TEL-AVIV UNIVERSITY
RAYMOND AND BEVERLY SACKLER
FACULTY OF EXACT SCIENCES
BLAVATNIK SCHOOL OF COMPUTER SCIENCE

Coresets and Their Applications

Thesis submitted in partial fulfillment of the requirements for the
Ph.D. degree in the Blavatnik School of Computer Science,
Tel-Aviv University

by

Dan Feldman

The research work for this thesis has been carried out at
Tel-Aviv University, under the supervision of
Prof. Amos Fiat and Prof. Micha Sharir

Submitted to the Senate of Tel-Aviv University

December 2010

Acknowledgements

First, I am heartily thankful to my advisors, Prof. Amos Fiat and Prof Micha Sharir, for the huge amount of time they spent with me during my research. It is a real honor to work with such two great researchers from different areas of computer sciences. Their continuous support and constance guidance during the recent years encouraged my passion for doing good research.

For my parents, Edna and Itzhak Feldman, for giving me life in the first place, and for their unconditional support and encouragement to pursue my interests over the years. They are responsible for discovering my excitement in computers when I was only eight years old, by sending me for a computer science course nearby.

I owe my deepest gratitude to my wife Adi and my son Ariel for providing me the time and space that was needed during my research, sometimes on account of the usual tasks as a family person.

My research would not have been possible without all the freedom that I got from my advisors and family for doing theoretical work while getting very practical support.

I am also very grateful to my co-authors Dr. Michael Langberg and Dr. Christiran Sohler for inviting me to research with them, and to Prof. Kobbi Nissim, Prof. Haim Kaplan, Morteza Monemizadeh, and Dr. David Woodruf for the great discussions and joint work we had together, and hopefully will have in the future.

Special thanks goes to Hanna and Shmulik David, and to Sagi Hed, for helping me with all the arrangements that were needed in order to submit this thesis while I was abroad.

Abstract

In this thesis we investigate the construction and applications of *coresets* (small sets which approximately represent much larger input sets, in term of various objective measures) to several problems in geometric optimization.

Bi-criteria approximation algorithms

We consider the problem of approximating a set P of n points in $\mathbb{R}^d$ by a collection of k j-dimensional flats, and extensions thereof, under the median / mean / center measures, in which we wish to minimize, respectively, the sum of the Euclidean distances from each point of P to its nearest flat, the sum of the squares of these distances, and the maximum such distance. Problems of this kind belong to the area of *projective clustering*.

Such problems cannot be approximated in polynomial time, for every approximation factor, unless *P=NP* but do allow *bi-criteria approximations*, where one allows some leeway in both the number of flats and the quality of the objective function. We give a very simple randomized bi-criteria approximation algorithm, which produces, with high probability, at most $\alpha(k, j, n) = \log n \cdot (jk \log\log n)^{O(j)}$ flats, which exceeds the optimal objective value for any k j-dimensional flats by a factor of no more than $\beta(j) = 2^{O(j)}$.

We use this bi-criteria approximation in the construction of coresets for projective clustering; see Chapter 4. Our bi-criteria algorithm has many advantages over previous work, in that it is much more widely applicable (wider set of objective functions and classes of clusters) and much more efficient — reducing the running time bound from $O(n^{\mathrm{poly}(k,j)})$ to $O(dn) \cdot (jk)^{O(j)}$.

We give full details of this work in Chapter 3. A preliminary version has appeared in [FFSS07]; Since the publication of [FFSS07] in 2007 it has been cited and used in subsequent work [FL08, FFKN09, FMSW10].

Coresets for projective clustering

We develop efficient $(1+\varepsilon)$-approximation algorithms for projective clustering problems, where $k=1$ or $j=1$ (one j-dimensional flat or many lines in $\mathbb{R}^d$).

To achieve coresets for projective clustering we introduce *coresets for weighted (point) facilities*. These prove to be useful for such generalized facility location problems, and may be of additional independent interest.

Applications include approximations for generalized k-median and k-mean line problems, i.e., finding k lines that minimize the sum (or sum of squares) of the distances from each input point to its nearest line. Other applications are generalizations of linear regression problems to multiple regression lines, new SVD/PCA generalizations, and many more. The results significantly improve on previous work, which deals efficiently only with special cases.

We give full details of this work in Chapter 4. A preliminary version has appeared in [FFS06]; Since the publication of [FFS06] in 2006 it has been cited and used in many subsequent works [FMSW10, Des07, DV07, FFSS07, FMS07, SV07, DDH$^+$08, LS08].

Contents

Chapter 1

Background

In this chapter we give the background required to place our contributions in context.

We introduce coresets and give examples of their use. We introduce a variety of problems related to projective clustering.

1.1 Coresets

Approximation algorithms in computational geometry often make use of random sampling [CKMN01, Mul93], feature extraction [DM98, CC01], and ε-samples [Hau92].

Coresets can be viewed as a general concept that includes all of the above, and more. See a comprehensive survey on this topic by Agarwal, Har-Peled, and Varadarajan [AHPV04].

It is not clear that there is any commonly agreed-upon definition of a coreset, despite several inconsistent attempts to do so [HPM04, AHPV05, Cla05, HPK07, DRVW06, FMS07].

In our context, the input is a set P of points in $\mathbb{R}^d$, and we consider a set F of k points or affine subspaces in $\mathbb{R}^d$. P is typically much larger than F. Let cost be some function of P and F. We interpret $\mathrm{cost}(P, F)$ as the result of a query F on dataset P.

Typically, we will be interested in queries that measure the distance between P and F, where this distance could be the sum of the Euclidean distances from each point of P to the nearest object in F, or the sum of squares of such distances, or the maximum such distance. In addition to point sets P, we also consider

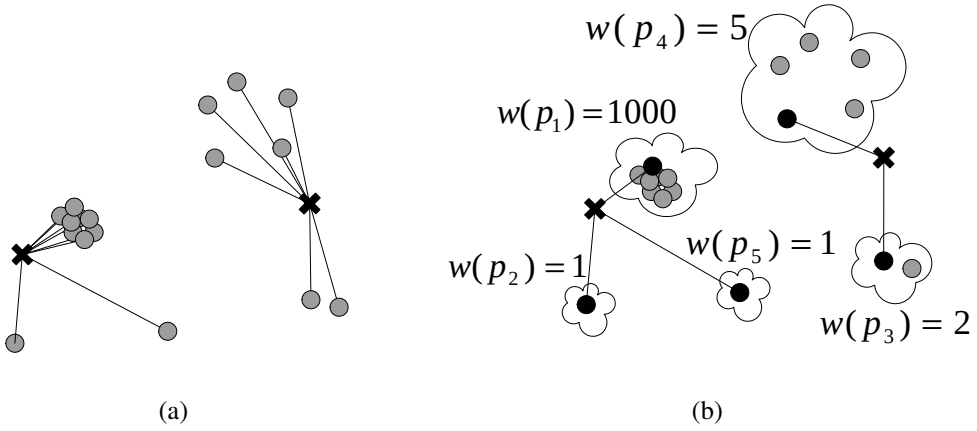


Fig. 1.1: ***(a)*** A set P of n points in the plane and two facilities. ***(b)*** A weighted coreset (black points) of P with the same two facilities.

weighted point sets S, where every $p \in S$ has an associated multiplicative weight $w(p) \in \mathbb{R}$. We extend the definition of $\text{cost}(P, F)$ in a natural manner.

A concrete example is the set of k-median queries, where each query is of the form $F = \{f_1, f_2, \ldots, f_k\}$, where each f_i is a point, or an affine subspace of $\mathbb{R}^d$, and returns the sum of distances

$$\text{cost}(P, F) = \sum_{p \in P} \min_{1 \le i \le k} \text{dist}(p, f_i).$$

For a set S of weighted points, we define

$$\text{cost}(S, F) = \sum_{p \in S} w(p) \cdot \min_{1 \le i \le k} \text{dist}(p, f_i).$$

Let $\mathcal{F}$ be the set of all possible queries. Fix $\mathcal{F}$, i.e., fix both the types of objects in F (called *facilities*) and the function $\text{cost}(P, F)$. A coreset scheme $\mathcal{A}$ for a class of queries $\mathcal{F}$ is an algorithm that gets as input a finite set P of points and a parameter $\varepsilon > 0$, and outputs a set $\mathcal{A}(P) = S$ of weighted points in $\mathbb{R}^d$ such that for every $F \in \mathcal{F}$:

$$(1 - \varepsilon) \cdot \text{cost}(P, F) \le \text{cost}(\mathcal{A}(P), F) \le (1 + \varepsilon) \cdot \text{cost}(P, F).$$

The set S is called a *coreset*. Typically, one expects S to be much smaller than P. In the concrete example given above, it would be a *coreset for k-median*.

As a motivating example, let P be a set of n points in $\mathbb{R}^2$. Har-Peled and Kushal [HPK07] describe how to construct a coreset for k-median of size independent of n. See Fig 1.1.

The sum of distances is used in "median" problems; in "mean" problems we replace it by the sum of squared distances. Har-Peled and Kushal [HPK07] also provide a construction of a small coreset for k-mean. That is, rather than sum of the Euclidean distances to the nearest facilities, one takes the sum of the squared Euclidean distances.

Such coresets imply an efficient approximation algorithm for the k-median (or k-mean) problem: an optimal set of k facilities for S is a good approximation to the optimal set for P, and, since S is small, the former set may possibly be found efficiently, e.g., via brute force.

Coresets have been the subject of many recent papers [HP04a, APV02, BHPI02, HPV02, HP04b, HPM04, Cha04, AHPV05, HPK07, Che06, FFS06, FMS07] and several surveys [AHPV05, CS07]. Coresets have been used to great effect for a host of geometric and graph problems, including k-median [HPM04, HPK07, Che06, FMS07], k-mean [HPM04, HPK07, FMS07], k-center [HPV04], subspace approximation [HPV02, HP04b, FFS06], shape fitting [AHPV04], k-line median [FFS06], k-line center [HPV02, HPV04], moving points [HP04a], max TSP [FS05], minimum spanning tree [CEF$^+$05, FIS08], maximal margin learning, etc. Coresets also imply streaming algorithms for many of these problems [HPM04, AHPV05, FS05, BFLS07, FMS07, FIS08, LS08].

1.2 Projective Clustering

Clustering is the process of partitioning objects into clusters such that objects in the same cluster are similar, and objects in different clusters are dissimilar. Clustering is a central problem in computer science. It has many applications in different areas including bioinformatics, pattern recognition, data compression, and information retrieval. It is relevant to issues of unsupervised learning, classification, databases, spatial range-searching, data-mining, etc.

Input points to be clustered may be in a very high-dimensional space, (e.g., documents represented as a bag of English words in 600,000-dimensional space or gene expression data for 10,000 genes). Because of the wide variety of applications, there is no general formulation of clustering (except for the vague one given

above).

Let $P \subset \mathbb{R}^d$ be a set of n points in d-dimensional space. A reasonable goal is to "approximate" P by a small collection, F, of "shapes" in $\mathbb{R}^d$. Depending on the problem, elements of F may be restricted to be single points, lines, or j-dimensional subspaces of $\mathbb{R}^d$ ($j < d$). (Of course, shapes can also be non linear, such as spheres or cylinders, but in this thesis we only consider the linear case.)

For a point $p \in P$, let $c = c(p) \in F$ be the element $c \in F$ closest to p in Euclidean distance, i.e. $\mathrm{dist}(p, c) = \min_{q \in c} \|p - q\|$ (ties broken arbitrarily). Every $c \in F$ represents a cluster, and point $p \in P$ is said to be associated with cluster $c(p)$. The set F is called a *projective clustering* of P.

Typically, the projective clustering problem pre-specifies the class of allowable cluster shapes in F and their number, k. The *value* of a projective clustering F is some function of the distances between points $p \in P$ and their associated clusters $c(p)$. A good projective clustering is one of small value.

Common objectives are to minimize $\sum_{p \in P} \mathrm{dist}(p, c(p))$, $\sum_{p \in P} \big(\mathrm{dist}(p, c(p))\big)^2$, or $\max_{p \in P} \mathrm{dist}(p, c(p))$, where $\mathrm{dist}(\cdot, \cdot)$ is the Euclidean distance. A *projective clustering* F that minimizes one of these three main objective functions, is referred to as a *k-median*, *k-mean*, or *k-center*, respectively.

Example: k-line median. In Fig 1.2 we see a set $F = \{\ell_1, \ell_2\}$ of two lines in $\mathbb{R}^2$, and $c(p) = \ell_1$ (as p is closer to ℓ_1 than to ℓ_2). The k-line median is a special case of projective clustering problem, where the set F is k lines in $\mathbb{R}^d$, that minimizes the sum of distances $\sum_{p \in P} \mathrm{dist}(p, c(p))$; see Fig. 1.3.

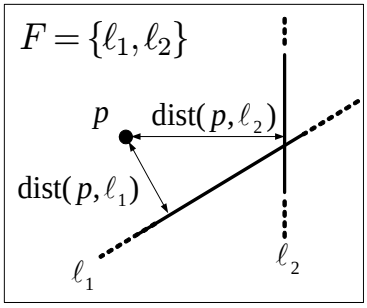


Fig. 1.2: $\mathrm{dist}(p, F) = \min\{\mathrm{dist}(p, \ell_1), \mathrm{dist}(p, \ell_2)\}$

The case where F is a set of k low-dimensional flats (affine subspaces) in $\mathbb{R}^d$ has been the subject of many studies (for example, [AGGR98, APW+99, AP00,

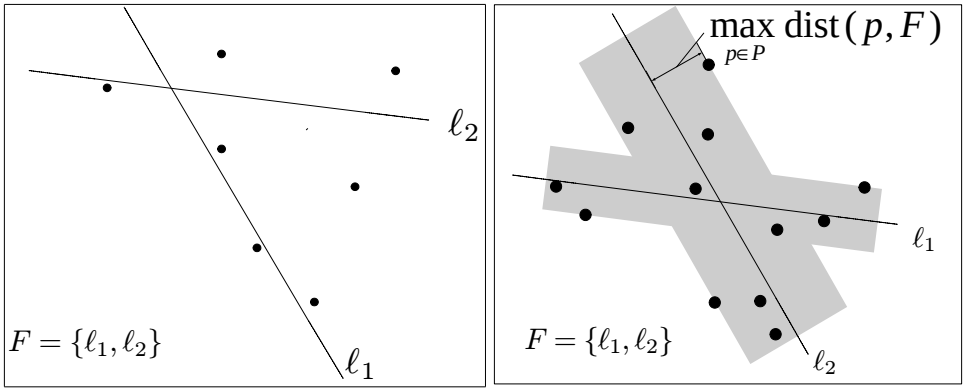


Fig. 1.3: **(left)** 2-line median: $\min_F \sum_{p \in P} \mathrm{dist}(p, F)$. **(right)** 2-line center: $\min_F \max_{p \in P} \mathrm{dist}(p, F)$.

AY00, HP04a, AJMP02, AHPV05, DV07, SV07]). Heuristics for projective clustering can be found in [AM04]. In this case, the problem is to find a set of k low-dimensional flats that approximately matches the input points. This problem appears in a great many areas. For example, "one of the most fundamental problems in computer vision is to find straight lines in an image" [Bre96]. Other examples include: matrix approximation [DRVW06], other problems in image processing [TT96], data compression [Ric86], graphics [KS90], socioeconomics [KA04], and many more.

A β-*approximation* algorithm for a k-projective clustering problem should produce a k-projective clustering, F, with value not greater than β times the optimum, i.e., the smallest value of any k-projective clustering.

Unfortunately, even for planar point sets P, it is *NP*-complete to determine whether there exist k lines (1-flats) whose union covers P [MT83], when k is part of the input. If the k lines indeed cover the points of P, then the sum of distances, sum of squared distances, and maximal distance, are all zero. Hence, any finite approximation to the k-line median, mean, or center problems is *NP*-hard, even for point sets P in the plane.

In Table 1.1 we summarize recent work on approximate projective clustering of k j-dimensional flats in $\mathbb{R}^d$. The constant in the notation $O(\cdot)$ is assumed to be an absolute constant, independent of j and k.

Note that all the algorithms in the table are (at least) exponential in k. Also, for the general case of $j > 1$ and $k > 1$ the existing algorithms are inefficient, and

take time $\Omega(n^{\text{poly}(k,j)})$. Heuristics that address the projective clustering problem for $j, k > 1$ include PROCLUS [APW+99], ORCLUS [AY00], DOC [AJMP02], and CLIQUE [AGGR98]. Other heuristics for projective clustering can be found in [AM04], with more references therein.

In the next section we define a different kind of approximation, known as an (α, β) bi-criteria approximation. In Sections 1.2.2 and 1.2.3, we discuss related work that gives constant factor approximations of this kind for projective clustering when either $j = 1$ or $k = 1$.

1.2.1 Bi-criteria Approximations

As noted in the previous section and Section 1.1, the best known algorithms for projective clustering where $k, j > 1$ take time $\Omega(n^{\text{poly}(k,j)})$. Moreover, as also mentioned, the problem is NP-hard for non-constant k, even for $j = 1$ and $d = 2$; see [MT83]. It is thus natural to try to find a *bi-criteria approximation*, where one allows some leeway in both the number of flats and the quality of the objective function. Bi-criteria approximation have appeared in many contexts [MI94, AP00, HP04a, HPM04, ABG06, Che06, DV07, Yan08].

Definition 1.1 ((α, β)-bi-criteria approximation for projective clustering)**.** *For a given point set $P \subset \mathbb{R}^d$, an (α, β)-bi-criteria approximation for k-projective clustering by j-dimensional flats is a set F of α j-dimensional flats whose value is within a factor of β from the minimal value of any k j-dimensional flats.*

The parameters α and β in Definition 1.1 may depend on k, j, d, and n, where the dependence on n should be small (say, polylogarithmic), or — even better — independent of n.

In Table 1.2 we summarize the current state of affairs regarding such bi-criteria approximations for projective clustering.

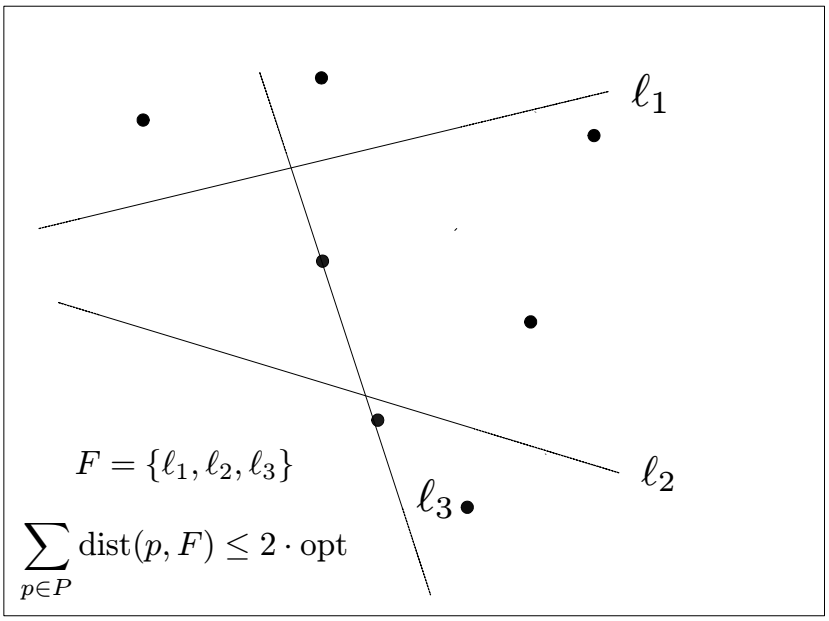
In Chapter 3, we present a new algorithm for constructing bi-criteria approximation for the general projective clustering problem, that takes time linear in n and only polynomial in k. These results appear in rows marked $\star$ in Table 1.2, and have appeared in print as [FFSS07].

