most $(1/50)3^{N-i}2^i$, by the two upper bounds on $\sum_{j=0}^{i-l-1} w(p_j)\text{dist}(p_j, C_j)$ and $\sum_{j=i+l+1}^{N} w(p_j)\text{dist}(p_j, C_i)$. However, for S to be an 1/100-coreset, it needs to be at least $(33/100)3^{N-i}2^i$:

$$\begin{aligned}
\text{dist}(S, C_i) &\geq (1 - 1/100)\text{dist}(P, C_i) && \text{(by definition of } \epsilon\text{-coreset)} \\
&\geq (1 - 1/100)\text{dist}(P_i, C_i) && \text{(points in } S \text{ have non-negative weights)} \\
&\geq \frac{99}{100} \cdot \frac{1}{3} \cdot \text{dist'}(P_i, C_i) && \text{(Lemma 4, Inequality 7.3)} \\
&= \frac{99}{100} \cdot \frac{1}{3} \cdot 3^{N-i}2^i \\
&= \frac{33}{100} \cdot 3^{N-i}2^i
\end{aligned}$$

□

7.3 Shape fitting problems with $\Omega(\log n)$ total sensitivity might admit constant-sized coreset

We have mentioned in the beginning of previous section that lower bound on total sensitivity does not imply a lower bound on the size of the core-set. In particular, the shape fitting problem with horizontal rays extending to the right, has total sensitivity $\Omega(\log n)$, yet it still admits a constant size coreset (Observation 1). Similarly, the shape fitting problem for line segments also have total sensitivity $\Omega(\log n)$, and it also admits constant-sized coreset.

It worth noting that in the construction of Observation 1, the i^{th} point has sensitivity roughly $1/i$. Here we show a construction of point set, which resembles the characteristic of the point set construction for circle fitting problem. Let P be a point set consisting of $\log n$ groups, where the i^{th} group contains 2^i copies of the point $(2^i, 0)$, $i = 0, \cdots, \log n$. The summation of sensitivities of the copies of the points in P_i is at least 1/4: because for a ray C starting at $(2^{i+1}, 0)$,

$$\text{dist}(P_i, C) = 2^i \cdot 2^i = 4^i,$$

and

$$\sum_{j=0}^{i} \text{dist}(P_j, C) = \sum_{j=0}^{i} (2^{i+1} - 2^j) \cdot 2^j < 4 \cdot 4^i,$$

$$\sum_{j=i+1}^{\log n} \text{dist}(P_j, C) = 0.$$

Therefore,

$$\sum_{p \in P_i} \sigma_P(p) \geq \frac{1}{4}.$$

However, the ray fitting problem has constant-sized coreset.

CHAPTER 8

CONCLUSION AND OPEN PROBLEMS

In this chapter, we summarize the main results in the thesis and point out some open problems.

The theme in this thesis is geometric approximation via core-set, and in particular, the concept of total sensitivity plays an important role in our algorithms for getting small coreset. Core-set is a succinct representation of the input point set, which has the property that for any shape, the fitting cost of the coreset approximates the cost of the input point set. Therefore, small coreset immediately provides a fast algorithm for finding a near-optimal solution for the input point set. In [19], it is shown that there is a close connection between total sensitivity and the size of coreset, that is, if a shape fitting problem has total sensitivity $\mathfrak{S}$, then it admits a coreset of size $O(\mathfrak{S}^2\tilde{d})$, where $\tilde{d}$ is roughly the VC-dimension of the family of shapes.

We summarize the contributions of this thesis. we show that shape fitting problems with small L_∞ core-set have small total sensitivity. This allows us to obtain small L_1 coreset for the a family of (j, k) projective clustering problems, for specific setting of j and k. We also obtain small coreset for integer (j, k) projective clustering problem, for general j and k. The sizes of coreset depends polylogarithmically in terms of n, which is the cardinality of the input point size, and exponentially in terms of k, j, and d, which is the dimension of the the input point set. Later, we show that the exponential dependence on d can be removed for (j, k) projective clustering problem from the upper bounds of total sensitivity. Hence, the sizes of the coreset only linearly depends on d, which is due to the fact that the quantity, $\tilde{d}$, still depends on d. For circle fitting problem on the plane, using the connection between L_∞ coreset and total sensitivity, we obtain coreset of size $O((\log n)^2)$, and we also show a lower bound of $\Omega(\log n)$ on the size of coreset for circle fitting problem.