1.2.2 k-Line Median/Mean Clustering ($j = 1$)

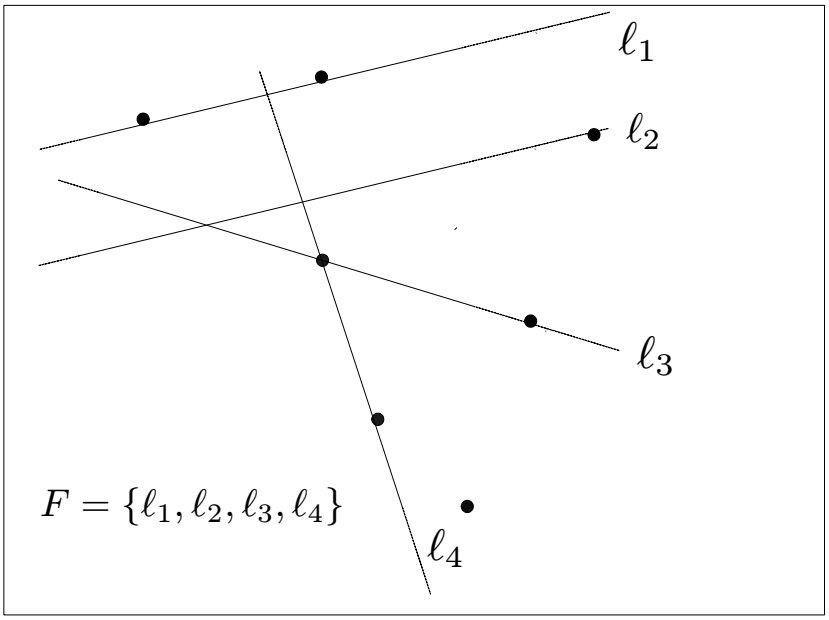
Recall that the problem of approximate k-line median (resp., mean) is the special case of approximate projective clustering, with $j = 1$, in which we seek a set of k

Flat dim. j	$k = \#$ Flats	Objective Function	Approx	Ref.	Time
1	$k \geq 1$	median mean	FPTAS	$\star\star$ [FFS06]	$nd \cdot k^{O(1)}$ $+(\varepsilon^{-d} \log n)^{O(dk^2)}$
$j \geq 1$	1	median	FPTAS	$\star\star$ [FFS06]	$nd \cdot (j)^{O(j^2)}$ $+(\varepsilon^{-1} \operatorname{polylog} n)^{O(d^2 j^2)}$
$j \geq 1$	1	mean	Exact	SVD [Pea01]	$\min\{O(nd^2), O(n^2 d)\}$
$j \geq 1$	1	mean	PTAS	[DV06, HP06b, Sar06]	$nd \operatorname{poly}(j, 1/\varepsilon)$
$j \geq 1$	$k \geq 1$	mean	PTAS	[DRVW06]	$d(n/\varepsilon)^{O(jk^3/\varepsilon)}$
$j \geq 1$	1	median	PTAS	[SV07]	$nd \cdot 2^{O(j/\varepsilon \log^2(1/\varepsilon))}$
$j = 1$ $d = 2$	1	median	Exact	[Dey98]	$O(n^{4/3} \log^2 n)$
$j \geq 1$	$k \geq 1$	median	PTAS	[SV07]	$d(n/\varepsilon)^{\operatorname{poly}(j,k,1/\varepsilon))}$
$j \geq 1$	1	center	PTAS	[HPV02]	$dn^{O\left(j/\varepsilon^5 \log(1/\varepsilon)\right)}$
$j \geq 1$	1	center	PTAS	[Pan04]	$dn \cdot \exp\left(\frac{2^{O(j^2)}}{\varepsilon^2} \log \frac{1}{\varepsilon}\right)$
$j \geq 1$	$k \geq 1$	center	PTAS	[HPV02]	$dn^{O\left(jk/\varepsilon^5 \log(1/\varepsilon)\right)}$
1	$k \geq 1$	center	FPTAS	[APV02]	$n \log n \cdot \varepsilon^{O(-d-k)} k^{O(k)}$ $+ \log n \cdot (k/\varepsilon)^{O(d^2 k^2)}$
$j = 1$ $d = 2$	2	center	Exact	[JK95]	$O(n^2 \log^2 n)$
$j = 1$ $d = 2$	2	center	3-approx	[AP00]	$O(n \log n)$

Table 1.1: Approximate projective clustering. The input is a set $P \subset \mathbb{R}^d$, $|P| = n$, the goal is to find a good approximation for P, within relative error $1 + \varepsilon$, using k j-dimensional flats. Unless *P=NP*, all such approximations must be superpolynomial in k. The first two rows above, marked $\star\star$, form part of the thesis; see Chapter 4.



(a)



(b)

Fig. 1.4: ***(a)*** A $(3, 2)$-approximation for the 2-line median of P. ***(b)*** A $(4, 1/2)$-approximation for the 2-line median of P.

$P \subset \mathbb{R}^d$	Flat dim. j	Objective Function	α	β	Ref.	Time
$d = 2$	$j = 1$	center	$O(k \log k)$	6	[AP00]	$O(nk^2 \log^4 n)$
$d = 2$	$j = 1$, $k \leq n^{1/6}$	center	$O(k \log k)$	1	[HP04a]	$O(n \log k)$
$d = 3$	$j = 2$	center	$O(k \log k)$	24	[AP00]	$n^{3/2}k^{11/4} \log^{O(1)} n$
$d \geq 1$	$j = 1$	center	$O(dk \log k)$	8	[AP00]	$O(dnk^3 \log^4 n)$
$d \geq 1$	$j \geq 1$	center mean median	$\log n$ $\cdot(jk \log \log n)^{O(j)}$	$2^{O(j)}$	$\star$ [FFSS07]	$O(dn) \cdot (jk)^{O(j)}$
$d \geq 1$	$j \geq 1$	center mean median	$(2^d jk \log n)^{O(j)}$	1/2	$\star$ [FFSS07]	$O(dn) \cdot (jk)^{O(j)}$ + $(2^d jk \log n)^{O(j)}$

Table 1.2: Results on bi-criteria approximate projective clustering. The input is a set $P \subset \mathbb{R}^d$, $|P| = n$, the goal is to find an approximation for P using α j-dimensional flats to within a β factor off the optimal such approximation by k j-dimensional flats. The last two entries are contributions of this thesis. Our bi-criteria approximation holds simultaneously for all three main objective functions.

lines in $\mathbb{R}^d$ such that the sum of the distances (resp., squared distances), from the points of P to their closest lines is minimized, up to a factor of $(1+\varepsilon)$.

Exact solutions are available in certain special cases. In particular, the 1-line mean can be computed in $O(n)$ time using the Singular Value Decomposition, for any fixed d; see [Pea01]. For $k = 1$ and $d = 2$, Yamamoto et al. [YKII88] give an $O(n^{1.5}\log^2 n)$-time algorithm that computes a 1-line median for a set of n input points. Using Dey's improved bound on the number of halving lines [Dey98], the algorithm can be improved to $O(n^{4/3}\log^2 n)$. In previous work [Fel04], we gave an exact (optimal) solution for the k-line-mean in the plane, which takes $O(n^3)$ time for $k = 2$, and $n^{O(k^2)}$ for $k \geq 3$.

No near linear time approximation algorithms are known for the k-line mean or k-median where $k > 1$, or for the 1-line median where $d > 2$. A simple example is, for an input set of points in $\mathbb{R}^3$, to find a 1-dimensional flat (line) in $\mathbb{R}^3$ that approximately minimizes the sum of distances from the points. Recently, Deshpadne et al. [DRVW06] gave an $(n/\varepsilon)^{O(k/\varepsilon)}$ PTAS (polynomial-time approximation scheme) for computing the k-line mean.

Many heuristics for the k-line median problem, such as the Hough transform and Independent Component Analysis (ICA), have also been proposed (see references in [HOK01]).

1.2.3 Approximation of Points by an Affine Subspace ($k = 1$)

Confronted with high-dimensional data arising from either word-document count, global climate patterns or any one of the myriad other sources, most scientific approaches attempt to extract a good low-dimensional summary. This desire to reduce dimensionality may be seen as a consequence of Occam's Razor, and the scientific methodologies we have in mind include data mining and statistics.

A flat (an affine subspace) f in $\mathbb{R}^d$ is defined to be a translation of a linear subspace. We are interested in the following approximate flat fitting problem: Given P as above, and an integer $1 \leq j \leq d-1$, find a j-dimensional flat f (a j-flat in short) such that the sum of distances (or the sum of squared distances) from the points of P to f is minimized, up to a factor of $(1+\varepsilon)$. We will refer to the special case where $\varepsilon = 0$ as the exact flat fitting problem.

The optimal j-subspace (which passes through the origin) that minimizes the sum of squared distances from P is obtained by the span of the j right singular vectors corresponding to the top j singular values of the singular value decomposition (SVD) of the $n \times d$ matrix whose rows correspond to the points of

P [Pea01]. This leads to a polynomial (in fact, $O(nd \min\{n, d\})$-time) algorithm for this problem; see the discussion in [Pea01].

Similarly, one can compute the j-flat f that minimizes the sum of squared distances from P (the j-flat mean problem) by using the fact that f contains the point $\bar{p} = \sum_{p \in P} p / \|p\|$ (also known as the center of mass). Hence, f can be computed by translating the origin to $\bar{p}$, and computing the optimal j-subspace for the new set $\{p - \bar{p} \mid p \in P\}$. This method is called *principal component analysis* (PCA) [Jol86].

For the ε-approximate problem for small j, recent work gives algorithms that are near linear in ndj/ε [SV07].

Although the j-flat mean can be computed in polynomial time, no analogous efficient algorithms are known for the j-flat median or its approximations, for $1 \leq j < d-1$. Prior to our work, no polynomial time approximation was known for a j-flat ($1 \leq j < d-1$) that minimizes the sum of distances to P (even for $j = 1$ and $d = 3$); these are cited as "interesting open problems" in [Sch99, DH02, BMS99].

The 1-point median problem is known as the Fermat-Weber problem, which reduces to minimizing a (complicated but) convex function over $\mathbb{R}^d$. A polynomial time approximation algorithm for the problem is given in [FMS07].

The case $j = d-1$ is referred to as the median hyperplane problem. Assuming the input point set P spans $\mathbb{R}^d$, it was observed that the optimal hyperplane is the span of a subset of d points of P. Based on this, algorithms that run in $O(n^d)$ time are known for this problem [BMS99]; see also the surveys [MS98, KM93].

In Chapter 4 of this thesis, we give algorithms that compute a $(1 + \varepsilon)$-approximation to the j-flat/subspace median in linear time, for any fixed d and for any $1 \leq j \leq d - 1$. A preliminary version of this result has appeared in [FFS06].

1.3 Variations of Projective Clustering

In this section we present several extensions and variations of the projective clustering problem that was introduced in Section 1.2.

Restricted Facility Location: Approximate the k-line median/mean or j-flat median/mean with additional constraints on the allowed location of the lines/flat, by forbidding them, or alternatively forcing them, to pass through certain locations.

Polynomial-time algorithms for a good approximate $(d - 1)$-flat with respect to the sum of distances or squared distances, and subject to additional restrictions,

are given in [DH02, Sch99]. Note that even in the case of one flat, or even one line in the plane ($j = 1$, $d = 2$), algebraic methods, such as the SVD/PCA, cannot handle constraints.

Approximate k-regression lines and M-estimators: Solve projective clustering with vertical (regression) distances (in the direction of the x_d-axis), squared or non-squared, instead of Euclidean distances.

A $(1 + \varepsilon)$-approximation for the j-flat mean, for squared regression distances with no constraints, can also be computed in $O(n)$ time using SVD. The 1-mean (regression) line in the plane ($d = 2$), can be computed in $O(n)$ time [YKII88]. For $d > 2$, a PTAS that takes $O(n \log n)d^{O(1)} + O(n)(1/\varepsilon)^{O(1)}$ time was recently suggested in [Cla05] for the hyperplane median problem ($j = d-1$) with vertical (regression) distances. Prior to the work described in this thesis, no results were known for the case $1 < j < d - 1$, or when there are constraints on the location of the flat.

Data Fitting with lines and points: For a fixed k and k', or for a fixed value of $k + k'$, find a set of k lines and k' points that minimizes the sum of distances, or of squared distances, from each input point to its nearest facility (with or without location constraints).

One possible interpretation of the data fitting problem is that we want to fit the data to k lines, and allow (up to) k' *clusters of outliers*; however, in this formulation the quality of the solution still depends on the distance from the outliers to the centers of their clusters. Since k' represents the number of outlier *clusters* and not the number of outliers, this may suggest a way to deal with outliers when their exact number is not known. Outliers were investigated for the k-(point) mean and median problems [COP03, HPW04].

Apart from the work described herein (Chapter 4) we do not know of other generalizations for linear facilities, even for a single line in the plane.

For all these variants of the problem, we give efficient approximations in Chapter 4.

Chapter 2

Our Contributions

In this chapter we describe the contributions of this thesis. In Section 2.1 we describe our results that relate to projective clustering of k affine j-dimensional subspaces in $\mathbb{R}^d$. In Section 2.2, we describe our coresets for the special cases $j = 1$ and $k = 1$ and their applications. We construct these coresets using coresets for weighted facilities, which are described in Section 2.3.

2.1 Bi-criteria Approximation Algorithms For Projective Clustering

In Chapter 3, we present an algorithm that produces an (α, β) bi-criteria approximation for k-projective clustering, for point sets in any dimension $d \geq 1$, by lines or flats of any dimension $j \leq d - 1$. Our algorithm is motivated by and related to prior work on bi-criteria approximations for other problems, such as [HP04a, HPM04, Ind99].

We achieve (α, β)-bi-criteria approximation with

$$\alpha(k, j, n) = \log n \cdot (jk \log\log n)^{O(j)} \text{ and } \beta(j) = 2^{O(j)},$$

in time $O(dn) \cdot (jk)^{O(j)}$. Furthermore, this bi-criteria approximation holds simultaneously for all three objective functions: median, mean, and center.

It is noteworthy that the running time has only linear dependence on both the dimension d, and the number of input points n.

As Table 1.2 states, prior work on such approximations has only dealt with very limited projective clustering problems, and only for k-center clustering problems.

Some implications of bi-criteria approximation

Table 1.1 includes projective clustering approximation results from this thesis. Rows marked $\star\star$ describe an FPTAS (fully polynomial-time approximation scheme) for the mean and median objective functions for any number k of line clusters or for a single j-flat cluster, $j \geq 2$. The FPTAS of [FFS06], which also described in Chapter 4, is obtained by first constructing a coreset for the corresponding problem. As in many other coreset constructions, the construction of this coreset requires a bi-criteria approximation for the problem to start with — the subject of Chapter 3.

We remark that many other results follow from our bi-criteria approximation. For example, using this approximation, one can derive an FPTAS for the k-line center clustering problem that takes $O(n)$ time, improving upon the $O(n \log n)$ bound of [APV02]. One can also derive explicit and efficient constructions for related *coresets* (see [AHPV05, FFS06]), previously unknown, such as coresets for a single j-flat or for k lines (center/mean/median). Some of these developments are given in this thesis.

2.2 Coresets for Projective Clustering

We develop efficient $(1+\varepsilon)$-approximation for linear facilities (lines or j-dimensional flats in $\mathbb{R}^d$). Although coresets for linear facilities are discussed in several places [AHPV05, DRVW06], no constructions have been suggested prior to the work described in this thesis.

Using these coresets we obtain an LTAS (linear-time approximation scheme, i.e., $O(n)$-time, $(1+\varepsilon)$-approximation algorithm) for the following problems and the extended problems in Section 1.3, all having as input a set P of n points in $\mathbb{R}^d$.

Coreset for linear and point facilities: We find a small weighted subset that well approximates the sum of distances, or of squared distances, from the points of P to *any* given set of $0 \leq i \leq k$ lines and at most $k - i$ points in $\mathbb{R}^d$, up to a factor of $(1+\varepsilon)$. We construct such coresets of size $\varepsilon^{-d-k}(\log n)^{O(1)}$ in $O(n)$ time, for any fixed $k, d \geq 1$. The same coreset also approximates the sum of (squared or unsquared) regression distances (i.e., distances measured in the x_d-direction) that is defined as follows. The regression distance from the point $x = (x_1, \cdots, x_d)$ to a hyperplane f is the minimum Euclidean distance between x to a point $y \in f$ such that $(y_1, \ldots, y_{d-1}) = (x_1, \ldots, x_{d-1})$. If there is no such point y, the regression

distance is infinite.

Coreset for a single flat: We find a small weighted subset that well approximates the sum of distances, or of squared distances, from P to *any* (single) j-dimensional flat, $1 \leq j \leq d-1$. We construct such coresets of size $\varepsilon^{-d-1}(\log n)^{O(j^2)}$ in $O(n)$ time, for any fixed $d \geq 1$.

Each of the problems in Section 1.3 is easy to solve once a coreset S is available: Since S has small size, we can use any (possibly inefficient) algorithm for computing, say, the exact or approximate k-line median for S (see, e.g., [DRVW06, YKII88]), and then report it as an approximate k-line median for the whole input set. The same approach handles each of the other variants.

2.3 Coresets For Weighted Facilities.

To tackle projective clustering problems that deal with linear facilities where either $k = 1$ or $j = 1$, we introduce a novel tool, called *coresets for weighted facilities*.

We define the notion of a (k, ε)-coreset S for weighted facilities for a point set P on a line. We give an algorithm to construct such (k, ε)-coresets of size $2^{O(k)}\varepsilon^{-2k-1}\log^{4k-3} n$ in $O(nk)$ time. Given any set of points $P \subset \mathbb{R}^d$, for fixed $d \geq 1$, we construct these weighted facility coresets for projections of subsets of P onto certain lines, and then combine them to form the desired coreset for P itself. Following the publication of this result in [FFS06], Har-Peled [HP06a] proposed a different, more involved construction of a coreset for k-weighted facility, of size $2^{O(k)}\varepsilon^{-k-1}\log^{k+1} n$.

Formally, let P be a set of weighted points *on a line* ℓ, and let C be a set of weighted facilities (points) in $\mathbb{R}^d$, where each $c \in C$ has some positive multiplicative weight $W(c)$. We define $\nu'_C(P)$ $\big($resp., $\mu'_C(P)\big)$ as the overall sum of the weighted distances (resp., weighted squared distances) from each point to its nearest facility. That is,

$$\begin{aligned} \nu'_C(P) &= \sum_{p \in P} \left(w(p) \cdot \min_{c \in C} \{W(c)\,\|p-c\|\} \right), \text{ and} \\ \mu'_C(P) &= \sum_{p \in P} \left(w(p) \cdot \min_{c \in C} \left\{(W(c)\,\|p-c\|)^2\right\} \right). \end{aligned}$$

Fix k and $\varepsilon > 0$. A (possibly differently) weighted set $S \subseteq P$ is called a (k, ε)-*coreset for weighted facilities*, if *for any* weighted set of k facilities (points)

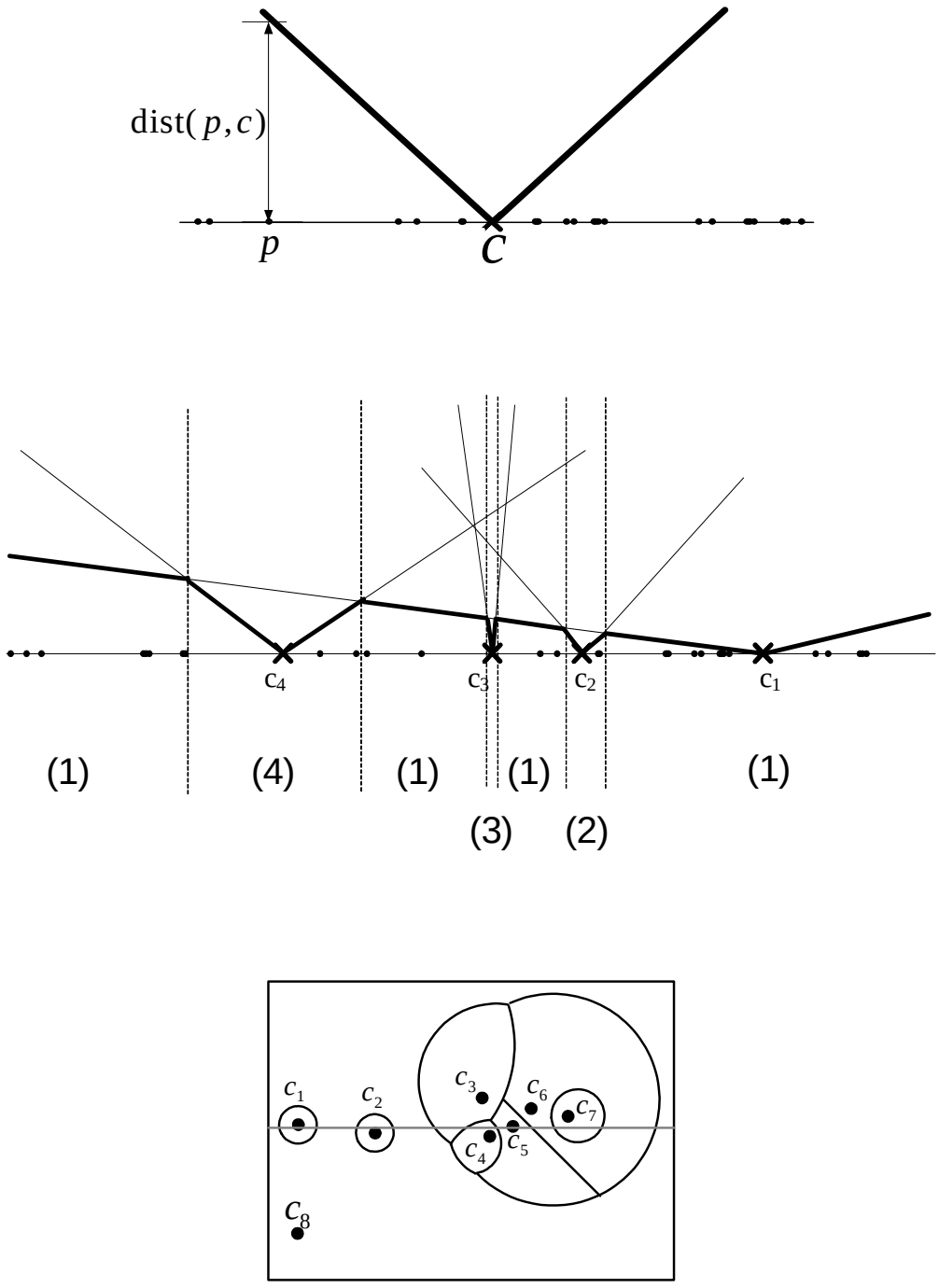


Fig. 2.1: **(top)** The distance function $\mathrm{dist}(p, c)$ from a fixed center c to a point p on the x-axis . ***(middle)*** Four weighted facilities on a line, the lower envelope of their respective weighted distance functions, and their corresponding Voronoi intervals. **(bottom)** Eight weighted facilities in the plane, and the resulting partition of ℓ into 12 Voronoi intervals, induced by their eight planar Voronoi regions.

$C \subset \mathbb{R}^d$, (*i*) and (*ii*) hold:

$$\begin{aligned} (i) \quad & (1-\varepsilon)\nu'_C(P) \leq \nu'_C(S) \leq (1+\varepsilon)\nu'_C(P), \\ (ii) \quad & (1-\varepsilon)\mu'_C(P) \leq \mu'_C(S) \leq (1+\varepsilon)\mu'_C(P). \end{aligned} \tag{2.1}$$

In other words, a coreset for weighted facilities is a (weighted) subset of the input set, so that *for any k facilities*, with *any associated weights*, the sum of minimum weighted (squared or unsquared) distances to the facilities is about the same for the original set and for the subset.

This problem is interesting in its own right, and arises naturally in facility location (see [DH02]). However, we only know how to construct (k, ε)-coresets for weighted facilities when the points of P all lie on a *line* (but the facilities can be anywhere in $\mathbb{R}^d$), and it is open at the moment whether the construction can be extended to arbitrary input sets in $\mathbb{R}^d$, $d \geq 2$.

Nevertheless, (k, ε)-coresets for weighted facilities for point sets on a line, are sufficient for solving optimization problems for generalized facilities of the kinds mentioned above, for arbitrary point sets in $\mathbb{R}^d$. Specifically, they lead to construction of new coresets for generalized facilities, with no restriction on the input set P in $\mathbb{R}^d$.

For a collection of flats Y, let $\mathrm{dist}(p, Y)$, $p \in \mathbb{R}^d$, denote distance from point p to the closest flat $y \in Y$. We obtain coresets for *linear and point facilities* for arbitrary $P \subset \mathbb{R}^d$. That is, given k and ε, the coreset S computed from P has the property that for any (mixed) set Y that contains $0 \leq i \leq k$ lines and at most $k-i$ points in $\mathbb{R}^d$, (*i*) and (*ii*) hold:

$$\begin{aligned} (i) \quad & (1-\varepsilon)\nu_Y(P) \leq \nu_Y(S) \leq (1+\varepsilon)\nu_Y(P) \\ (ii) \quad & (1-\varepsilon)\mu_Y(P) \leq \mu_Y(S) \leq (1+\varepsilon)\mu_Y(P), \end{aligned}$$

where

$$\nu_Y(P) = \sum_{p \in P} \big(w(p) \cdot \mathrm{dist}(p, Y) \big),$$

and

$$\mu_Y(P) = \sum_{p \in P} \left(w(p) \cdot (\mathrm{dist}(p, Y))^2 \right).$$

Thus, this coreset is a generalization of coresets for k-median, and simultaneously, a generalization of coresets for k-mean. Additionally, this coreset approximately preserves distances to both *point* facilities and *line* facilities. It is interesting that, unlike prior constructions (such as [HPM04]), we get the *same*

coreset for both k-mean and k-median (for point and line facilities). However, the significance of our construction mainly lies in its applications to generalized linear facilities.

In addition, for arbitrary input point sets P in $\mathbb{R}^d$, our corresponding coreset S has the property that, for any single j-flat f, with $0 \leq j \leq d-1$, (*i*) and (*ii*) hold:

$$(i) \quad (1-\varepsilon)\nu_{\{f\}}(P) \leq \nu_{\{f\}}(S) \leq (1+\varepsilon)\nu_{\{f\}}(P)$$
$$(ii) \quad (1-\varepsilon)\mu_{\{f\}}(P) \leq \mu_{\{f\}}(S) \leq (1+\varepsilon)\mu_{\{f\}}(P),$$

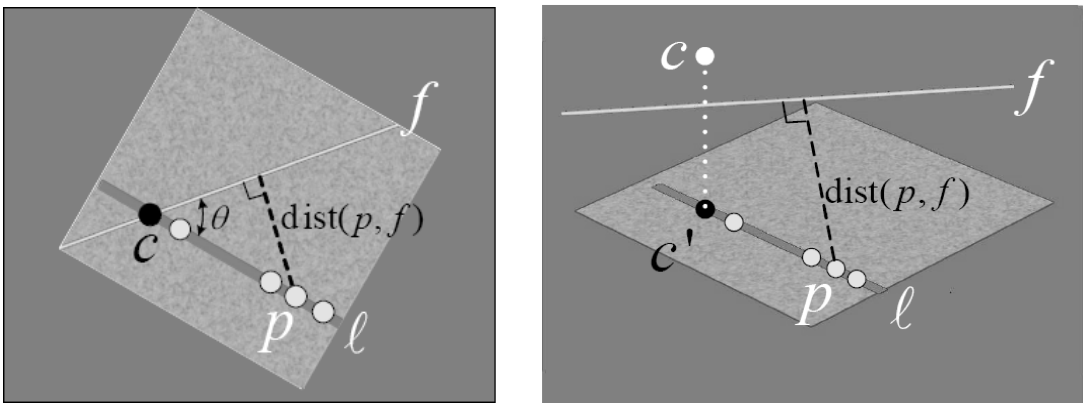
where $\nu_{\{f\}}(\cdot)$, and $\mu_{\{f\}}(\cdot)$ are defined in an analogous manner to the preceding definitions.

Why coresets for weighted facilities?

To motivate the relationship between weighted facilities and linear facilities, consider the following (restrictive) scenario: The (unweighted) input point set P resides on some line $\ell \subset \mathbb{R}^d$, $f \subset \mathbb{R}^d$ is another line, and $\ell \cap f \neq \emptyset$. It follows from elementary trigonometry, that the distance between a point $p \in \ell$ and f is equal to $\|c-p\| \sin\theta$, where c is the point $\ell \cap f$, and θ is the angle formed at c by these two lines. See Fig. 2.2(left).

This simple observation lies at the heart of our work. It extends to arbitrary (skew) lines ℓ and f $\big($see Fig. 2.2(right)$\big)$. I.e., for any lines ℓ and f, such that f is not a translation of ℓ, there exist some weighted point facility $c \in \mathbb{R}^d$ such that the (weighted) distance from any point $p \in \ell$ to c is equal to the distance between p and f. This claim can be further generalized to the case where f is a j-flat, of arbitrary dimension $j \leq d-1$, and also for vertical (regression) distances between points and hyperplanes. See Fig. 2.3. This seemingly suggests a very general transformation. Subject to the restriction that the input point set P be contained in some line, there is a general reduction from any optimization problem that involves distances between points of P and arbitrary j-flats, to another optimization problem that involves distances between the points of P and weighted (point) facilities.

Unfortunately, for general sets of points $P \subset \mathbb{R}^d$, and for an arbitrary linear facility f, there is no point $c \in \mathbb{R}^d$ such that the distance between f and a point $p \in P$ is proportional to the distance between p and c. We show how to overcome this setback by reducing the general case to several subproblems involving points on a line. This machinery is presented in full detail in Chapter 4.



(a) $f \cap \ell \neq \emptyset$ (b) $f \cap \ell = \emptyset$

Fig. 2.2: ***(left)*** $\mathrm{dist}(p, f) = W(c) \cdot \mathrm{dist}(p, c)$, with $W(c) = \sin\theta$. Hence, c, weighted by $\sin\theta$, replaces f for points on ℓ. ***(right)*** $\mathrm{dist}(p, f) = W(c) \cdot \mathrm{dist}(p, c)$ with $W(c) = \sin\theta$, for any pair $(\ell,\ f)$ of lines in $\mathbb{R}^d$, where c is a point on the line that spans the shortest distance between ℓ and f, placed at distance $\mathrm{dist}(\ell, f)/\sin\theta$ from the point $c' \in \ell$, nearest to f, and θ is the angle between the (orientations of the) lines ℓ and f (a routine exercise in stereometry).

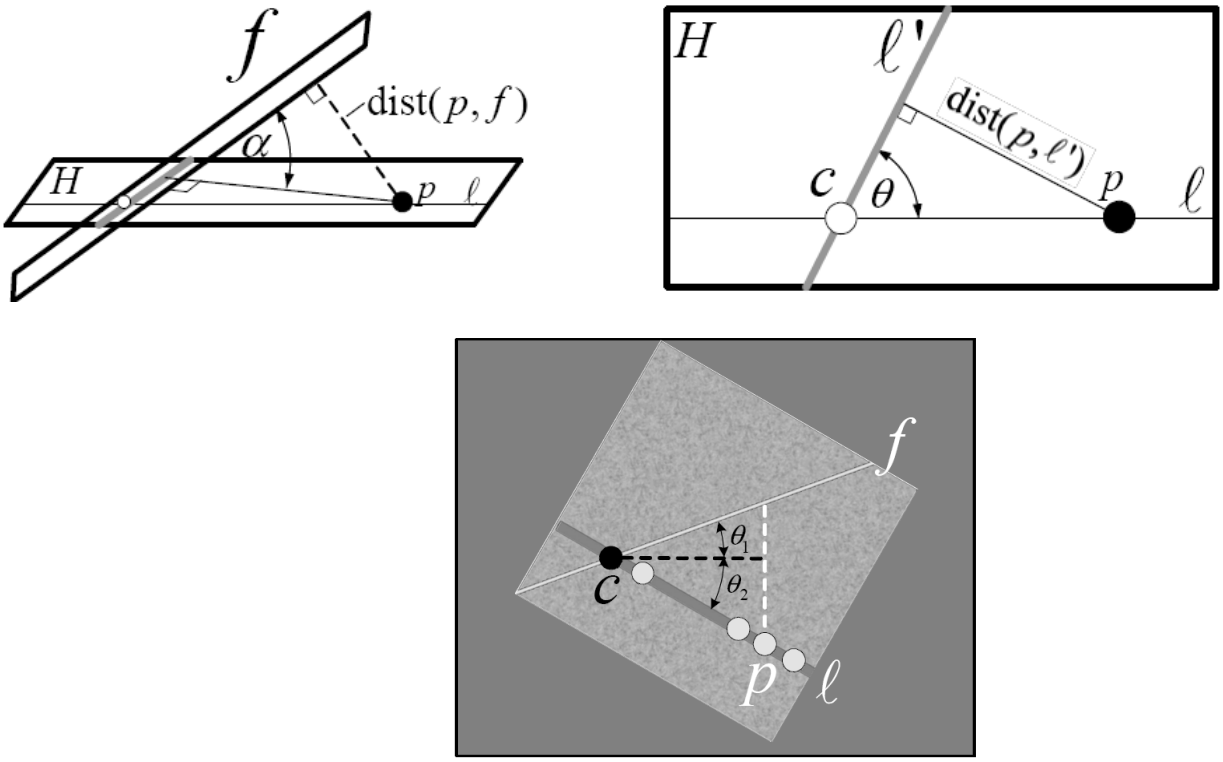


Fig. 2.3: **(top)** The distance from a point p in the plane H to another plane f, is $\mathrm{dist}(p, f) = \sin\alpha \cdot \mathrm{dist}(p, \ell')$, where α is the angle between H and f, and ℓ' is the intersection of H and f. By denoting θ as the angle between ℓ and ℓ', we thus get $\mathrm{dist}(p, f) = W(c)\mathrm{dist}(p, c)$, where $W(c) = \sin\alpha \sin\theta$ and c is the intersection between ℓ and ℓ'. **(bottom)** In the plane, the vertical distance from p to a line f is $W(c)\mathrm{dist}(p, c)$, where c is the intersection between ℓ and f, and $W(c) = \tan\theta_1 \cos\theta_2 + \sin\theta_2$.

Chapter 3

Bi-criteria Linear-time Approximations

3.1 Informal Overview

Recall the setup that we face, as described in Chapters 1 and 2. We have a set $P \subseteq \mathbb{R}^d$ of n points, and two parameters $k \geq 1$, $1 \leq j \leq d-1$, and we seek a small set F of α j-flats so that the value of the objective function (median, mean, or center) at F is not much larger than that at the optimal k j-flats. One can view our algorithm as an instance of the following "meta algorithm" for a bi-criteria projective clustering for input point sets $P \subset \mathbb{R}^d$:

- Choose a set F' of k' j-flats (for some parameter k'), for which there exists a set $P' \subset P$ of size $\geq |P|/2$, such that the value of the objective function (or, rather, of all three objective functions) for F' on P' is no more than c times the value of the optimal k j-flats (for P) on P', for some constant factor c.
- Set $P = P \setminus P'$ and repeat until P is very small, in which case take F' to be the set of all j-flats spanned by P.

As $|P|$ keeps shrinking by factors of 2, this process can be repeated at most $\log |P|$ times. By taking the union of the sets F', we get a set F of $k' \log |P|$ j-flats, for which the value of the objective function, over the entire set P, is off by no more than a factor of c.

In fact, our real algorithm, given below, is very similar to the meta algorithm above, with the following (minor and technical) variations:

- The set F' is simply the set of all j-flats determined by a small set of randomly chosen points from P.
- The set P' consists of the $|P|/2$ points of P that are closest to the flats of F'. Some intuition comes from the argument that *many* of the points near the flats of F' are not much farther from F' than they are to some other (arbitrary) set of k j-flats.
- Unfortunately, not all points "close" to F' have the property that F' is a good approximation to the optimal set of flats; these are "bad" points.
- Fortunately, we can amortize the high contribution to the objective function by these "bad" points against the next round of points to be chosen. The contribution to the objective function, appropriately scaled, of the good points of the next round will dominate that of the current "bad" points.

3.2 The Algorithm

We first briefly review some notation. For a $(j+1)$-tuple $X = (p_1, \cdots p_{j+1})$ of $j+1$ (not necessarily distinct) points in $\mathbb{R}^d$, we denote by $\mathrm{flat}(X)$ a j-flat that passes through all the points of X. If there is more than one such flat, we choose one of them arbitrarily. For $k \geq 1$, we denote by $\mathbb{F}(k, j, d)$ the collection of all sets of at most k flats in $\mathbb{R}^d$, each of dimension at most j. For a j-flat f and a point p in $\mathbb{R}^d$, we denote by $\mathrm{dist}(p, f)$ the minimum Euclidean distance from p to f. For a set of flats F, we denote by $\mathrm{dist}(p, F) = \min_{f \in F} \mathrm{dist}(p, f)$ the distance of p to its nearest flat in F.
The pseudo-code of our bi-criteria approximation algorithm is given in Figure 3.1.

Theorem 3.1. *Let P be a set of n points in $\mathbb{R}^d$, and k, j integers, such that $k \geq 1$ and $1 \leq j \leq d-1$. Then the procedure* APPROX-K-J-FLATS(P, k, j)*, given in Figure 3.1, returns a set F of $\log n \cdot (jk \log\log n)^{O(j)}$ j-flats, such that, with probability at least $1/2$, we have, for every integer $v \geq 1$,*

$$\sum_{p \in P} \big(\mathrm{dist}(p, F)\big)^v \leq 2^{v(j+1)+1} \min_{F^* \in \mathbb{F}(k,j,d)} \sum_{p \in P} \mathrm{dist}(p, F^*)$$

$$\max_{p \in P} \mathrm{dist}(p, F) \leq 2^{j+1} \min_{F^* \in \mathbb{F}(k,j,d)} \max_{p \in P} \mathrm{dist}(p, F^*).$$

The running time of this procedure is $O(dn) \cdot (jk)^{O(j)}$.

Algorithm APPROX-K-J-FLATS(P, k, j)
Input. A set of n points $P \subset \mathbb{R}^d$, and two integers $k \geq 1$, $1 \leq j \leq d-1$.
Output. A set of j-flats F that satisfies Theorem 3.2.

$t \leftarrow 1$, $Q \leftarrow P$, $F \leftarrow \emptyset$
while $|Q| \geq 32k(j+1)$
  for $i \leftarrow 0$ to j
    Pick a random sample S_i of
    $\lceil 32k(j+1)\big(2 + \log(j+1) + \log k + \min\{t, \log\log n\}\big)\rceil$ i.i.d. points from Q.
  $F' \leftarrow \{\text{flat}(X) \mid X \in S_0 \times S_1 \times \cdots \times S_j\}$.
  Compute a set $R_t \subseteq Q$ of the closest $\lceil |Q|/2 \rceil$ points to F',
  where ties are broken arbitrarily.
  $F \leftarrow F \cup F'$
  $Q \leftarrow Q \setminus R_t$
  $t \leftarrow t+1$
$F \leftarrow F \cup \{\text{flat}(X) \mid X \in Q^{j+1}\}$
$t_{\max} \leftarrow t$, $R_{t_{\max}} \leftarrow Q$ (used only for analysis)
return F

Fig. 3.1: The bi-criteria algorithm APPROX-K-J-FLATS.

The proof of Theorem 3.1 relies on the following main technical result.

Theorem 3.2. *Let P be a set of n points in $\mathbb{R}^d$, and k, j integers, such that $k \geq 1$ and $1 \leq j \leq d-1$. Let F be the set of flats that is returned by the bi-criteria approximation algorithm* APPROX-K-J-FLATS(P, k, j) *(see Fig. 3.1). For an arbitrary set of flats $F^* \in \mathbb{F}(k, j, d)$, define*

$$P_{\text{bad}} = \{p_{\text{bad}} \in P \mid \text{dist}(p_{\text{bad}}, F) > 2^{j+1}\text{dist}(p_{\text{bad}}, F^*)\}.$$

Then, with probability at least $1/2$, we can map each point $p_{\text{bad}} \in P_{\text{bad}}$ to a distinct point $p \in P \setminus P_{\text{bad}}$, such that $\text{dist}(p_{\text{bad}}, F) \leq 2^{j+1}\text{dist}(p, F^*)$.

We defer the proof of Theorem 3.2 to Section 3.3.

Proof of Theorem 3.1. Let F^* be an arbitrary set of flats in $\mathbb{F}(k, j, d)$. Assuming Theorem 3.2 holds, we now conclude the proof of Theorem 3.1. Then we have, with probability at least $1/2$, for all $v \geq 1$, that

$$\begin{aligned}
\sum_{p \in P} \big(\text{dist}(p, F)\big)^v &= \sum_{p \in P \setminus P_{\text{bad}}} \big(\text{dist}(p, F)\big)^v + \sum_{b \in P_{\text{bad}}} \big(\text{dist}(b, F)\big)^v \\
&\leq \sum_{p \in P \setminus P_{\text{bad}}} \Big(\big(\text{dist}(p, F)\big)^v + 2^{v(j+1)}\big(\text{dist}(p, F^*)\big)^v\Big) \\
&\leq 2^{v(j+1)+1} \sum_{p \in P} \big(\text{dist}(p, F^*)\big)^v,
\end{aligned}$$

where the first inequality follows from Theorem 3.2, and the second inequality follows from the definition of $P \setminus P_{\text{bad}}$. In particular, $v = 1$ implies the case of sum of distances and $v = 2$ implies the case of sum of squares. The same arguments imply that

$$\begin{aligned}
\max_{p \in P} \text{dist}(p, F) &= \max\left\{\max_{p \in P \setminus P_{\text{bad}}} \text{dist}(p, F), \max_{b \in P_{\text{bad}}} \text{dist}(b, F)\right\} \\
&\leq \max_{p \in P \setminus P_{\text{bad}}} \left\{\text{dist}(p, F), 2^{j+1}\text{dist}(p, F^*)\right\} \\
&\leq 2^{j+1} \max_{p \in P} \text{dist}(p, F^*).
\end{aligned}$$

We next analyze the size of F and the time for its construction. Since the size of Q is reduced by at least half in each iteration, we have $t_{\max} - 1 \leq \log n$

iterations. In line 10, at most $\big(32k(j+1)\big)^{j+1}$ flats are added to F (as $|Q| < 32k(j+1)$ by Line 2). The overall size of the output set of flats is

$$\begin{aligned}
&\sum_{t=1}^{t_{\max}-1} \lceil 32k(j+1)\big(2+\log(j+1)+\log k+\min\{t,\log\log n\}\big)\rceil^{j+1} \\
&\qquad + \big(32k(j+1)\big)^{j+1} \\
&= \sum_{t=1}^{t_{\max}-1} \big(O(jk)\cdot(jk+\log\log n)\big)^{j+1} \\
&= \log n\cdot(jk\log\log n)^{O(j)}.
\end{aligned}$$

The running time of the t^{th} iteration is dominated by the running time of Line 6 which, using the simplest algorithm that goes point by point, takes time

$$\begin{aligned}
O(d\,|Q|\cdot|F'|) &= O(dn/2^t)\cdot\big(32k(j+1)(2+\log(j+1)+\log k+t)\big)^{j+1} \\
&= O(dn/2^t)\cdot(64kj)^{j+1}\cdot\big(2+\log(2jk)+t\big)^{j+1}\ .
\end{aligned}$$

Summing this over all iterations t, we get a sum of the form

$$O\left(dn\cdot(64jk)^{j+1}\sum_{t\geq 1}\frac{\big(2+\log(2jk)+t\big)^{j+1}}{2^t}\right) = dn\cdot f(j,k),$$

where

$$\begin{aligned}
f(j,k) &= (2jk)^{O(j)}\sum_{t\geq 1}\frac{\big(2+t+\log(2jk)\big)^{j+1}}{2^t} \\
&= (2jk)^{O(j)}\sum_{t=1}^{\log(2jk)}\frac{\big(2\log(jk)\big)^{j+1}}{2^t} + (jk)^{O(j)}\sum_{t\geq\log(jk)+1}\frac{t^{j+1}}{2^t} \\
&= (2jk)^{O(j)}\left[\big(\log(2jk)\big)^{j+1}+j^{j+1}\right] = (2jk)^{O(j)}.
\end{aligned}$$

This concludes the proof of Theorem 3.1. □

The probability that the resulting set F of APPROX-K-J-FLATS satisfies the inequalities of Theorem 3.1 can be made arbitrarily close to 1, by running APPROX-K-J-FLATS repeatedly x times with independent random choices each time. Then we take the three sets which respectively minimize the three expressions in Theorem 3.1, over all the x runs. The union of these sets will satisfy all three inequalities, with probability at least $1-1/2^x$.