In the following, we point out some open problems in the area of geometric approximation via coreset.

Problem 6 (near-linear algorithm for (j, k) projective clustering in high dimension)**.** We have obtained near-linear algorithm for the integer (j, k) projective clustering problem in high dimension via L_1-coreset. One interesting problem is whether it is possible to obtain the result without the extra assumption that points have integer coordinates that are polynomially bounded.

Problem 7 (small coreset for subspace approximation)**.** Depending on the distance function, the size of the coreset could vary. In [22], it is shown that for squared Euclidean distance, the subspace approximation problem (which corresponds to setting k to be 1 in (j, k) projective clustering), admits coreset whose size depends on neither the cardinality, nor the dimension of the input point set. It is an open problem whether this result could be generalized to the case of L_1 fitting, with the distance function being the Euclidean distance.

Problem 8 (Lower bound on the size of coreset for circle fitting problem)**.** In Theorem 19, we have shown that the lower bound on the size of coreset for circle fitting problem is $\Omega(\log n)$. In Theorem 3, the upper bound on the size of coreset is roughly

$O((\log n)^2)$. It would be interesting if the lower bound and upper bound can match: that is, either prove that the lower bound is $\Omega((\log n)^2)$, or improve the size of the coreset to $O(\log n)$.

REFERENCES

[1] http://en.wikipedia.org/wiki/curse_of_dimensionality. http://en.wikipedia.org/wiki/Curse_of_dimensionality.

[2] k-means clustering. http://en.wikipedia.org/wiki/K-means_clustering.

[3] k-medians clustering. http://en.wikipedia.org/wiki/K-medians_clustering.

[4] Latent semantic indexing. http://en.wikipedia.org/wiki/Latent_semantic_indexing.

[5] Linear regression. http://en.wikipedia.org/wiki/Linear_regression.

[6] Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Approximating extent measures of points. *J. ACM*, 51(4):606–635, July 2004.

[7] Pankaj K. Agarwal, Sariel Har-peled, and Kasturi R. Varadarajan. Geometric approximation via coresets. In *Combinatorial and Computational Geometry, MSRI*, pages 1–30. University Press, 2005.

[8] Pankaj K. Agarwal, Cecilia Magdalena Procopiuc, and Kasturi R. Varadarajan. Approximation algorithms for k-line center. In *Proceedings of the 10th Annual European Symposium on Algorithms*, ESA '02, pages 54–63, London, UK, UK, 2002. Springer-Verlag.

[9] Mihai Bādoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, STOC '02, pages 250–257, New York, NY, USA, 2002. ACM.

[10] Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-ramanujan sparsifiers. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 255–262, New York, NY, USA, 2009. ACM.

[11] T. A. Boroson and T. R. Lauer. Exploring the spectral space of low redshift qsos. *The Astronomical Journal*, 140, 2010.

[12] McGurk Rosalie C. and Ivezic Z.and Kimball A. E. Principal component analysis of sdss stellar spect. *The Astronomical Journal*, 139, 2010.

[13] Ke Chen. On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. *SIAM J. Comput.*, 39(3):923–947, August 2009.

[14] Anirban Dasgupta, Petros Drineas, Boulos Harb, Ravi Kumar, and Michael W. Mahoney. Sampling algorithms and coresets for ℓ_p regression. *SIAM J. Comput.*, 38(5):2060–2078, 2009.

[15] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[16] Amit Deshpande and Kasturi Varadarajan. Sampling-based dimension reduction for subspace approximation. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, STOC '07, pages 641–650, New York, NY, USA, 2007. ACM.

[17] Michael Edwards and Kasturi Varadarajan. No coreset, no cry: II. In *Proceedings of the 25th international conference on Foundations of Software Technology and Theoretical Computer Science*, FSTTCS '05, pages 107–115, Berlin, Heidelberg, 2005. Springer-Verlag.

[18] Dan Feldman, Amos Fiat, and Micha Sharir. Coresets forweighted facilities and their applications. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '06, pages 315–324, Washington, DC, USA, 2006. IEEE Computer Society.

[19] Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *STOC*, pages 569–578, 2011.