3.3 Proof of Theorem 3.2

We first provide a brief overview of the proof. It begins with Lemma 3.3, which is a simple probabilistic lemma, giving a bound on the size of a random sample from a set Q that guarantees, with high probability, that it hits each of k given subsets of Q of some given size.

Lemma 3.4 says that if we choose an arbitrary line ℓ through the origin, and a line $\mathrm{sp}(b)$ connecting some arbitrary point b to the origin, then for all points whose angle with ℓ is greater than the angle between $\mathrm{sp}(b)$ and ℓ, the distance to $\mathrm{sp}(b)$ is at most a constant factor times the distance to ℓ. This simple observation is later generalized to higher-dimensional flats in Lemma 3.6.

Lemma 3.7 deals with one iteration of the algorithm. It uses the preceding lemmas to argue that the set of flats F' chosen by the algorithm has the property that the set of bad points (points close to F' that are much closer to F^*) is small.

Finally, the proof amortizes the contribution of the (relatively few) bad points against the contribution of other good points in subsequent steps of the algorithm, concluding the proof of the theorem.

Lemma 3.3. *Let Q be a set of m points, $k \geq 1$ an integer, and $c > k$, $1 \leq \beta \leq m$ parameters. Let $Q_1, Q_2, \ldots, Q_k$ be any k subsets of Q, each containing β points. Assume that we pick at least $(m/\beta)\ln c$ random independent samples from Q (with or without repetitions). Then the probability that every one of the subsets contains a sample point is at least $1 - k/c$.*

Proof. The probability that the first sampled point is not contained in Q_1 is $1 - \beta/m$. Therefore, the probability that none of the sampled points are in Q_1 is at most

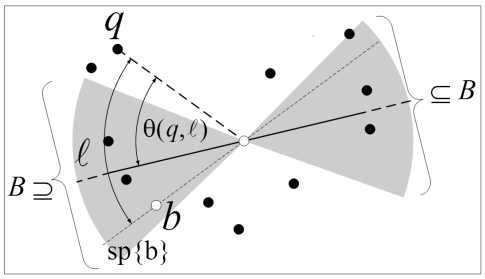
$$\left(1 - \frac{\beta}{m}\right)^{(m/\beta)\ln c} \leq e^{-\ln c} = \frac{1}{c}.$$

The same holds for each Q_i, $1 \leq i \leq k$. Hence, the probability that at least one of the Q_i does not intersect the sample is at most k/c. □

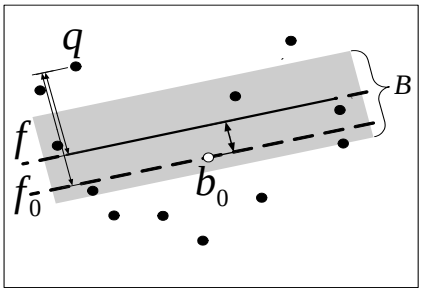
In the following analysis, we use the notation $\mathrm{sp}(X)$ for the linear span of a set X; when X is a singleton b, the shorthand notation $\mathrm{sp}(b)$ thus denotes the line through b and the origin.

Lemma 3.4. *Let ℓ be a line in $\mathbb{R}^d$ that passes through the origin. Let Q be a set of points in $\mathbb{R}^d$. Then, for any natural number $\beta \leq |Q|$ there is a set $B \subseteq Q$ of β points, such that for all $b \in B$ and $q \in Q \setminus B$*

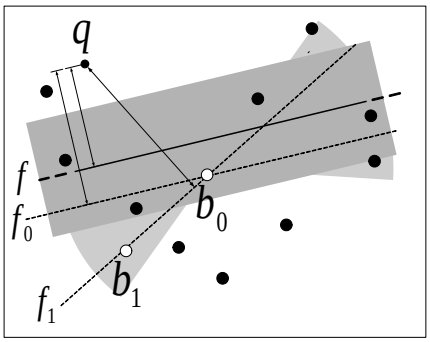
$$\mathrm{dist}(q, \mathrm{sp}(b)) \leq 2\mathrm{dist}(q, \ell).$$



(a) $\theta(q, \mathrm{sp}(b)) \leq 2\theta(q, \ell)$



(b) $\mathrm{dist}(q, f_0) \leq 2\mathrm{dist}(q, f)$



(c) $\mathrm{dist}(q, f_1) \leq 4\mathrm{dist}(q, f)$

Fig. 3.2: The case of one line in the plane ($d = 2$, $j = 1$). **(a)** The set B contains the β points in the gray areas. **(b)** The set B consists of the β points of P closest to f. **(c)** $\mathrm{dist}(q, f_1) \leq 2\mathrm{dist}(q, f_0) \leq 4\mathrm{dist}(q, f)$ for every q outside the gray area ($q \in Q \setminus Q_{\mathrm{bad}}$).

Proof. For a point $q \in Q$, denote by $\theta(q, \ell)$ the acute angle formed by the lines $\mathrm{sp}(q)$ and ℓ; see Figure 3.2(a) for the planar case. Let $B \subseteq Q$ be the set consisting of the β points q with the smallest values of $\theta(q, \ell)$, and let $b \in B$. For $q \in Q \setminus B$ we thus have $\theta(b, \ell) \leq \theta(q, \ell)$, and therefore

$$\theta(q, \mathrm{sp}(b)) \leq \theta(q, \ell) + \theta(b, \ell) \leq 2\theta(q, \ell),$$

or $\theta(q, \mathrm{sp}(b))/2 \leq \theta(q, \ell)$, which implies that

$$\begin{aligned}\sin\theta(q, \mathrm{sp}(b)) &= 2\sin\frac{\theta(q, \mathrm{sp}(b))}{2}\cos\frac{\theta(q, \mathrm{sp}(b))}{2} \\ &\leq 2\sin\frac{\theta(q, \mathrm{sp}(b))}{2} \leq 2\sin\theta(q, \ell).\end{aligned}$$

The distance from q to $\mathrm{sp}(b)$ can then be bounded by

$$\begin{aligned}\mathrm{dist}(q, \mathrm{sp}(b)) &= \|q\|\sin\theta(q, \mathrm{sp}(b)) \\ &\leq \|q\| \cdot 2\sin\theta(q, \ell) = 2\mathrm{dist}(q, \ell).\end{aligned}$$

□

Lemma 3.5. *Let $f = \mathrm{sp}(v_1, \ldots, v_{j-1}, v_j)$ be a j-dimensional subspace of $\mathbb{R}^d$, for some given tuple of j mutually orthogonal unit vectors $v_1, \ldots, v_j$.*
Let $\{v_{j+1}, \ldots, v_d\}$ be a set of mutually orthogonal unit vectors that span the subspace orthogonal to f.
Let $q \in \mathbb{R}^d$, and let q' denote the projection of q on the subspace $M = \mathrm{sp}(v_1, v_{j+1}, v_{j+2}, \ldots, v_d)$. Then

$$\mathrm{dist}(q, f) = \mathrm{dist}(q', \mathrm{sp}(v_1)).$$

Proof. Without loss of generality, we assume that $v_1, \ldots, v_d$ is the standard base of $\mathbb{R}^d$, where v_i is a unit vector in the x_i-direction, $1 \leq i \leq d$. This can always be enforced by an appropriate rotation of the coordinate frame. Hence,

$$\mathrm{dist}(q, f) = \sqrt{\sum_{i=j+1}^{d} q_i^2} = \sqrt{\sum_{i=2}^{d} (q_i')^2} = \mathrm{dist}(q', \mathrm{sp}(v_1)).$$

□

Lemma 3.6. *Let Q be a set of n points in $\mathbb{R}^d$, and $f = \mathrm{sp}(v_1, \ldots, v_{j-1}, v_j)$ be a j-dimensional subspace of $\mathbb{R}^d$, for some given tuple of j mutually orthogonal unit vectors $v_1, \ldots, v_j$. Then, for any natural number $\beta \leq n$, there exists a subset $\mathcal{Z} \subseteq Q$ of β points, such that for every point $b \in \mathcal{Z}$, and the corresponding subspace $f(b) = \mathrm{sp}(b, v_2, v_3, \ldots, v_j)$, we have*

$$\mathrm{dist}\big(q, f(b)\big) \leq 2\mathrm{dist}(q, f),$$

for all $q \in Q \setminus \mathcal{Z}$.

Proof. Let $\{v_{j+1}, \ldots, v_d\}$ be a set of mutually orthogonal unit vectors that span the subspace orthogonal to f. For a point $x \in \mathbb{R}^d$ we denote by x' the projection of x onto the subspace $M = \mathrm{sp}(v_1, v_{j+1}, v_{j+2}, \ldots, v_d)$. For a set $X \subseteq \mathbb{R}^d$, we define $X' = \{x' \mid x \in X\}$.

By applying Lemma 3.4 with $\ell = \mathrm{sp}(v_1') = \mathrm{sp}(v_1)$ (by construction, $v_1' = v_1$), and with the projection set Q' as the set Q in that lemma, we conclude that for any natural number $\beta \leq n$ there exists a set $B' \subseteq Q'$ of β points, such that for every $b' \in B'$, the corresponding line $\mathrm{sp}(b')$ satisfies

$$\mathrm{dist}(q', \mathrm{sp}(b')) \leq 2\mathrm{dist}(q', \mathrm{sp}(v_1')) = 2\mathrm{dist}(q', \mathrm{sp}(v_1)), \tag{3.1}$$

for all $q' \in Q' \setminus B'$.

We define B to be the set of those $b \in Q$ such that $b' \in B'$. We claim that for each point $b \in B$, its corresponding subspace $f(b) = \mathrm{sp}(b, v_2, v_3, \ldots, v_j)$ satisfies $\mathrm{dist}\big(q, f(b)\big) \leq 2\mathrm{dist}(q, f)$, for all $q \in Q \setminus B$. Indeed, let q be any point in $Q \setminus B$. Since $b - b' \in \mathrm{sp}(v_2, v_3, \ldots, v_j)$, we have

$$f(b) = \mathrm{sp}(b, v_2, v_3, \ldots, v_j) = \mathrm{sp}(b', v_2, \ldots, v_j) = \mathrm{sp}(b'/\left\|b'\right\|, v_2, \ldots, v_j).$$

Using this equation, applying Lemma 3.5 with $f = f(b)$ and $v_1 = b'/\left\|b'\right\|$ yields

$$\mathrm{dist}\big(q, f(b)\big) = \mathrm{dist}(q', \mathrm{sp}(b'/\left\|b'\right\|)) = \mathrm{dist}(q', \mathrm{sp}(b')).$$

Similarly, by Lemma 3.5 (with the original f and v_1) we have

$$\mathrm{dist}(q, f) = \mathrm{dist}(q', \mathrm{sp}(v_1)).$$

Using the last two equations and Equation (3.1) gives us

$$\mathrm{dist}\big(q, f(b)\big) = \mathrm{dist}(q', \mathrm{sp}(b')) \leq 2\mathrm{dist}(q', \mathrm{sp}(v_1)) = 2\mathrm{dist}(q, f),$$

which completes the proof of Lemma 3.6. □

Lemma 3.7. *Let F^* be a set of k arbitrary j-flats in $\mathbb{R}^d$, where $k \geq 1$ and $1 \leq j \leq d-1$. Consider the sets Q and F' at the t^{th} iteration of* APPROX-K-J-FLATS(P, k, j)*, and define*

$$Q_{\mathrm{bad}} = \{q \in Q \mid \mathrm{dist}(q, F') > 2^{j+1}\mathrm{dist}(q, F^*)\}.$$

Then $|Q_{\mathrm{bad}}| \leq |Q|/16$ with probability at least $1 - 2^{-2-\min\{t, \log\log n\}}$.

Proof. For a j-flat $f \in F^*$, let $B \subset Q$ be the set of the $\beta = \lfloor |Q| / \big(16k(j+1)\big) \rfloor$ points of Q closest to f, where ties are broken arbitrarily; see Fig. 3.2(b). Fix a point $b_0 \in B$, and let f_0 be the j-flat that is parallel to f and passes through b_0. Note that for every point $q \in Q \setminus B$ we have $\mathrm{dist}(b_0, f) \leq \mathrm{dist}(q, f)$ by definition of B. Thus,

$$\mathrm{dist}(q, f_0) \leq \mathrm{dist}(q, f) + \mathrm{dist}(b_0, f) \leq 2\mathrm{dist}(q, f). \tag{3.2}$$

Without loss of generality, we assume that the point b_0 is the origin, and $f_0 = \mathrm{sp}(v_1, v_2, \ldots, v_{j-1}, v_j)$, for an appropriate set of j mutually orthogonal vectors $v_1, \cdots, v_j$. By Lemma 3.6, there exists a set $B(b_0) \subseteq Q$ of β points, such that for every $b_1 \in B(b_0)$, and the corresponding j-flat $f_1 = \mathrm{sp}(b_1, v_2, \ldots, v_j)$, we have

$$\mathrm{dist}(q, f_1) \leq 2\mathrm{dist}(q, f_0) \leq 4\mathrm{dist}(q, f) \tag{3.3}$$

for all $q \in Q \setminus B(b_0)$; see Fig. 3.2(c).

Fix a point $b_1 \in B(b_0)$. By substituting $f = f_1$ in Lemma 3.6, we conclude that there is a set $B(b_0, b_1) \subseteq Q$ of β points, such that for every $b_2 \in B(b_0, b_1)$, and the corresponding j-flat $f_2 = \mathrm{sp}(b_1, b_2, v_3, v_4, \ldots, v_j)$, we have

$$\mathrm{dist}(q, f_2) \leq 2\mathrm{dist}(q, f_1),$$

for all $q \in Q \setminus B(b_0, b_1)$. Combining (3.3) with the last equation yields

$$\mathrm{dist}(q, f_2) \leq 2\mathrm{dist}(q, f_1) \leq 8\mathrm{dist}(q, f),$$

for all $q \in Q \setminus \big(B \cup B(b_0) \cup B(b_0, b_1)\big)$.

Similarly, by induction, for every j-flat $f \in F^*$, and $0 \leq i \leq j$, there is a set $B_f(b_0^f, b_1^f, \ldots, b_{i-1}^f) \subseteq Q$ of β points (for $i = 0$, we denote the set simply as B_f), such that for every $b_i^f \in B_f(b_0^f, b_1^f, \ldots, b_{i-1}^f)$, and the corresponding j-flat $f_i = b_0^f + \mathrm{sp}(b_1^f, b_2^f, \ldots, b_i^f, v_{i+1}^f, \ldots, v_j^f)$, we have

$$\mathrm{dist}(q, f_i) \leq 2^{i+1}\mathrm{dist}(q, f), \tag{3.4}$$

for all $q \in Q \setminus \bigcup_{0 \leq i \leq j} B_f(b_0^f, \ldots, b_{i-1}^f)$.

Consider the set S_i for every $1 \leq i \leq j$, as defined in Line 4 of Fig 3.1). We claim that with probability at least $1 - 2^{-2-\min\{t, \log\log n\}}$, for each $f \in F^*$ and $0 \leq i \leq j$, the set S_i contains a point $b_i^f \in B_f(b_0^f, b_1^f, \ldots, b_{i-1}^f)$. Indeed, we have k sets B_f, of size β each, for $f \in F^*$. Lemma 3.3 shows that if we sample at least

$\frac{|Q|}{\beta} \ln c$ points from Q, the probability that at least one of the sets B_f will not contain any sample point is at most k/c. Let $c = 2^{2+\log(j+1)+\log k+\min\{t,\log\log n\}}$, and note that, by Line 2 of APPROX-K-J-FLATS, we have $|Q| / \big(32k(j+1)\big) \geq 1$, so

$$\beta = \lfloor |Q| / \big(16k(j+1)\big) \rfloor \geq |Q| / \big(32k(j+1)\big).$$

Hence

$$(|Q|/\beta) \ln c \leq \lceil 32k(j+1)\big(2 + \log(j+1) + \log k + \min\{t, \log\log n\}\big) \rceil = |S_0|,$$

and thus the probability that S_0 misses at least one of the sets B_f is at most

$$\begin{aligned} k/c &= k/2^{2+\log(j+1)+\log k+\min\{t,\log\log n\}} \\ &\leq 2^{-2-\log(j+1)-\min\{t,\log\log n\}}. \end{aligned}$$

Assume that this event does not arise (which happens with probability at least $1 - 2^{-2-\log(j+1)-\min\{t,\log\log n\}}$). Pick a point $b_0^f \in B_f \cap S_0$ for each $f \in F^*$, and consider the k sets $B_f(b_0^f)$, $f \in F^*$. As in the case for S_0, it can be shown that S_1 misses at least one of the sets $B_f(b_0^f)$ with probability at most $k/c \leq 2^{-2-\log(j+1)-\min\{t,\log\log n\}}$.

By repeating this process, we conclude that, for every $f \in F^*$, the set $S_0 \times S_1 \times \ldots \times S_j$ contains a $(j+1)$-tuple $b_0^f, b_1^f, \ldots, b_j^f$ such that $b_i^f \in B_f(b_0^f, \ldots, b_{i-1}^f)$ for each $0 \leq i \leq j$, with probability at least

$$1 - \frac{(j+1)k}{c} \geq 1 - (j+1) \cdot 2^{-2-\log(j+1)-\min\{t,\log\log n\}} \geq 1 - 2^{-2-\min\{t,\log\log n\}}.$$

This implies that, with the same probability, F' contains a j-flat f_j that passes through $b_0^f, b_1^f, \ldots, b_j^f$ for every $f \in F^*$. Refer to this event as E, and assume that it occurs. In this case, by (3.4), $\mathrm{dist}(q, f_j) \leq 2^{j+1}\mathrm{dist}(q, f)$ for all $q \in Q \setminus \bigcup_{0 \leq i \leq j} B_f(b_0^f, \ldots, b_{i-1}^f)$, where b_i^f is one of the points in $S_i \cap B_f(b_0^f, \ldots, b_{i-1}^f)$ which, since we assume that E occurs, is nonempty. Hence,

$$Q_{\mathrm{bad}} \subseteq \bigcup_{f \in F^*} \bigcup_{0 \leq i \leq j} B_f(b_0^f, \ldots, b_{i-1}^f).$$

Since, by construction, each of the sets in the union is of size β, we get

$$\begin{aligned} |Q_{\mathrm{bad}}| &\leq \left| \bigcup_{f \in F^*} \bigcup_{0 \leq i \leq j} B_f(b_0^f, \ldots, b_{i-1}^f) \right| \\ &\leq (j+1)k\beta \leq |Q|/16 \end{aligned} \tag{3.5}$$

with probability at least $1 - 2^{-2-\min\{t,\log\log n\}}$.

This completes the proof of Lemma 3.7. □

Now we are ready to prove Theorem 3.2.

Proof of Theorem 3.2. Note that $(R_1, R_2, \ldots, R_{t_{\max}})$ is a partition of P, and for every $p \in R_{t_{\max}}$ we have $\mathrm{dist}(p, F) = 0$, by Line 10 (i.e., $P_{\mathrm{bad}} \cap R_{t_{\max}} = \emptyset$). Thus,

$$P_{\mathrm{bad}} = \bigcup_{1 \le t \le t_{\max}-1} P_{\mathrm{bad}} \cap R_t. \tag{3.6}$$

Consider the sets Q and F' at the t^{th} iteration, for some $1 \le t \le t_{\max} - 1$, of APPROX-K-J-FLATS, and define

$$Q_{\mathrm{bad}} = \{p_{\mathrm{bad}} \in Q \mid \mathrm{dist}(p_{\mathrm{bad}}, F') > 2^{j+1}\mathrm{dist}(p_{\mathrm{bad}}, F^*)\}.$$

We first prove that, with probability at least $1 - 2^{-2-\min\{t,\log\log n\}}$, we have

$$|Q_{\mathrm{bad}} \cap R_t| \le |R_{t+1} \setminus Q_{\mathrm{bad}}|. \tag{3.7}$$

Indeed, in Lemma 3.7 we proved that, with probability at least $1-2^{-2-\min\{t,\log\log n\}}$, we have $|Q_{\mathrm{bad}}| \le |Q|/16$. By Line 2, $|Q| \ge 20$, so, by definition of R_{t+1} we have $|Q|/5 \le \lfloor |Q|/4 \rfloor \le |R_{t+1}|$. Hence,

$$\begin{aligned} |Q_{\mathrm{bad}} \cap R_t| \le |Q_{\mathrm{bad}}| \le |Q|/16 < |Q|/5 - |Q|/16 \\ \le |R_{t+1}| - |Q_{\mathrm{bad}}| \le |R_{t+1} \setminus Q_{\mathrm{bad}}|, \end{aligned} \tag{3.8}$$

with probability at least $1 - 2^{-2-\min\{t,\log\log n\}}$.

Since $F \supseteq F'$, and every point in R_t is closer to F' than any point in R_{t+1}, we have by (3.7) that we can map each point $b \in Q_{\mathrm{bad}} \cap R_t$ to a distinct point $p_b \in R_{t+1} \setminus Q_{\mathrm{bad}}$, such that

$$\mathrm{dist}(b, F) \le \mathrm{dist}(b, F') \le \mathrm{dist}(p_b, F') \le 2^{j+1}\mathrm{dist}(p_b, F^*).$$

Note that $P_{\mathrm{bad}} \cap R_t \subseteq Q_{\mathrm{bad}} \cap R_t$, because R_t is a subset of the present Q, and if an element of Q is in P_{bad} then it is also in Q_{bad} (because $\mathrm{dist}(b, F') \ge \mathrm{dist}(b, F)$). Similarly, we have $R_{t+1} \setminus Q_{\mathrm{bad}} \subseteq R_{t+1} \setminus P_{\mathrm{bad}}$. We thus conclude that, with probability at least $1 - 2^{-2-\min\{t,\log\log n\}}$, we can map each point $p_{\mathrm{bad}} \in P_{\mathrm{bad}} \cap R_t$

to a distinct point $p_b \in R_{t+1} \setminus P_{\text{bad}}$ such that $\text{dist}(b, F) \leq 2^{j+1}\text{dist}(p_b, F^*)$. Thus, the probability that this holds for all the $t_{\max} - 1 \leq \log n$ iterations is at least

$$\begin{aligned}
&1 - \sum_{t=1}^{t_{\max}} 2^{-2-\min\{t,\log\log n\}} \\
&= 1 - \sum_{t=1}^{\lfloor \log\log n \rfloor} 2^{-2-t} - \sum_{t=\lfloor \log\log n \rfloor + 1}^{t_{\max}} 2^{-2-\log\log n} \\
&\geq 1 - \frac{1}{4} - \frac{\log n}{2^{2+\log\log n}} = \frac{1}{2}.
\end{aligned}$$

Using (3.6), this concludes the proof of Theorem 3.2. □

Chapter 4

Coresets for Weighted and Linear Facilities

In this chapter we deal with constructing coresets for weighted and linear facilities. We also obtain, using the same construction with somewhat different parameters, coresets for a single j-flat.

We are given a set P of points in R^d as input. We seek to answer queries of the form: what is the sum of distances from P to a set $Q = \{\ell_1, \ell_2, \ldots, \ell_k\}$ of k lines (in R^d). We want to be able to answer such queries by replacing P by a smaller coreset, so that the answer for the coreset will be a good approximation to the answer for P. To construct such coresets efficiently, we make use of the seemingly much simpler problem where the points of P are not in a general position but are all co-located on some line in R^d. Dealing with more general point sets P is possible by choosing a (relatively) small number of lines and then projecting each point of P onto its closest line, solving the problem for the projected points, and then taking the union of all such coresets.

Coresets for one j-flat allow us to answer approximate queries of the form: what is the distance from P to a set Q?, where Q is an affine space of dimension j (j-flat).

Surprisingly, we show a strong connection between the problem of a coreset for k lines in R^d, when the input points of P are co-located on a line, to another problem, that of coresets for weighted (point) facilities. The problem of coresets for weighted facilities assumes that the input points of P are on a line, and that the query, Q, is a set of k weighted points, not necessarily on the line, where the query output is the sum, over all points in $p \in P$, of the minimum of the weighted Euclidean distances from p to the points $q \in Q$, or the sum of squares of these

distances, where every such distance is multiplicatively weighted by the weight of q.

We start with a technical lemma, Lemma 4.1, which we later use in two different contexts, that of a coreset for point queries, and to estimate the errors when projecting points onto a collection of lines.

Most of this chapter deals with the conversion from k line queries to k weighted point queries. Furthermore, we convert the problem of weighted point queries to something we call V-coresets that deals with weighted point queries by considering the number of intervals along a line in the arrangement of the Voronoi regions implied by weighted points. So, the parameter is not the number of points (k) but the number of intervals induced by the k Voronoi regions. See Lemmas 4.5 and 4.9.

To get coresets for k lines, we project the points of P onto many lines and compute coresets for weighted facilities separately for the projected points on each line. To get coresets for a single j-flat, we project the points of P onto several j-flats, and continue recursively, taking the union of the outputs to all these subproblems as the final coreset.

4.1 ε-Coresets For a Single Facility

Let P be a weighted set of points in $\mathbb{R}^d$ and $0 < \varepsilon \leq 1$. We recall some notations introduced in Chapter 1. For a set $C \subseteq \mathbb{R}^d$, we define

$$\nu_C(P) = \sum_{p \in P} \big(w(p) \cdot \operatorname{dist}(p, C)\big),$$

and

$$\mu_C(P) = \sum_{p \in P} \left(w(p) \cdot (\operatorname{dist}(p, C))^2\right).$$

A weighted set $S \subseteq P$ is called an *ε-coreset for a single facility* if, for every facility (point) $c \in \mathbb{R}^d$, (*i*) and (*ii*) hold:

$$(i) \quad (1-\varepsilon)\nu_{\{c\}}(P) \leq \nu_{\{c\}}(S) \leq (1+\varepsilon)\nu_{\{c\}}(P) \tag{4.1}$$

$$(ii) \quad (1-\varepsilon)\mu_{\{c\}}(P) \leq \mu_{\{c\}}(S) \leq (1+\varepsilon)\mu_{\{c\}}(P).$$

Note that an ε-coreset for a single facility approximates the sum (or sum of squares) of distances also in the case that the facility itself is also weighted.

Algorithm SINGLE-FACILITY-CORESET(P, ε)
Input. A set of n points $P \subset \mathbb{R}^d$, and $\varepsilon > 0$.
Output. A single facility 9ε-coreset for P.

0 **if** $nd < \frac{1}{\varepsilon} \log n$
  then return P
1 $W \leftarrow \sum_{p \in P} w(p)$; $\quad \overline{p} \leftarrow \sum_{p \in P} \frac{w(p)}{W} \cdot p$;
  $T \leftarrow \frac{1}{W} \sum_{p \in P} \|p - \overline{p}\|$; $\quad S \leftarrow \emptyset$
2 **for** $j \leftarrow 1$ to $\lceil \log W \rceil$
3 **do** $B_j \leftarrow$ the closed ball in $\mathbb{R}^d$ with radius $2^{j-1}T$ centered at $\overline{p}$.
  $\triangleright$ Note: $P \subset B_{\lceil \log W \rceil}$, since $\|p - \overline{p}\| \leq WT \quad \forall p \in P$.
4 $G_j \leftarrow$ an infinite grid of cell size $2^{j-2}\varepsilon T/\sqrt{d}$ with $\overline{p}$ as a vertex.
5 **if** $j = 1$
6 **then** $V_1 \leftarrow G_1 \cap B_1$
7 **else** $V_j \leftarrow G_j \cap (B_j \setminus B_{j-1})$
8 **for** each cell $\Delta \in V_j$ intersecting P
9 **do** choose an arbitrary point p' in $P \cap \Delta$
  $\triangleright$ Note: $\forall p \in P \cap \Delta$, $\|p - p'\| \leq \varepsilon \|p - \overline{p}\|$ if $j > 1$,
  $\triangleright$ i.e., $\|p - p'\| \leq \varepsilon \cdot \max\{T, \|p - \overline{p}\|\}\, \forall j \geq 1$.
10 $w(p') \leftarrow \sum_{p \in P \cap \Delta} w(p)$
11 $S \leftarrow S \cup \{p'\}$
12 **return** S

Fig. 4.1: The algorithm SINGLE-FACILITY-CORESET

The algorithm SINGLE-FACILITY-CORESET given in Fig. 4.1 is very similar to the one in [HPM04], but, unlike [HPM04], it produces a single coreset that satisfies both (4.1)(i) and (ii). We use this algorithm later in this section, and in Section 4.4.

The proof that SINGLE-FACILITY-CORESET indeed returns an ε-coreset for P is a consequence of the following lemma; its somewhat cumbersome notation is needed for further applications, where we use it to prove the correctness of constructions of other coresets of interest, in Section 4.4.

Lemma 4.1. *Let P and S be two weighted sets in $\mathbb{R}^d$, and let g be a mapping from P to S such that the weight $w(p')$ of $p' \in S$ is equal to the sum of the weights of all points $p \in P$ with $g(p) = p'$.*

Let $\{P_1, P_2, \ldots, P_m\}$ be a partition of P, and let $\{C_1, C_2, \ldots, C_m\}$ be a col-

lection of m facilities (sets) in $\mathbb{R}^d$. Define $R = \sum_{i=1}^{m} \nu_{C_i}(P_i)/w(P)$, where $w(P) = \sum_{p\in P} w(p)$. Assume that for some $0 < \varepsilon \leq 1$ we have

$$\|p - g(p)\| \leq \varepsilon \cdot \max\{R, \operatorname{dist}(p, C_i)\},$$

for every $p \in P_i$, and every $1 \leq i \leq m$. Then, for any $Q \subset \mathbb{R}^d$,

(i) $\sum_{i=1}^{m} \nu_{C_i}(P_i) \leq \alpha\nu_Q(P)$, *for some $\alpha \geq 0$, implies that*

$$|\nu_Q(P) - \nu_Q(S)| \leq 2\alpha\varepsilon \cdot \nu_Q(P).$$

(ii) $\sum_{i=1}^{m} \mu_{C_i}(P_i) \leq \beta\mu_Q(P)$, *for some $\beta \geq 1$, implies that*

$$|\mu_Q(P) - \mu_Q(S)| \leq 9\beta\varepsilon \cdot \mu_Q(P).$$

Proof. **(i)** Let Q be any set in $\mathbb{R}^d$ such that $\sum_{i=1}^{m} \nu_{C_i}(P_i) \leq \alpha\nu_Q(P)$. By the triangle inequality, for any pair of points $p, p' \in P$ we have $\operatorname{dist}(p', Q) \leq \operatorname{dist}(p, Q) + \|p - p'\|$, and $\operatorname{dist}(p, Q) \leq \operatorname{dist}(p', Q) + \|p - p'\|$. Hence,

$$|\operatorname{dist}(p, Q) - \operatorname{dist}(p', Q)| \leq \|p - p'\| .$$

Thus, the error can be bounded by

$$\begin{aligned} |\nu_Q(P) - \nu_Q(S)| &= \left| \sum_{p\in P} w(p)\operatorname{dist}(p, Q) - \sum_{p'\in S} w(p')\operatorname{dist}(p', Q) \right| \\ &= \left| \sum_{p\in P} w(p)\big(\operatorname{dist}(p, Q) - \operatorname{dist}(g(p), Q)\big) \right| \\ &\leq \sum_{p\in P} w(p) \left|\operatorname{dist}(p, Q) - \operatorname{dist}(g(p), Q)\right| \\ &\leq \sum_{p\in P} w(p) \left\|p - g(p)\right\| . \end{aligned} \tag{4.2}$$

Let $P_R = \bigcup_{i=1}^{m} \{p \mid p \in P_i,\ \operatorname{dist}(p, C_i) \leq R\}$. By the assumption of the lemma, $\|p - g(p)\| \leq \varepsilon R$ for every $p \in P_R$, and $\|p - g(p)\| \leq \varepsilon \cdot \operatorname{dist}(p, C_i)$ for every $p \in P_i \setminus P_R$, and $i = 1, \cdots, m$. Hence,

$$\begin{aligned} \sum_{p\in P} w(p) \left\|p - g(p)\right\| &= \sum_{p\in P_R} w(p) \left\|p - g(p)\right\| + \sum_{i=1}^{m} \sum_{p\in P_i\setminus P_R} w(p) \left\|p - g(p)\right\| \\ &\leq w(P) \cdot \varepsilon R + \varepsilon \sum_{i=1}^{m} \nu_{C_i}(P_i). \end{aligned}$$

By definition of R, and the assumption $\sum_{i=1}^{m} \nu_{C_i}(P_i) \leq \alpha \nu_Q(P)$, this yields

$$\sum_{p \in P} w(p) \left\| p - g(p) \right\| \leq 2\varepsilon \sum_{i=1}^{m} \nu_{C_i}(P_i) \leq 2\varepsilon\alpha \cdot \nu_Q(P),$$

which, together with (4.2), concludes the proof of part (*i*) of the lemma.
(*ii*) Similarly, the overall error in this case is bounded by

$$\begin{aligned}
\left|\mu_Q(P) - \mu_Q(S)\right| &= \left| \sum_{p \in P} w(p) \big(\mathrm{dist}(p, Q)\big)^2 - \sum_{p' \in S} w(p') \big(\mathrm{dist}(p', Q)\big)^2 \right| \\
&= \left| \sum_{p \in P} w(p) \left(\big(\mathrm{dist}(p, Q)\big)^2 - \big(\mathrm{dist}(g(p), Q)\big)^2 \right) \right| \\
&\leq \sum_{p \in P} w(p) \left|\mathrm{dist}(p, Q) - \mathrm{dist}(g(p), Q)\right| \cdot \big(\mathrm{dist}(p, Q) + \mathrm{dist}(g(p), Q)\big).
\end{aligned}$$

As in the proof of (*i*), for any $p \in P$ we have $|\mathrm{dist}(p, Q) - \mathrm{dist}(g(p), Q)| \leq \|p - g(p)\|$, and also $\mathrm{dist}(p, Q) + \mathrm{dist}(g(p), Q) \leq 2\mathrm{dist}(p, Q) + \|p - g(p)\|$. We thus have

$$\begin{aligned}
&\left|\mu_Q(P) - \mu_Q(S)\right| \\
&\leq \sum_{p \in P} w(p) \left\| p - g(p) \right\| \big(2\mathrm{dist}(p, Q) + \left\| p - g(p) \right\| \big) \\
&= \sum_{p \in P} 2w(p) \left\| p - g(p) \right\| \cdot \mathrm{dist}(p, Q) + \sum_{p \in P} w(p) \left\| p - g(p) \right\|^2 .
\end{aligned} \tag{4.3}$$

For each $1 \leq i \leq m$ and each $p \in P_i$, put $x(p) = \max\{R, \mathrm{dist}(p, C_i), \mathrm{dist}(p, Q)\}$. Thus, $\mathrm{dist}(p, Q) \leq x(p)$. By the assumption of the lemma we have $\|p - g(p)\| \leq \varepsilon x(p)$. Substituting this in (4.3) yields (for $0 < \varepsilon < 1$),

$$\begin{aligned}
|\mu_Q(P) - \mu_Q(S)| &\leq 2\varepsilon \sum_{p \in P} w(p) x^2(p) + \varepsilon^2 \sum_{p \in P} w(p) x^2(p) \\
&\leq 3\varepsilon \sum_{p \in P} w(p) x^2(p).
\end{aligned} \tag{4.4}$$

Finally, we have

$$\begin{aligned}
\sum_{p \in P} w(p) x^2(p) &\leq R^2 \sum_{p \in P} w(p) + \sum_{i=1}^{m} \mu_{C_i}(P_i) + \sum_{p \in P} w(p) \cdot \big(\mathrm{dist}(p, Q)\big)^2 \\
&\leq w(P) R^2 + \beta \mu_Q(P) + \mu_Q(P).
\end{aligned} \tag{4.5}$$

Using the Cauchy-Schwarz inequality, we have

$$\begin{aligned} R^2 &= \frac{1}{w(P)^2}\left(\sum_{i=1}^{m}\nu_{C_i}(P_i)\right)^2 = \frac{1}{w(P)^2}\left(\sum_{i=1}^{m}\sum_{p\in P_i} w(p)\mathrm{dist}(p,C_i)\right)^2 \\ &\leq \frac{1}{w(P)^2}\left(\sum_{p\in P} w(p)\right)\cdot\left(\sum_{i=1}^{m}\sum_{p\in P_i} w(p)\big(\mathrm{dist}(p,C_i)\big)^2\right) \\ &= \frac{1}{w(P)}\cdot\sum_{i=1}^{m}\mu_{C_i}(P_i). \end{aligned}$$

Hence, by the assumption of the lemma,

$$R^2 \leq \frac{\beta\mu_Q(P)}{w(P)}$$

Substituting this in (4.5) gives us

$$\sum_{p\in P} w(p)x^2 \leq \beta\mu_Q(P) + \beta\mu_Q(P) + \mu_Q(P) \leq 3\beta\mu_Q(P),$$

which, together with (4.4), concludes the proof of part (*ii*) of the lemma. □

Corollary 4.2. *Let P be a set of n (unweighted) points in $\mathbb{R}^d$, and $0 < \varepsilon \leq 1$. Let $s = 2^{O(d)}\cdot(\sqrt{d}/\varepsilon)^d \log n$. Then Then, with an appropriate choice of the constant of proportionality,* SINGLE-FACILITY-CORESET$(P, \varepsilon/9)$ *returns, in $O(dn + s)$ time, a single-facility ε-coreset for P of size s.*

Proof. The size of S is bounded by the number of grid cells that contain the points of P. For each j, the number of cells of G_j (empty or not) inside B_j is

$$O\left(\frac{(2^jT)^d}{(2^j\varepsilon T\sqrt{d})^d}\right) = 2^{O(d)}\cdot(\sqrt{d}/\varepsilon)^d.$$

Summing over j, and observing that $W = n$ in this case, the bound on $|S|$ follows.

With careful implementation, SINGLE-FACILITY-CORESET takes $O(dn + s)$ time, using the log and floor functions. (One simply has to compute, for each $p \in P$, the grid cell containing p. The number of points in each of the s cells can then be computed using count sort with an array of size s.) We next argue that the output set S of a call to SINGLE-FACILITY-CORESET(P, ε) is a 9ε-coreset. This would prove the corollary by replacing ε with $\varepsilon/9$.

Note that P is an unweighted set of points, so $w(p) = 1$ for each $p \in P$ and $w(P) = n$. Let $\overline{p} = \frac{1}{n}\sum_{p\in P} p$, and $T = \nu_{\{\overline{p}\}}(P)/n$, be as defined in the procedure. As noted in Line 9 of SINGLE-FACILITY-CORESET, we have $\|p - g(p)\| \leq \varepsilon \max\{T, \|p - \overline{p}\|\}$ for every $p \in P$ and its representative $g(p)$ in the coreset. It is also well known (see [DHS00]) that for any $q \in \mathbb{R}^d$ we have $\nu_{\{\overline{p}\}}(P) \leq 2\nu_{\{q\}}(P)$, and $\mu_{\{\overline{p}\}}(P) \leq \mu_{\{q\}}(P)$. Thus, substituting $m = 1$, $C_1 = \{\overline{p}\}$, $Q = \{q\}$, $\alpha = 2$, $\beta = 1$ in Lemma 4.1, shows that S is indeed a 9ε-coreset, and thus completes the proof of the corollary. □

The following corollary is used in the next section. It states that in the special case that P is contained in a line (in $\mathbb{R}^d$), the size of its coreset is independent of d.

Corollary 4.3. *Let P be a set of n (unweighted) points on a line ℓ in $\mathbb{R}^d$, and $0 < \varepsilon \leq 1$. Then a single-facility ε-coreset for P of size $s = O\left(\frac{1}{\varepsilon}\log n\right)$ can be computed in $O(nd)$ time.*

Proof. If $nd < \frac{1}{\varepsilon}\log n$, the algorithm simply returns the input set P; see Line 0. Otherwise, we execute SINGLE-FACILITY-CORESET$(P, \varepsilon/9)$ with the following modifications: (i) We rotate the coordinate frame so that ℓ becomes one of the axes. (ii) We choose the size of the grid G_j (in Line 4) to be $2^{j-2}\varepsilon T$. Then the number of cells Δ in Line 8 is only $O(1/\varepsilon)$, and the claim follows. □

4.2 (k, ε)-Coresets for Weighted Facilities

In this section we assume that P is a set of n unweighted points on a line ℓ in $\mathbb{R}^d$. We first introduce several notations.

Voronoi region. Given a weighted set of point facilities $C \subset \mathbb{R}^d$, with an associated weight function $W\colon C \mapsto \mathbb{R}^+$, we define the *Voronoi region* $V(c)$ associated with $c \in C$ to be the set of points $x \in \mathbb{R}^d$ such that $W(c)\|x - c\| \leq W(c')\|x - c'\|$ for all $c' \in C$. See Fig. 2.1. The collection of these Voronoi regions constitutes the multiplicatively-weighted Voronoi diagram of C; see [Don08].

Voronoi intervals and boundaries. Given a line $\ell \subseteq \mathbb{R}^d$, a set of facilities $C \subset \mathbb{R}^d$, and an associated weight function $W\colon C \mapsto \mathbb{R}^+$, a *Voronoi interval* for a facility $c \in C$ is a connected component of $V(c) \cap \ell$. Endpoints of Voronoi intervals are called *Voronoi boundaries*. Two Voronoi intervals are called *adjacent* if they share a Voronoi boundary.

Remark 4.4. Note that if all the facilities have the same weight, then each facility has a single connected Voronoi interval; see Fig. 2.1(left). However, if their weights are unequal, then a single facility may "serve" multiple intervals (in the above sense); see Fig. 2.1(right).

Lemma 4.5. *Let $C \subset \mathbb{R}^d$ be a weighted set of k facilities. The total number of their Voronoi intervals on a fixed line ℓ is at most $2k-1$.*

Proof. Let $C = \{c_1, c_2, \ldots, c_k\}$. For every point t on ℓ, consider the k weighted distances $W(c_i)\,\|t - c_i\|$ for $1 \leq i \leq k$ and observe that t lies in a Voronoi interval of c_i if and only if $W(c_i)\,\|t - c_i\|$ attains the lower envelope of these functions at t. It is easily checked that any pair of these functions intersect at most twice. Hence, if we label each Voronoi interval on ℓ by the facility c that serves it, we obtain a Davenport-Schinzel sequence of order 2 on k symbols [SA95], so the number of resulting intervals is at most $\lambda_2(k) = 2k-1$. □

In what follows we assume, without loss of generality, that ℓ is the x_1-axis; for further simplicity, we simply refer to it as the x-axis.