[20] Dan Feldman, Morteza Monemizadeh, and Christian Sohler. A ptas for k-means clustering based on weak coresets. In *Proceedings of the twenty-third annual symposium on Computational geometry*, SCG '07, pages 11–18, New York, NY, USA, 2007. ACM.

[21] Dan Feldman, Morteza Monemizadeh, Christian Sohler, and David P. Woodruff. Coresets and sketches for high dimensional subspace approximation problems. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 630–649, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics.

[22] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 1434–1453. SIAM, 2013.

[23] Dan Feldman and Leonard J. Schulman. Data reduction for weighted and outlier-resistant clustering. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 1343–1354. SIAM, 2012.

[24] G W. Furnas, T K. Landauer, L M. Gomez, and S T. Dumais. Human factors in computer systems. chapter Statistical semantics: analysis of the potential performance of keyword information systems, pages 187–242. Ablex Publishing Corp., Norwood, NJ, USA, 1984.

[25] Jesús García-López and Pedro A. Ramos. Fitting a set of points by a circle. In *Proceedings of the Thirteenth Annual Symposium on Computational Geometry*, SCG '97, pages 139–146, New York, NY, USA, 1997. ACM.

[26] Sariel Har-Peled. No, coreset, no cry. In *FSTTCS*, pages 324–335, 2004.

[27] Sariel Har-Peled. Coresets for discrete integration and clustering. In *Proceedings of the 26th international conference on Foundations of Software Technology and Theoretical Computer Science*, FSTTCS'06, pages 33–44, Berlin, Heidelberg, 2006. Springer-Verlag.

[28] Sariel Har-Peled. How to get close to the median shape. In *Proceedings of the twenty-second annual symposium on Computational geometry*, SCG '06, pages 402–410, New York, NY, USA, 2006. ACM.

[29] Sariel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. In *Proceedings of the twenty-first annual symposium on Computational geometry*, SCG '05, pages 126–134, New York, NY, USA, 2005. ACM.

[30] Michael Langberg and Leonard J. Schulman. Universal epsilon-approximators for integrals. In *SODA*, pages 598–607, 2010.

[31] Jiri Matousek. *Lectures on Discrete Geometry*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[32] J. Lewis P. Drineas and P. Paschou. Inferring geographic coordinates of origin for europeans using small panels of ancestry informative markers. *PLoS ONE*, 5(8), 2010.

[33] P. Paschou, P. Drineas, J. Lewis, C.M. Nievergelt, D.A. Nickerson, J.D. Smith, P.M. Ridker, D.I. Chasman, R.M. Krauss, and E. Ziv. Tracing sub-structure in the european american population with pca-informative markers. *PLoS Genetics*, 4(7), 2008.

[34] P. Paschou, M. W. Mahoney, A. Javed, J. R. Kidd, A. J. Pakstis, S. Gu, K. K. Kidd, and P. Drineas. Intra and interpopulation genotype reconstruction from tagging snps. *Genome Research*, 17(1), 2007.

[35] Budavri Tams, Wild Vivienne, Szalay Alexander S., Dobos Lszl, and Yip Ching-Wa. Reliable eigenspectra for new generation surveys. *Monthly Notices of the Royal Astronomical Society*, 394, 2009.

[36] Carlo Tomasi. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.

[37] Roberto Tron and Ren Vidal. A benchmark for the comparison of 3d motion segmentation algorithms. In *In CVPR*, 2007.

[38] Kasturi Varadarajan and Xin Xiao. A near-linear algorithm for projective clustering integer points. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 1329–1342. SIAM, 2012.

[39] Kasturi R. Varadarajan and Xin Xiao. On the sensitivity of shape fitting problems. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, pages 486–497, 2012.

[40] R. Vidal. Subspace clustering: Applications in motion segmentation and face clustering. *Signal Processing Magazine, IEEE*, pages 52–68, March 2011.

[41] Michael E. Wall, Andreas Rechtsteiner, and Luis M. Rocha. Singular value decomposition and principal component analysis. In M. Granzow D.P. Berrar, W. Dubitzky, editor, *A Practical Approach to Microarray Data Analysis*, pages 91–109. Kluwer: Norwell, MA (2003), 2005.

[42] Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV (4)*, pages 94–106, 2006.

[43] Chee Yap. Exact computational geometry and tolerancing metrology, 1994.