(k, ε)-V-coreset for P. A weighted set $S \subseteq P$ is called a (k, ε)-*V-coreset*, if, for any weighted set $C \subset \mathbb{R}^d$ of facilities, such that P is contained in at most k adjacent Voronoi intervals of C, (*i*) and (*ii*) hold:

$$\begin{aligned} (i) \quad & (1-\varepsilon)\nu'_C(P) \leq \nu'_C(S) \leq (1+\varepsilon)\nu'_C(P), \\ (ii) \quad & (1-\varepsilon)\mu'_C(P) \leq \mu'_C(S) \leq (1+\varepsilon)\mu'_C(P), \end{aligned}$$

where

$$\begin{aligned} \nu'_C(P) &= \sum_{p \in P} \left(w(p) \cdot \min_{c \in C} \{W(c)\,\|p-c\|\} \right), \text{ and} \\ \mu'_C(P) &= \sum_{p \in P} \left(w(p) \cdot \min_{c \in C} \left\{ (W(c)\,\|p-c\|)^2 \right\} \right). \end{aligned}$$

Note that k here differs from the number of facilities (but at most by a factor of 2); recall also that here P is unweighted (i.e., the weights of its points are all 1).

4.3 The Construction of V-Coresets

Let P be a set of n points on the x-axis, $k \geq 1$ an integer, and $0 < \varepsilon \leq 1$. As we will shortly show, the algorithm V-CORESET, given in Fig. 4.3, returns a weighted subset $S \subseteq P$ of size

$$|S| = \left(\frac{\log n}{\varepsilon}\right)^{O(k)},$$

which is a (k, ε)-V-coreset for P.

In the main part of the algorithm, we assume that $|P| > \lceil \delta/\varepsilon \rceil$ (for the constant δ specified in the algorithm). Otherwise, we take P itself as the coreset (Line 2 of V-CORESET). The algorithm is recursive and makes use of $(k-1, \varepsilon)$-V-coresets for various subsets of P, where the base case for the recursion is the case $k = 1$, discussed in Section 4.1, and solved using the SINGLE-FACILITY-CORESET routine. In this case (Line 4) the weight of the single facility is irrelevant for the property that we seek. Thus, an ε-coreset for P, as constructed in Line 4, is also a $(1, \varepsilon)$-V-coreset for P (a single facility always introduces a single Voronoi interval, namely, the entire line).

Otherwise $(k > 1)$, the loop in Lines 7-10 splits the left half of P into subsets $B_1, B_2, \ldots,$ by intersecting P with a sequence of intervals, drawn from right to left, whose lengths increase by a factor of 2. Then, the loop in lines 12-23 splits each set B_i into subsets $B_{i1}, B_{i2}, \ldots$. It scans B_1 from left to right and the sets B_j, $j > 1$, from right to left. During the scan, it creates $\lfloor \delta/\varepsilon \rfloor$ subsets of size 1, then $\lfloor \delta/\varepsilon \rfloor$ subsets of size 2, and keeps doubling the sizes until all of B_i is exhausted.

The collection of these sets $\{B_{ij}\}$, over all i and j, is denoted by $\mathcal{Z}$. In Lines 24-26, we construct recursively a coreset for $k-1$ weighted facilities, for each $B \in \mathcal{Z}$. We let S_ℓ be the union of the resulting coresets. So far the construction is applied only for the left side of P. In Line 27 we repeat a mirror-image construction for the right half of P, resulting in a set S_r. The output coreset is the union of these two sets S_ℓ and S_r.

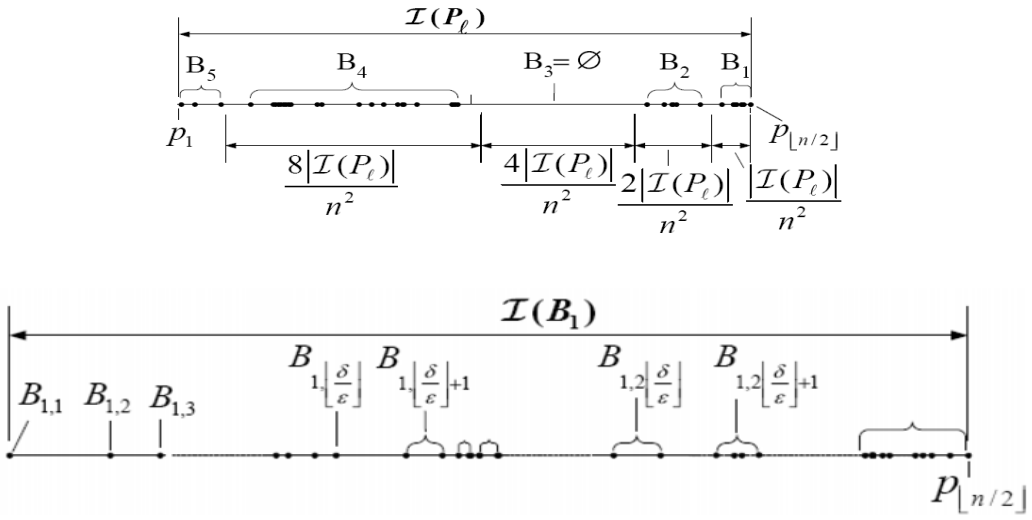


Fig. 4.2: **(top)** The high-level partition of the set P_ℓ of the $\lfloor n/2 \rfloor$ leftmost points of P into intervals and sets. **(bottom)** The partition of B_1 into subsets. The other subsets B_i, for $i > 1$, are similarly partitioned, but from right to left rather than from left to right.

Algorithm V-CORESET (P, k, ε)
Input: P : set of n points on a line, $k \geq 1$ an integer, and $0 < \varepsilon \leq 1$.
Output: $(k, 3\varepsilon)$-V-coreset of P for weighted facilities.

1 **if** $|P| \leq \lceil \delta/\varepsilon \rceil$ $\quad \triangleright$ δ is a constant, defined later in Lemma 4.10
2 **then return** P
3 **if** $k = 1$
4 **then return** SINGLE-FACILITY-CORESET$(P, \varepsilon/3)$
5 $p_1 \leftarrow$ leftmost point of P; $\quad p_{\lfloor n/2 \rfloor} \leftarrow \lfloor n/2 \rfloor$-leftmost point of P
6 $end \leftarrow p_{\lfloor n/2 \rfloor}$
7 **for** $i \leftarrow 1$ to $2\lceil \log n \rceil + 1$
8 **do** $begin \leftarrow end - 2^{i-1} \left| p_{\lfloor n/2 \rfloor} - p_1 \right| / n^2$
9 $B_i \leftarrow P \cap (begin, end]$
10 $end \leftarrow begin$
11 $\mathcal{Z} \leftarrow \emptyset$ $\quad \triangleright$ $\mathcal{Z}$ is a collection of sets
12 **for** $i \leftarrow 1$ to $2\lceil \log n \rceil + 1$
13 **do** $B_{i1} \leftarrow \emptyset$; $\quad size \leftarrow 1$; $\quad j \leftarrow 1$;
14 **for** $m \leftarrow 1$ to $|B_i|$
15 **do if** $i = 1$
16 **then** add to B_{ij} the mth leftmost point of B_i
17 **else** add to B_{ij} the mth rightmost point of B_i
18 **if** $|B_{ij}| = size$
19 **then** $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{B_{ij}\}$
20 $j \leftarrow j + 1$
21 $B_{ij} \leftarrow \emptyset$
22 **if** $(j \mod \lceil \delta/\varepsilon \rceil) = 0$
23 **then** $size \leftarrow 2 \cdot size$
24 $S_\ell \leftarrow \emptyset$
25 **for** each $B \in \mathcal{Z}$
26 $S_\ell \leftarrow S_\ell \cup$ V-CORESET$(B, k-1, \varepsilon/3)$
27 Repeat Lines 5–26 for the $\lceil n/2 \rceil$ rightmost points of P, resulting in a set S_r
$\triangleright$ (Use a mirror-image construction)
28 **return** $S_\ell \cup S_r$

Fig. 4.3: The algorithm V-CORESET

Lemma 4.6. $|\mathcal{Z}| = O(\varepsilon^{-1} \log^2 n)$.

Proof. No $B_{ij} \in \mathcal{Z}$ can have more than $\lceil 2\varepsilon n/\delta \rceil$ points. Indeed, for $j \leq \lfloor \delta/\varepsilon \rfloor$, $|B_{ij}| = 1$ by construction. Suppose to the contrary that for $j > \lfloor \delta/\varepsilon \rfloor$, there exists a set B_{ij} with more than $\lceil 2\varepsilon n/\delta \rceil$ points. Then, by construction, each of the $\lfloor \delta/\varepsilon \rfloor$ sets $B_{ij'}$ that precedes B_{ij} satisfies $|B_{i,j'}| \geq 1/2\,|B_{ij}| \geq \lceil \varepsilon n/\delta \rceil$. This would imply, in turn, that $|P| > n$, a contradiction. The size of the largest subset of each B_i is thus at most $\lceil 2\varepsilon n/\delta \rceil$. Since the partition of B_i consists of $\lceil \delta/\varepsilon \rceil$ subsets of size 2^t, for $t = 0, 1, \ldots$, and since the maximum size is $\lceil 2\varepsilon n/\delta \rceil$ it follows that the number of subsets of B_i is $O(\varepsilon^{-1} \log(\varepsilon n)) = O(\varepsilon^{-1} \log n)$. Since there are $O(\log n)$ sets B_i, it follows that $|\mathcal{Z}| = O(\varepsilon^{-1} \log^2 n)$. □

Lemma 4.7. *The number of points in the set S is at most*

$$2^{O(k)} \varepsilon^{-k} \log^{2k-1} n.$$

Proof. We only consider S_ℓ; the proof is similar for S_r. Define $T(k, \varepsilon, n)$ to be the maximum size of S_ℓ for given k and ε, and for any input set P of n points. We establish the upper bound $T(k, \varepsilon, n) \leq b^k \varepsilon^{-k} \log^{2k-1} n$, for an appropriate absolute constant b, using induction on k. From the construction for $k = 1$, it follows that $T(1, \varepsilon, n) = O\left(\frac{\log n}{\varepsilon}\right)$, which satisfies the bound asserted in the lemma for an appropriate choice of b. We also make b sufficiently large so that the bound in Lemma 4.6 is at most $b\varepsilon^{-1} \log^2 n$. Then, by construction and the induction hypothesis, we have

$$T(k, \varepsilon, n) \leq |\mathcal{Z}| \cdot T(k-1, \varepsilon, n) \leq b^k \varepsilon^{-k} \log^{2k-1} n,$$

as claimed. □

To prove that S is a (k, ε)-V-coreset, we will frequently use the following simple observation. We denote by $\mathcal{I}(X)$ the smallest interval containing a set X. We denote by P_ℓ the set of $\lfloor n/2 \rfloor$ leftmost points of P.

Observation 4.8. **(i)** *The length of the interval $\mathcal{I}(B_i)$, for $i > 1$, is less than twice the length of the interval spanned by the rightmost point of B_i and the rightmost points of P_ℓ (i.e., $p_{\lfloor n/2 \rfloor}$; cf. Fig. 4.2(top)).*
(ii) *For any $B_{ij} \in \mathcal{Z}$ that contains at least two points, we have (cf. Fig. 4.2(bottom))* $|B_{ij}| \leq (2\varepsilon/\delta) \sum_{m<j} |B_{im}|$.

Theorem 4.9. *Let P be a set of n points on a line in $\mathbb{R}^d$, $k \geq 1$ an integer, and $0 < \varepsilon \leq 1$. The algorithm* V-CORESET(P, k, ε) *returns, in $O(ndk)$ time, a $(k, 3\varepsilon)$-V-coreset S for P of size $2^{O(k)} \varepsilon^{-k} \log^{2k-1} n$.*

Proof. The bound on the size of S is given in Lemma 4.7. Each execution of V-CORESET, excluding the recursive calls, can be implemented in $O(nd)$ time. Indeed, Lines 5–10 of V-CORESET can be easily implemented, in $O(nd)$ time (without sorting P), using the log and floor functions. Lines 11–23 can be implemented in $O(nd)$ time (again, avoiding sorting) by computing the sets B_{ij} backwards, finding the points in the $\lfloor \delta/\varepsilon \rfloor$ last (largest) subsets $B_{ij} \subseteq B_i$ in $O(|B_i|)$ time, using a linear-time algorithm for order statistics, and then by continuing recursively on the remaining points. The cost of each phase is linear in the size of the subset of remaining points, and since these sizes form a geometric progression, the claim follows. Since the elements of $\mathcal{Z}$ are pairwise disjoint, a simple induction on k, concerning the recursive call at Line 26, shows that the overall running time of V-CORESET, excluding the executions of Line 4 at the bottom of recursion (when $k = 1$), is $O(ndk)$. Each of these executions has as input a some subset B of P, and takes $O(|B| \cdot d)$ time according to Corollary 4.3. Since these subsets are disjoint, the overall running time for all the executions when $k = 1$ is therefore $O(nd)$. This establishes the asserted bound on the running time.

We next show that S is a $(k, 3\varepsilon)$-V-coreset. For the case $k = 1$ the weight of the single facility is irrelevant for the property that we seek. Thus, the 3ε-coreset that is returned in Line 4 is also a $(1, 3\varepsilon)$-V-coreset for P. It remains to prove the case $k > 1$. Interestingly enough, the following proof of correctness for non-squared distances remains true for squared distances, if we use everywhere the cost function μ' instead of ν', and replace $\|\cdot\|$ by $\|\cdot\|^2$.

Let $C \subset \mathbb{R}^d$ be any weighted set of facilities, such that P falls into no more than k adjacent Voronoi intervals of C. By Line 28, $S = S_\ell \cup S_r$, where S_ℓ is the coreset for P_ℓ and S_r is the coreset for $P_r = P \setminus P_\ell$. We have

$$\begin{aligned} &|\nu'_C(P) - \nu'_C(S)| = \\ &\left|\left(\nu'_C(P_\ell) + \nu'_C(P_r)\right) - \left(\nu'_C(S_\ell) + \nu'_C(S_r)\right)\right| \\ &\leq |\nu'_C(P_\ell) - \nu'_C(S_\ell)| + |\nu'_C(P_r) - \nu'_C(S_r)| \,. \end{aligned} \tag{4.6}$$

We will prove that $|\nu'_C(P_\ell) - \nu'_C(S_\ell)| \leq (3/2)\varepsilon\nu'_C(P)$. A symmetric proof will then imply $|\nu'_C(P_r) - \nu'_C(S_r)| \leq (3/2)\varepsilon\nu'_C(P)$. The coreset property of S then follows from (4.6).

If it so happens that for every $B \in \mathcal{Z}$, the interval $\mathcal{I}(B)$ intersects no more than $k - 1$ Voronoi intervals, we are done, because then, by the recursive construction,

$$|\nu'_C(B) - \nu'_C(S_B)| \leq \varepsilon\nu'_C(B)$$

for each $B \in \mathcal{Z}$, where S_B is the coreset computed for B in Line 26. The coreset

S_ℓ is the union of the coresets S_B, for all $B \in \mathcal{Z}$, and thus

$$|\nu'_C(P_\ell) - \nu'_C(S_\ell)| = \left| \sum_{B \in \mathcal{Z}} \left(\nu'_C(B) - \nu'_C(S_B) \right) \right|$$

$$\leq \sum_{B \in \mathcal{Z}} |\nu'_C(B) - \nu'_C(S_B)| \leq \sum_{B \in \mathcal{Z}} \varepsilon \nu'_C(B)$$

$$= \varepsilon \nu'_C(P_\ell) \leq \varepsilon \nu'_C(P) < (3/2)\varepsilon \nu'_C(P).$$

We are left to handle the case where there is some set $B \in \mathcal{Z}$ such that $\mathcal{I}(B)$ intersects all k Voronoi intervals (and thus contains $k-1$ Voronoi boundaries — see Fig. 4.4 (top)). In this case the sum of errors contributed by the rest of the $(k-1, \varepsilon)$-V-coresets is then (here each of the corresponding intervals $\mathcal{I}(x)$ intersects a single Voronoi interval, so the induction hypothesis applies).

$$\sum_{X \in \mathcal{Z} \setminus \{B\}} |\nu'_C(X) - \nu'_C(S_X)| \leq \sum_{X \in \mathcal{Z} \setminus \{B\}} \varepsilon \nu'_C(X)$$

$$\leq \sum_{X \in \mathcal{Z}} \varepsilon \nu'_C(X) = \varepsilon \nu'_C(P_\ell) \leq \varepsilon \nu'_C(P).$$

We will show that in this case

$$|\nu'_C(B) - \nu'_C(S_B)| \leq \frac{\varepsilon}{2} \nu'_C(P), \tag{4.7}$$

and thus

$$|\nu'_C(P_\ell) - \nu'_C(S_\ell)| \leq \varepsilon \nu'_C(P) + \frac{\varepsilon}{2} \nu'_C(P) = \frac{3\varepsilon}{2} \nu'_C(P).$$

To prove (4.7), let c be any facility in C. For simplicity, we abuse notation, and write $\nu'_c(P)$ instead of $\nu'_{\{c\}}(P)$. By construction, S_B is a $(k-1, \varepsilon)$-V-coreset, so, by definition, S_B is also a $(1, \varepsilon)$-V-coreset. Hence, $|\nu'_c(S_B) - \nu'_c(B)| \leq \varepsilon \nu'_c(B) \leq \nu'_c(B)$, so, $\nu'_c(S_B) \leq 2\nu'_c(B)$. Thus, for any facility $c \in C$, the left-hand side of (4.7) can be bounded by

$$|\nu'_C(B) - \nu'_C(S_B)| \leq \nu'_C(B) + \nu'_C(S_B) \tag{4.8}$$

$$\leq \nu'_c(B) + 2\nu'_c(B) = 3\nu'_c(B).$$

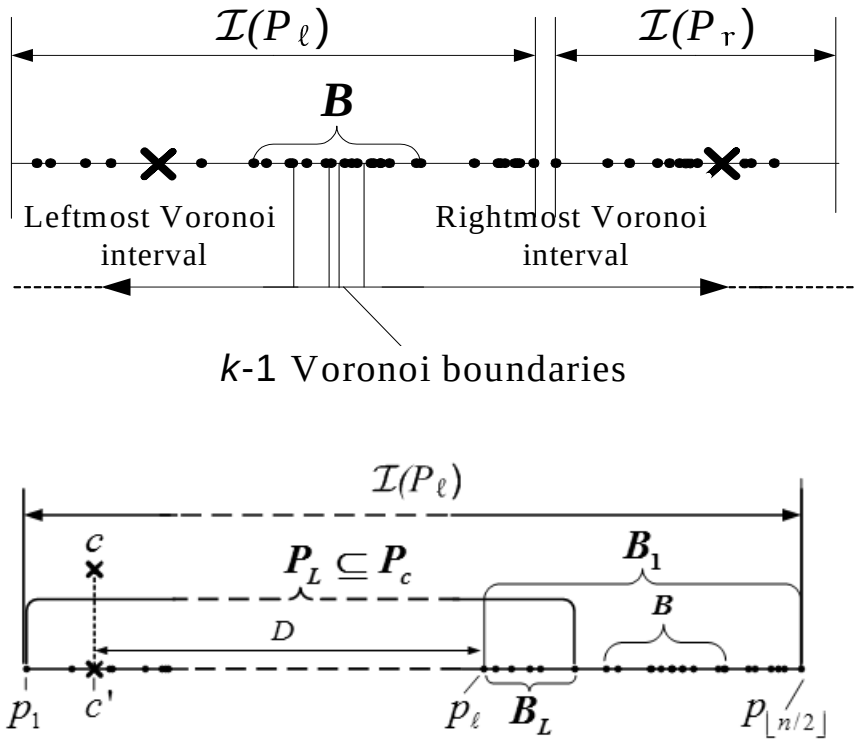


Fig. 4.4: **(top)** All the k Voronoi intervals intersect $\mathcal{I}(B)$ for some $B \in \mathcal{Z}$. The two 'x' facilities in this figure serve the leftmost and rightmost Voronoi intervals (in general, they do not have to lie on the line). **(bottom)** B intersects k Voronoi intervals, and is also contained in B_1. The facility $c \in C$ serves the leftmost Voronoi interval, and c' denotes its projection on the line. Since c' can be anywhere on the line, its nearest point in B_1 can be any point of B_1.

Let the facility c' be the projection of c on the x-axis, with weight $W(c') = W(c)$. See Fig. 4.4 (bottom). Using the triangle inequality, $\nu'_c(B)$ can be bounded by

$$\begin{aligned}\nu'_c(B) &= W(c)\sum_{p\in B}\|p-c\| \qquad (4.9)\\ &\le W(c)\sum_{p\in B}(\|c-c'\|+\|p-c'\|)\\ &= W(c)\,|B|\cdot\|c-c'\|+\nu'_{c'}(B).\end{aligned}$$

We now argue that, with an appropriate choice of the facility c and the constant parameter δ, each of the two terms in the right-hand side of (4.9) is at most $(\varepsilon/12)\nu'_C(P)$, which, using (4.8), will prove (4.7) and conclude the proof of the theorem. Let B_i be the set that contains $B = B_{ij}$. We distinguish between the following two cases.

(i) $B_i = B_1$: Let $c \in C$ be the facility that serves the leftmost Voronoi interval, and denote by P_c the points of P that are served by c. Also, let B_L denotes the set of points of B_1 that lie to the left of B, and note that $B_L \subseteq P_c$ (because the Voronoi interval of c and ends "inside" B_j (see Fig. 4.4(bottom)). By Observation 4.8(*ii*) we have, $|B| \le (\varepsilon/12)\,|B_L|$, if we choose $\delta \ge 24$, and thus $|B| \le (\varepsilon/12)\,|P_c|$. Clearly, c' is the nearest point on the x-axis to c, and therefore $\|c-c'\| \le \|p-c\|$ for any $p \in P$. Hence, $|P_c|\cdot\|c-c'\| \le \nu_c(P_c)$. Altogether we have

$$\begin{aligned}W(c)\,|B|\cdot\|c-c'\| &\le W(c)\cdot\frac{\varepsilon}{12}\,|P_c|\cdot\|c-c'\|\\ &\le \frac{\varepsilon}{12}\nu'_c(P_c) \le \frac{\varepsilon}{12}\nu'_C(P).\end{aligned}$$

To bound the second term of (4.9), let P_L denote the points of P to the left of B, and note that $P_L \subseteq P_c$; see Fig. 4.4(bottom). Clearly, $\|p-c'\| \le \|p-c\|$ for every p on the x-axis, and we get $\nu_{c'}(P_L) \le \nu_c(P_L) \le \nu_c(P_c)$. Using Lemma 4.10(*i*) that follows, we have $\nu_{c'}(B) \le (\varepsilon/12)\nu_{c'}(P_L)$ (note that $|B| > 1$, since a single point cannot intersect $k > 1$ Voronoi intervals). After multiplying by $W(c)$, this yields $\nu'_{c'}(B) \le (\varepsilon/12)\nu'_c(P_c) \le (\varepsilon/12)\nu_C(P)$.

(ii) $B \subseteq B_i \ne B_1$: The proof is symmetric, taking c to be the facility that serves the *rightmost* Voronoi interval. The sets B_R, P_R, defined symmetrically to the definitions of B_L, P_L, and Lemma 4.10(*ii*), should then replace B_L, P_L and Lemma 4.10(*i*), respectively. □

To conclude the proof of Theorem 4.9, we still need to show that $\nu_{c'}(B) \le (\varepsilon/12)\nu_{c'}(P_L)$ (for $B \subseteq B_1$) or $\nu_{c'}(B) \le (\varepsilon/12)\nu_{c'}(P_R)$ (for $B \subseteq B_i \ne B_1$),

for a facility c' on the x-axis, and that the analogous inequality hold for squared distances too. All this is established in the following lemma. An additional feature of the lemma is that it shows how to choose the constant δ used by the algorithm and the preceding analysis.

Lemma 4.10. *Let $P \subset \mathbb{R}$ be a finite set of points. Let $\mathcal{Z} = \{B_{ij}\}$ be the partition of P given in Lines 11–23 of the algorithm* V-CORESET*, for the specified $0 < \varepsilon \leq 1$ and k. Consider a set $B = B_{ij} \in \mathcal{Z}$, where $|B| > 1$ (i.e., $j > \lfloor \delta/\varepsilon \rfloor$, see Fig. 4.2(bottom)), and let P_L, P_R denotes the set of points of P that lie to the left and to the right of B, respectively $\big($see Fig. 4.4(bottom)$\big)$. Then, by choosing $\delta \geq 1152$, for any facility $c' \in \mathbb{R}$ we have*
(i) for $i = 1$, $\nu_{c'}(B) \leq (\varepsilon/12)\nu_{c'}(P_L)$, and $\mu_{c'}(B) \leq (\varepsilon/12)\mu_{c'}(P_L)$;
(ii) for $i > 1$, $\nu_{c'}(B) \leq (\varepsilon/12)\nu_{c'}(P_R)$, and $\mu_{c'}(B) \leq (\varepsilon/12)\mu_{c'}(P_R)$.

Proof. **(*i*)** Let $D = \operatorname{dist}(c', B_1)$ denote the distance between c' and the nearest point in B_1 (see Fig. 4.4(bottom)). Since $B \subseteq B_1$, we have $\nu_{c'}(B) \leq |B| \big(D + |\mathcal{I}(B_1)|\big)$. As above, let B_L be the set of points of B_1 that lie to the left of B. By definition, $|c' - p| \geq \operatorname{dist}(c', B_1) = D$ for each $p \in B_L$. By Observation 4.8(*ii*) we have $|B| \leq (\varepsilon/72)\,|B_L|$, by choosing a sufficiently large constant δ (at least 144) in the construction. Hence, by the way B_1 is constructed, in Lines 8-9 of the algorithm,

$$\nu_{c'}(B) \leq \frac{\varepsilon}{72}\,|B_L| \cdot \big(D + |\mathcal{I}(B_1)|\big) \leq \frac{\varepsilon}{72}\,|B_L| \cdot D + \frac{\varepsilon}{72}\frac{|\mathcal{I}(P_\ell)|}{n}. \tag{4.10}$$

To conclude the proof of (*i*), we now bound each term in the right-hand side of (4.10) by $(\varepsilon/72)\nu_{c'}(P_L)$, as follows. By definition, B_L is contained in B_1, so $|B_L| \cdot D \leq \nu_{c'}(B_L)$. Since $B_L \subseteq P_L$, this yields $|B_L| \cdot D \leq \nu_{c'}(P_L)$, hence the first term of (4.10) is at most $(\varepsilon/72)\nu_{c'}(P_L)$. For the second term, we use the fact that $n \geq 2$ (otherwise we take P itself as the coreset), and thus

$$\frac{|\mathcal{I}(P_\ell)|}{n} \leq |\mathcal{I}(P_\ell)| - \frac{|\mathcal{I}(P_\ell)|}{n^2} = |\mathcal{I}(P_\ell)| - |\mathcal{I}(B_1)|\,. \tag{4.11}$$

The points p_1 and p_ℓ (the leftmost point of B_1) are both in P_L; see Fig. 4.4(bottom). This means that $|\mathcal{I}(P_\ell)| - |\mathcal{I}(B_1)| = p_\ell - p_1 \leq |p_\ell - c'| + |p_1 - c'| \leq \nu_{c'}(P_L)$. Substituting this in (4.11) yields the desired bound on the second term of (4.10).

For squared distances, (4.10) should be replaced by

$$\begin{aligned} \mu_{c'}(B) &\leq \frac{\varepsilon}{72}\,|B_L| \cdot \big(D + |\mathcal{I}(B_1)|\big)^2 \\ &\leq \frac{\varepsilon}{36}\,|B_L| \cdot D^2 + \frac{\varepsilon}{36}\left(\frac{|\mathcal{I}(P_\ell)|}{n}\right)^2, \end{aligned} \tag{4.12}$$

for the same choice of δ. Using similar arguments to the non-squared case, we have $|B_L| \cdot D^2 \leq \mu_{c'}(P_L)$ and

$$\begin{aligned}\big(\,|\mathcal{I}(P_\ell)|\,/n\big)^2 &\leq (p_\ell - p_1)^2 \leq (|p_\ell - c'| + |p_1 - c'|)^2 \\ &\leq 2|p_\ell - c'|^2 + 2|p_1 - c'|^2 \leq 2\mu_{c'}(P_L),\end{aligned}$$

which yields the desired bound also for the squared distances case.
(ii) Let $D = \mathrm{dist}(c', B_i)$ and let B_R denote the set of points in B_i to the right of B; see Fig. 4.5. As in (*i*), we have $\nu_{c'}(B) \leq |B| \cdot D + |B| \cdot |\mathcal{I}(B_i)|$, and the first term is bounded by $(\varepsilon/72)\nu_{c'}(P_R)$ in the same way, using Observation 4.8(*ii*) and choosing $\delta \geq 144$. We next bound the second term by $(\varepsilon/18)\nu_{c'}(P_R)$. To do so, let p_r and $p_{\lfloor n/2 \rfloor}$ be the rightmost points of B_i and of P_ℓ, respectively, and define $p_{\text{mid}} = (p_{\lfloor n/2 \rfloor} + p_r)/2$; see Fig. 4.5. By Observations 4.8(*i*) and (*ii*)

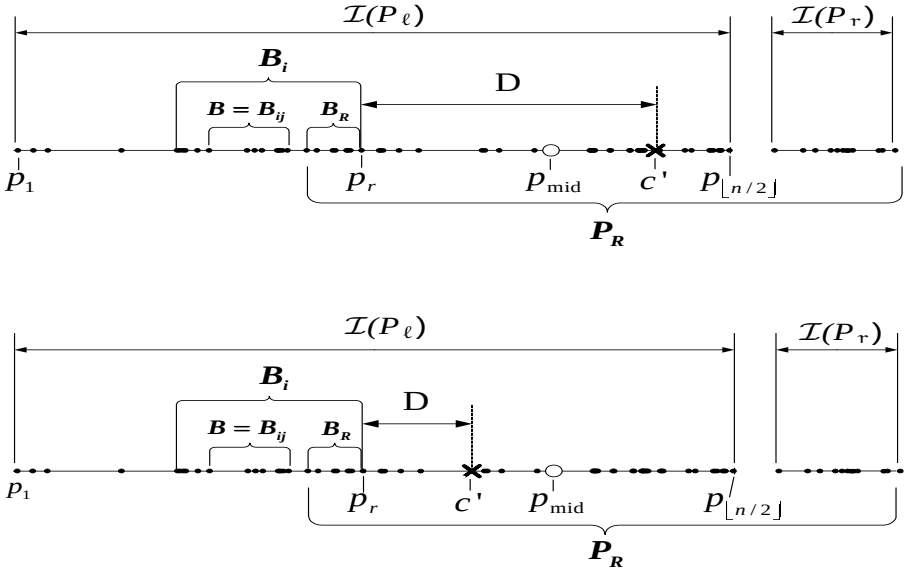


Fig. 4.5: The two cases of Lemma 4.10(*ii*), where **(top)** $p_{\text{mid}} \leq c'$, and **(bottom)** $p_{\text{mid}} > c'$.

$$|B| \cdot |\mathcal{I}(B_i)| \leq \frac{2\varepsilon}{\delta}\,|B_R| \cdot 2(p_{\lfloor n/2 \rfloor} - p_r). \tag{4.13}$$

In case $p_{\text{mid}} \leq c'$ (see Fig. 4.5(top)), we have $(p_{\lfloor n/2 \rfloor} - p_r)/2 = p_{\text{mid}} - p_r \leq c' - p$ for every point $p \in B_R$, and in case $p_{\text{mid}} > c'$ (see Fig. 4.5(bottom)) we have

$(p_{\lfloor n/2 \rfloor} - p_r)/2 = p_{\lfloor n/2 \rfloor} - p_{\text{mid}} < p - c'$, for every point $p \in P_r$ (the $\lceil n/2 \rceil \geq |B_R|$ rightmost points of P). Since $B_R, P_r \subseteq P_R$, we conclude that, in any case, there are at least $|B_R|$ points in P_R that have a distance at least $(p_{\lfloor n/2 \rfloor} - p_r)/2$ to c'. In other words, $|B_R|\,(p_{\lfloor n/2 \rfloor} - p_r) \leq 2\nu_{c'}(P_R)$. Then, for $\delta \geq 144$, the right-hand side of (4.13) is at most $(\varepsilon/18) \cdot \nu_{c'}(P_R)$. This concludes the proof of (ii) for the non-squared distances.

For squared distances, we have $\mu_{c'}(B) \leq |B| \cdot (D + |\mathcal{I}(B_i)|)^2 \leq 2|B| \cdot D^2 + 2|B| \cdot |\mathcal{I}(B_i)|)^2$. The first term of the right hand side is bounded by $(\varepsilon/36)\mu_{c'}(P_R)$ as in case (*i*). For the second term we replace (4.13) by

$$2\,|B| \cdot |\mathcal{I}(B_i)|^2 \leq \frac{4\varepsilon}{\delta}\,|B_R| \cdot 4(p_{\lfloor n/2 \rfloor} - p_r)^2. \tag{4.14}$$

As already explained, there are at least $|B_R|$ points in P_R that have a distance of at least $(p_{\lfloor n/2 \rfloor} - p_r)/2$ to c'. In other words, $|B_R|\,(p_{\lfloor n/2 \rfloor} - p_r)^2 \leq 4\mu_{c'}(P_R)$. Substituting this in (4.14) concludes the proof of (ii) for $\delta \geq 1152$. We note that we made no real attempt to optimize the choice of δ. In fact, for case (i), $\delta \geq 144$ suffices. □

Lemma 4.5 implies the following main result of this section, which is a trivial modification of Theorem 4.9.

Theorem 4.11. *Let P be a set of n points on a line in $\mathbb{R}^d$, $k \geq 1$ an integer, and $\varepsilon > 0$. The algorithm* V-CORESET$(P, 2k-1, \varepsilon/3)$ *returns, in $O(ndk)$ time, a weighted-facilities (k, ε)-coreset for P, of size $|S| = 2^{O(k)}\varepsilon^{-2k+1}\log^{4k-3} n$.*

4.4 Coresets for $P \subseteq \mathbb{R}^d$

So far we have constructed (k, ε)-coresets for a set of points on a fixed line (and for weighted point facilities). In this section we use these coresets to construct the following kind of coreset for a set of points in $\mathbb{R}^d$.

(k, j, ε)-Coreset. Let P be a set of n points in $\mathbb{R}^d$, $k \geq 1$ and $1 \leq j \leq d-1$ integers. A weighted set $S \subset P$ is called a (k, j, ε)-*coreset* for P if all the following properties hold.

(*i*) For any set L of any number $0 \leq k' \leq k$ of lines and of at most $k - k'$ points, we have

$$(1-\varepsilon)\nu_L(P) \leq \nu_L(S) \leq (1+\varepsilon)\nu_L(P).$$

(*ii*) For any flat h of dimension at most j, we have

$$(1-\varepsilon)\nu_{\{h\}}(P) \leq \nu_{\{h\}}(S) \leq (1+\varepsilon)\nu_{\{h\}}(P).$$

(*iii*) Properties (i) and (ii) hold for squared distances, and for regression distances to a hyperplane (squared and non-squared).

Our construction of this coreset crucially relies on a randomized bi-criteria constant-factor approximation algorithm APPROX-K-J-FLATS2(P, k, j, δ), which is an enhancement of the algorithm presented in Chapter 3. The procedure APPROX-K-J-FLATS2 receives as input a point set $P \subset \mathbb{R}^d$, $\delta > 0$, and integers k, j, such that $k \geq 1$ and $1 \leq j \leq d-1$. It outputs a set $F = \{f_1, f_2, \ldots, f_m\}$ of $m = \log(1/\delta)\log n \cdot (jk\log\log n)^{O(j)}$ j-dimensional flats and a partition $\Pi = \{P_1, P_2, \ldots, P_m\}$ of P, such that, with probability at least $1-\delta$, for any set Y of at most k flats, all of dimensions at most j,

$$\sum_{i=1}^{m} \nu_{f_i}(P_i) \leq 2^{j+2} \cdot \nu_Y(P) \quad \text{and} \quad \sum_{i=1}^{m} \mu_{f_i}(P_i) \leq 2^{2j+3} \cdot \mu_Y(P). \tag{4.15}$$

In particular, F is a constant-factor approximation of the "k j-flat-median" and "k j-flat-mean" problems (if we regard j as a constant); it is a bi-criteria approximation in that it produces $m \gg k$ j-flats which yield a median cost and a mean cost which lie within constant factors of the optimal such costs involving k j-flats.

The algorithm APPROX-K-J-FLATS2(P, k, j, δ) makes $\lceil\log(1/\delta)\rceil$ calls to the procedure APPROX-K-J-FLATS(P, k, j) that is described in Chapter 3. It then returns the union F of the two output sets of flats $F^\nu = \{f_1^\nu, f_2^\nu, \ldots\}$ and $F^\mu = \{f_1^\mu, f_2^\mu, \ldots\}$ (among the $\lceil\log(1/\delta)\rceil$ outputs) that minimize $\sum_{i=1}^m \nu_{f_i^\nu}(P_i)$, and $\sum_{i=1}^m \mu_{f_i^\mu}(P_i)$, respectively. The algorithm also returns the partition Π of P that is defined as follows.

Consider the partition $\{R_1, R_2, \ldots, R_{t_{\max}}\}$ of P that is computed during a call to the procedure APPROX-K-J-FLATS(P, k, j); see Fig. 3.1. Let $\Pi^\nu = \{R_1^\nu, R_2^\nu, \ldots\}$ and $\Pi^\mu = \{R_1^\mu, R_2^\mu, \ldots\}$ be such two partitions of P that correspond to the calls that returned F_ν and F_μ, respectively. Put s and t such that $1 \leq s \leq |\Pi^\nu|$ and $1 \leq t \leq |\Pi^\mu|$. For every $p \in R_s^\nu$, let $p^\nu = \operatorname{dist}(p, f_s^\nu)$. Similarly, for every $p \in R_t^\mu$, let $p^\mu = \operatorname{dist}(p, f_t^\mu)$. We define $P_s^\nu = \{p \in R_s^\nu \mid p^\nu \leq p^\mu\}$, and $P_t^\mu = \{p \in R_t^\mu \mid p^\mu < p^\nu\}$. Finally, we define the partition Π of P to be the union of $\left\{P_1^\nu, \ldots, P_{|\Pi^\nu|}^\nu\right\}$ with $\left\{P_1^\mu, \ldots, P_{|\Pi^\mu|}^\mu\right\}$.

The two equations in (4.15) hold for Π and F, since for every set Y of at most k flats, all of dimension at most j, we have

$$\sum_{i=1}^{|\Pi^\nu|} \nu_{f_i^\nu}(P_i^\nu) + \sum_{i=1}^{|\Pi^\mu|} \nu_{f_i^\mu}(P_i^\mu) \leq \sum_{i=1}^{|\Pi^\nu|} \nu_{f_i^\nu}(R_i^\nu) \leq 2^{j+2} \cdot \nu_Y(P),$$

where the last derivation is by applying Theorem 3.1 with $v = 1$. Similarly, by applying Theorem 3.1 with $v = 2$, we have

$$\sum_{i=1}^{|\Pi^\nu|} \mu_{f_i^\nu}(P_i^\nu) + \sum_{i=1}^{|\Pi^\mu|} \mu_{f_i^\mu}(P_i^\mu) \leq \sum_{i=1}^{|\Pi^\nu|} \mu_{f_i^\nu}(R_i^\nu) \leq 2^{j+3} \cdot \nu_Y(P).$$

In order to compute Π, we modify the procedure APPROX-K-J-FLATS(P, k, j) in Fig. 3.1 so that in Line 6 the distance $\mathrm{dist}(p, F')$ is associated and stored in memory for every point $p \in R_t$. These distances are then returned as output of APPROX-K-J-FLATS together with F. By comparing the output distances for the two calls that returned F_μ and F_ν for every $p \in P$, we can compute Π in time $O(n)$. The algorithm APPROX-K-J-FLATS2(P, k, j, δ) thus takes $nd \cdot (2jk)^{O(j)} \cdot \log(1/\delta)$ time overall; see Chapter 3 for details.

Since the algorithm APPROX-K-J-FLATS(P, k, j) succeeds with probability at least $1/2$, the algorithm APPROX-K-J-FLATS2(P, k, j, δ) succeeds with probability at least $1 - 1/2^{\lceil \log(1/\delta) \rceil} \geq 1 - \delta$.

Algorithm LINEAR-FACILITIES-CORESET$(P, k, j, \varepsilon, \delta)$
Input: A set of points $P \subset \mathbb{R}^d$, $\delta > 0$, $0 < \varepsilon \leq 1$, and integers k, j, where $k \geq 1$ and $1 \leq j \leq d-1$.
Output: A set $S \subseteq P$ with the properties stated in Theorem 4.13 below.

1 $(F, \Pi) \leftarrow$ APPROX-K-J-FLATS2$(P, k, j, \delta/2)$
2 $S \leftarrow \emptyset$
3 **for** $i \leftarrow 1$ to $|F|$
4 **do** $f_i \leftarrow$ the ith j-dimensional flat in F
5 $P_i \leftarrow$ the ith set of points in Π
6 $f_i^{\perp} \leftarrow$ an arbitrary $(d-j)$-dimensional flat orthogonal to f_i
▷ See Fig. 4.6(top)
7 $P_i^* \leftarrow \mathrm{proj}(P_i, f_i^{\perp})$
8 **for** each $p^* \in P_i^*$ **do** $w(p^*) \leftarrow |P| \,/\, |P_i|$
9 $S_i^* \leftarrow$ SINGLE-FACILITY-CORESET$(P_i^*, \varepsilon/9)$
10 $\mathcal{F} \leftarrow \emptyset$
11 **for** each $p' \in S_i^*$
12 **do** $f \leftarrow$ the j-dimensional flat that passes through p' and is parallel to f_i
▷ See Fig. 4.6(bottom)
13 $P_f \leftarrow$ the set of those $p \in P_i$, such that p' is the representative of $p^* = \mathrm{proj}(p, f_i^{\perp})$ in S_i^* ▷ See Line 9 of SINGLE-FACILITY-CORESET
14 $\tilde{P}_f \leftarrow \mathrm{proj}(P_f, f)$
15 $\mathcal{F} \leftarrow \mathcal{F} \cup \{f\}$
16 **for** each $f \in \mathcal{F}$
17 **do if** $j = 1$
18 **then** $\tilde{S}_f \leftarrow$ V-CORESET$(\tilde{P}_f, 2k-1, \varepsilon/3)$
19 **else** $\tilde{S}_f \leftarrow$LINEAR-FACILITIES-CORESET$\big(\tilde{P}_f, 1, j-1, \varepsilon, \delta/(2\,|\mathcal{F}|)\big)$
20 $S \leftarrow S \cup \{p \in P_f \mid \mathrm{proj}(p, f) \in \tilde{S}_f\}$
▷ each point in S is assigned the weight of its corresponding point in $\tilde{S}_f$
21 **return** S

Overview of the algorithm LINEAR-FACILITIES-CORESET. In this algorithm, which is described in Fig. 4.6, $\mathrm{proj}(q, X)$ denotes the projection of a point q on a set of points X (i.e., $\mathrm{proj}(q, X)$ is the point of X nearest to q). For a set Q, we define $\mathrm{proj}(Q, X) = \{\mathrm{proj}(q, X) \mid q \in Q\}$.

In Line 1, the algorithm computes a set $F = \{f_1, f_2, \ldots\}$ of flats, and a corresponding partition $\Pi = \{P_1, P_2, \cdots\}$ of P as described above. In Lines 3–20, the

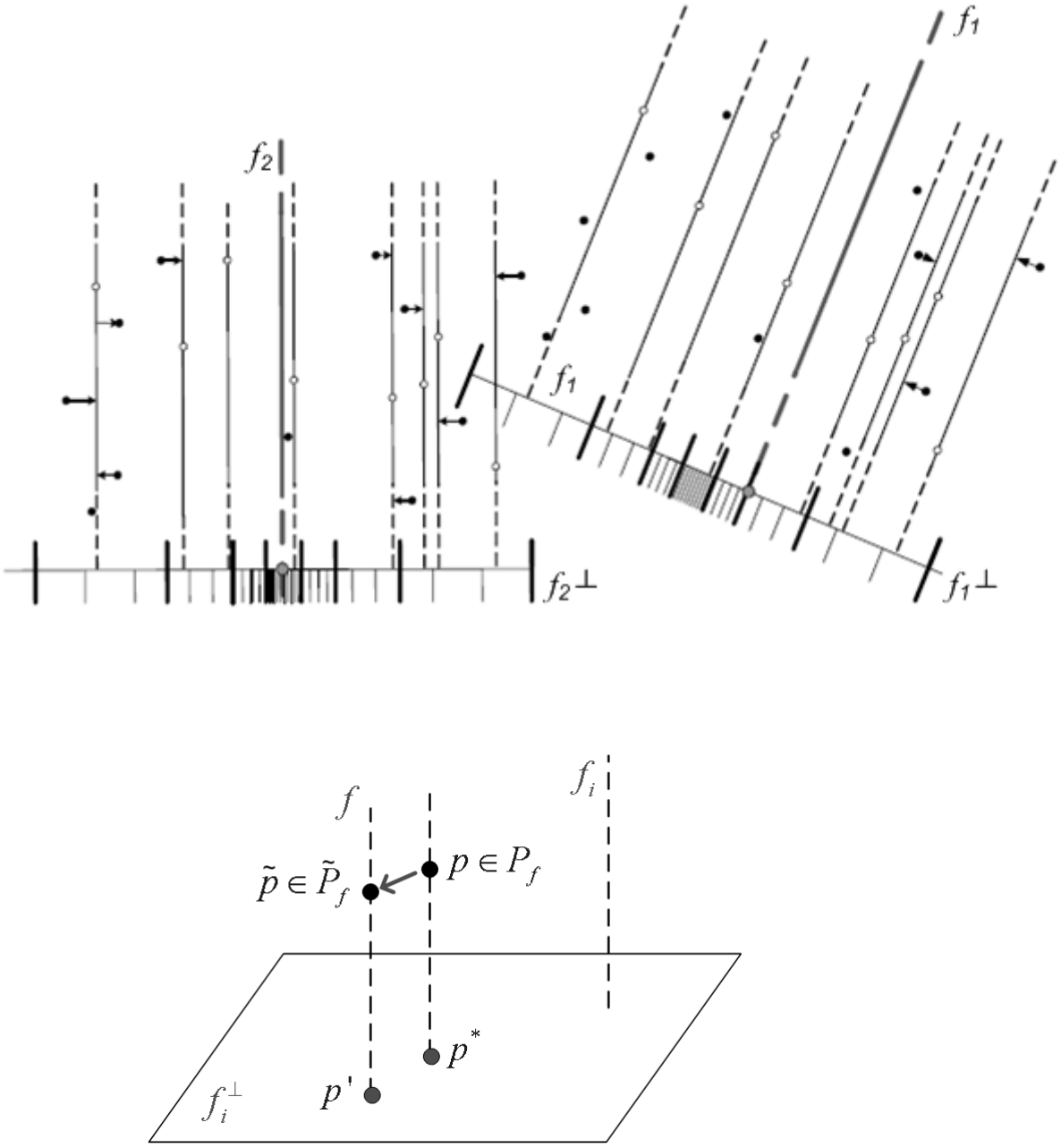


Fig. 4.6: **(top)** For $j = 1$ and $P \subset \mathbb{R}^2$, each $f_i \in F$ is a line, as its orthogonal $f_i^\perp$. **(bottom)** The various mappings used in the algorithm: p is a point of P_i (nearest to f_i); p^* is its projection onto $f_i^\perp$; p' is the representative of p^* in the coreset S_i^*; f passes through p' and is parallel to f_i; $\tilde{p}$ is the projection of p onto f.

algorithm then constructs a coreset for every set P_i independently, and outputs the union S of these coresets in Line 21.

The coreset for P_i is computed as follows. In Lines 4–7 we compute P_i^* which is the projection of P_i on the $(d-j)$-flat $f_i^\perp$ that is orthogonal to f_i. In Lines 8–9, we then construct for P_i^* an ε-coreset S_i^* for a single facility, as described in Section 4.1. Each projected point $p' \in S_i^*$ corresponds to a j-flat f of $\mathbb{R}^d$ that passes through p' and is parallel to f_i. This set $\mathcal{F}$ of j-flats is constructed in Lines 12–15. The set $P_f \subseteq P_i$ denotes the set of points that are (roughly speaking) closer to f than any other flat in $\mathcal{F}$. The set $\tilde{P}_f$ denotes the projection of P_f on f.

For every $f \in \mathcal{F}$ we construct in Lines 16–19 a coreset $\tilde{S}_f$ of $\tilde{P}_f$. If f is a line ($j = 1$), $\tilde{S}_f$ is a coresets for weighted facilities, as described in Section 4.3. Otherwise, $\tilde{S}_f$ is the output of a recursive call to LINEAR-FACILITIES-CORESET with the set $\tilde{P}_f$ instead of P, and $j-1$ instead of j. In both cases, the coreset $\tilde{S}_f$ consists of weighted points from $\tilde{P}_f$. That is, every weighted point in $\tilde{S}_f$ is a projection of a corresponding weighted point $p \in P_f$. In Line 20, we add the union of these weighted points of P_f to the output coreset S, .

Although the algorithm is formulated for arbitrary k and j, we apply (and analyze) it only for the two special cases $k = 1$ (coreset for a single j-flat) or $j = 1$ (coreset for k lines)[1]. Specifically we have:

Lemma 4.12. *Let P be a set of n points in $\mathbb{R}^d$, $0 < \varepsilon \leq 1$, $\delta > 0$, and k, j integers such that $k \geq 1$ and $1 \leq j \leq d-1$. Then*
(i) LINEAR-FACILITIES-CORESET$(P, k, 1, \varepsilon, \delta)$ *takes $nd\log(1/\delta)\cdot k^{O(1)}$ time and returns a set S of size*

$$\log^{4k} n \cdot \log(1/\delta) \cdot (\sqrt{d}^d/\varepsilon^{d+2k-1}) \cdot 2^{O(d+k)}.$$

(ii) LINEAR-FACILITIES-CORESET$(P, 1, j, \varepsilon, \delta)$ *takes $nd\log(1/\delta)\cdot j^{O(j)}$ time, and returns a set S of size*

$$(j\log n)^{O(j^2)}\left(\log\frac{1}{\delta} + d\log\frac{1}{\varepsilon} + d\log d + \log\log n\right)^j (\sqrt{d}/\varepsilon)^{dj+1}\cdot 2^{O(dj)}.$$

Proof. (i) The call to APPROX-K-J-FLATS2$(P, k, 1, \delta/2)$ in Line 1 returns a set F of $\log(1/\delta)\log n \cdot (k\log\log n)^{O(1)}$ lines, as noted above. For each line in F, the construction of a single facility coreset in Line 9 returns a set S_i^* of $2^{O(d)}\cdot$

[1]The general case $k, j > 1$ is not treated in the thesis, and it is still an open problem to solve it using similar techniques with comparable performance bounds. See Chapter 5.

$(\sqrt{d}/\varepsilon)^d \log n$ points (see Corollary 4.2). For each point in S_i^*, we construct in Line 18 a V-coreset of size $2^{O(k)}\varepsilon^{-2k+1}\log^{4k-3} n$ (see Theorem 4.11). Hence, the overall size of S is

$$\begin{aligned}\log(1/\delta)\log n \cdot (k\log\log n)^{O(1)} \cdot 2^{O(d)} \cdot (\sqrt{d}/\varepsilon)^d \log n \cdot 2^{O(k)}\varepsilon^{-2k+1}\log^{4k-3} n \\ = \log^{4k} n \cdot \log(1/\delta) \cdot (\sqrt{d}^d/\varepsilon^{d+2k-1}) \cdot 2^{O(d+k)}, \qquad (4.16)\end{aligned}$$

where we bound $(\log\log n)^{O(1)}$ by $\log n$ (for n sufficiently large). The call to APPROX-K-J-FLATS2$(P, k, 1, \delta/2)$ takes $nd\log(1/\delta)\cdot k^{O(1)}$ time. The execution of SINGLE-FACILITY-CORESET and V-CORESET can be performed in time $O(mdk)$ for a set of m points (see Corollary 4.3 and Theorem 4.9), i.e., in

$$O(dk)\sum_{f\in\mathcal{F}}|\tilde{P}_f| = O(dk)\sum_{f\in\mathcal{F}}|P_f| = O(ndk)$$

time. Hence, the overall time bound is dominated by the cost of the call to APPROX-K-J-FLATS2, and is therefore $nd\log(1/\delta)\cdot k^{O(1)}$ time.

(ii) Let $S_j(\delta)$ denote the overall size of the output set S (where the other parameters n, ε are fixed). By (4.16) we have

$$S_1(\delta) = \log^4 n \cdot \log(1/\delta) \cdot (\sqrt{d}^d/\varepsilon^{d+1}) \cdot 2^{O(d)}.$$

The call to APPROX-K-J-FLATS2$(P, 1, j, \delta/2)$ in Line 1 returns a set F of $\log(1/\delta)\cdot\log n\cdot(j\log\log n)^{O(j)}$ j-flats. For each j-flat in F, the call to SINGLE-FACILITY-CORESET in Line 9 returns a set S_i^* of $2^{O(d)}\cdot(\sqrt{d}/\varepsilon)^d\log n$ points (see Corollary 4.2). For each point in S_i^*, the call in Line 19 to the algorithm LINEAR-FACILITIES-CORESET$\big(\tilde{P}_f, k, j-1, \varepsilon, \delta/(2|\mathcal{F}|)\big)$ returns a set of size

$$S_{j-1}\left(\frac{\delta}{2|\mathcal{F}|}\right) = S_{j-1}\left(\frac{\delta}{2|S_i^*|}\right) = S_{j-1}\left(\frac{\delta\varepsilon^d}{2^{O(d)}\cdot(\sqrt{d})^d\log n}\right).$$

We thus have, for $j \geq 2$,

$$S_j(\delta) \leq \log(1/\delta)\cdot\log^2 n\cdot(j\log\log n)^{O(j)}\cdot 2^{O(d)}\cdot(\sqrt{d}/\varepsilon)^d\cdot S_{j-1}\left(\frac{\delta\varepsilon^d}{2^{O(d)}\cdot(\sqrt{d})^d\log n}\right).$$

Put $\beta = \dfrac{\varepsilon^d}{2^{O(d)}\cdot(\sqrt{d})^d\log n}$. Unfolding the above recurrence, we have

$$S_j(\delta) = \log^{O(j^2)} n\cdot(\sqrt{d}/\varepsilon)^{dj+1}\cdot 2^{O(dj+j^2\log j)}\cdot\prod_{i=0}^{j-1}\log\left(\frac{1}{\delta\beta^i}\right).$$

We estimate the product by replacing $1/(\delta\beta^i)$ by $1/(\delta\beta)^i$, for $i \geq 1$, to obtain an upper bound of

$$j! \log^j\left(\frac{1}{\delta\beta}\right) = 2^{O(j\log j)}\left(\log\frac{1}{\delta} + \log\left(\frac{2^{O(d)}(\sqrt{d})^d \log n}{\varepsilon^d}\right)\right)^j.$$

Hence, we have

$$S_j(\delta) = (j\log n)^{O(j^2)}\left(\log\frac{1}{\delta} + d\log\frac{1}{\varepsilon} + d\log d + \log\log n\right)^j (\sqrt{d}/\varepsilon)^{dj+1}\cdot 2^{O(dj)}.$$

Using Corollary 4.3, the running time of a single iteration of the recursion is dominated by the call to APPROX-K-J-FLATS2 with $k = 1$ in Line 1 , which is $nd\log(1/\delta)\cdot(2j)^{O(j)}$. Since the depth of the recursion is $O(j)$, and the points of P split among the recursive calls (in line 19), the total running time is

$$ndj\log(1/\delta)\cdot(2j)^{O(j)} = O(nd)\cdot\log(1/\delta)\cdot j^{O(j)}.$$

□

Theorem 4.13. *Let P be a finite set of points in $\mathbb{R}^d$. Let $0 < \varepsilon \leq 1/2$, $\delta > 0$, and let k, j be integers satisfying $k \geq 1$ and $1 \leq j \leq d-1$. Let $b_j = 2^{j^2+14j}$. Then*
(i) LINEAR-FACILITIES-CORESET$(P, k, 1, \varepsilon, \delta)$ *computes, with probability at least $1-\delta$, a $(k, 1, \varepsilon)$-coreset for P.*
(ii) LINEAR-FACILITIES-CORESET$(P, 1, j, \varepsilon/b_j, \delta)$ *computes, with probability at least $1-\delta$, a $(1, j, \varepsilon)$-coreset for P.*

Proof. To simplify the calculations, we assume (in (ii)) that LINEAR-FACILITIES-CORESET is called with parameters P, 1, j, ε, δ. At the end, we will replace ε by ε/b_j and obtain the asserted property. (Technically, we also have to do so in case (i), but then we replace ε by ε/b_0 and since $b_0 = 1$, no change is needed.)

Let Y be either an arbitrary set of any number $k' \leq k$ of lines and at most $k-k'$ points in $\mathbb{R}^d$, or a single flat of dimension at most j. We follow the notation in the procedure LINEAR-FACILITIES-CORESET. Let $\tilde{P} = \bigcup_{f\in\mathcal{F}} \tilde{P}_f$ and $\tilde{S} = \bigcup_{f\in\mathcal{F}} \tilde{S}_f$. We first bound $\left|\nu_Y(P) - \nu_Y(\tilde{P})\right|$ and $\left|\nu_Y(\tilde{S}) - \nu_Y(S)\right|$. Let $1 \leq i \leq |F|$, and f_i' denote the j-flat that is parallel to f_i and passes through the center of mass $\overline{P_i^*}$, given by $\overline{P_i^*} = \sum_{p\in P_i^*} p/\left|P_i^*\right|$. We define

$$R = \sum_{i=1}^{|F|} \frac{\sum_{p\in P_i^*}\left\|p - \overline{P_i^*}\right\|}{|P|} = \sum_{i=1}^{|F|}\frac{\nu_{f_i'}(P_i)}{|P|}. \tag{4.17}$$

As noted in Line 9 of SINGLE-FACILITY-CORESET (see Section 4.1), and since $w(P_i) = |P|$, we have the following bound for every $p \in P_i$, where p^* is its projection on $f_i^{\perp}$ and $p' \in S_i^*$ is the representative of p^* in the coreset S_i^*,

$$\begin{aligned} \|p^* - p'\| &\leq \varepsilon \cdot \max\left\{ \frac{\sum_{p \in P_i^*} \left\| p - \overline{P_i^*} \right\|}{w(P_i)}, \operatorname{dist}(p^*, \overline{P_i^*}) \right\} \\ &= \varepsilon \cdot \max\{R, \operatorname{dist}(p^*, \overline{P_i^*})\}. \end{aligned} \tag{4.18}$$

Let f be the j-dimensional flat that passes through p' and is parallel to f_i, and $\tilde{p} = \operatorname{proj}(p, f) \in \tilde{P}_f$ (see Fig. 4.6 (bottom)). From (4.18) we get

$$\|p - \tilde{p}\| \leq \varepsilon \cdot \max\{R, \operatorname{dist}(p, f_i')\}. \tag{4.19}$$

As noted in the proof of Corollary 4.2, for every $q \in \mathbb{R}^d$ we have $\nu_{\overline{P_i^*}}(P_i^*) \leq 2\nu_q(P_i^*)$. Substituting $q = f_i \cap f_i^{\perp}$, and noting that $\|p^* - q\| = \operatorname{dist}(p, f_i)$ and $\left\|p^* - \overline{P_i^*}\right\| = \operatorname{dist}(p, f_i')$, yield

$$\nu_{f_i'}(P_i) = \sum_{p \in P_i^*} \left\| p - \overline{P_i^*} \right\| \leq 2 \sum_{p \in P_i^*} \|p - q\| = 2\nu_{f_i}(P_i). \tag{4.20}$$

Theorem 3.1, and the way APPROX-K-J-FLATS2 is implemented, imply that the output of the call to APPROX-K-J-FLATS2 in Line 1 satisfies, with probability at least $1 - \delta/2$,

$$\sum_{i=1}^{|F|} \nu_{f_i}(P_i) \leq 2^{j+2} \cdot \nu_Y(P). \tag{4.21}$$

For the rest of the proof we assume that this equation holds. Together with (4.20) we get

$$\sum_{i=1}^{|F|} \nu_{f_i'}(P_i) \leq 2 \sum_{i=1}^{|F|} \nu_{f_i}(P_i) \leq 2^{j+3} \cdot \nu_Y(P) = 2^{j+3} \cdot \nu_Q(P), \tag{4.22}$$

where $Q = \bigcup_{y \in Y} y$ (note that replacing Y by its union Q does not affect the quantities $\nu_Y(P)$). Substitute in Lemma 4.1 $S = \tilde{P}$, $C_i = f_i'$ (for $1 \leq i \leq |F|$), $\alpha = 2^{j+3}$ and Q as just defined. Using (4.19) and (4.22), the lemma implies

$$\left| \nu_Y(P) - \nu_Y(\tilde{P}) \right| \leq 2^{j+4} \varepsilon \nu_Y(P). \tag{4.23}$$

We now bound $\left|\nu_Y(\tilde{S}) - \nu_Y(S)\right|$. Let $\tilde{P}_i$ denote the set of points $\tilde{p} \in \tilde{P}$ such that $p \in P_i$, and define $\tilde{S}_i = \tilde{S} \cap \tilde{P}_i$. For each $f \in \mathcal{F}$ (constructed at the ith iteration), since f and f'_i are parallel, we have, for every $p \in f$, $\operatorname{dist}(p, f'_i) = \left\|p^* - \overline{P_i^*}\right\|$, so $\nu_{f'_i}(P_i) = \nu_{\overline{P_i^*}}(P_i^*)$ and $\nu_{\overline{P_i^*}}(S_i^*) = \nu_{f'_i}(\tilde{S}_i)$. By Corollary 4.2, S_i^* is an ε-coreset of P_i^*, so we have $\nu_{\overline{P_i^*}}(P_i^*) \leq (1+\varepsilon)\nu_{\overline{P_i^*}}(S_i^*)$, and thus

$$\begin{aligned} \nu_{f'_i}(P_i) = \nu_{\overline{P_i^*}}(P_i^*) &\leq (1+\varepsilon)\nu_{\overline{P_i^*}}(S_i^*) \\ &\leq 2\nu_{\overline{P_i^*}}(S_i^*) = 2\nu_{f'_i}(\tilde{S}_i). \end{aligned} \tag{4.24}$$

We define $R' = \sum_{i=1}^{|F|} \nu_{f'_i}(\tilde{S}_i)/\left|P\right|$ and, using (4.17) and (4.24), get

$$R = \sum_{i=1}^{|F|} \frac{\nu_{f'_i}(P_i)}{|P|} \leq \sum_{i=1}^{|F|} \frac{2\nu_{f'_i}(\tilde{S}_i)}{|P|} = 2R'.$$

By the construction of S_i^* we then have for every $p \in S$, similarly to (4.19),

$$\begin{aligned} \|\tilde{p} - p\| &\leq \varepsilon \cdot \max\{R, \operatorname{dist}(\tilde{p}, f'_i)\} \\ &\leq 2\varepsilon \cdot \max\{R', \operatorname{dist}(\tilde{p}, f'_i)\}. \end{aligned} \tag{4.25}$$

Similarly to (4.24), we have $\nu_{f'_i}(\tilde{S}_i) \leq 2\nu_{f'_i}(P_i)$. Using (4.22), this yields

$$\sum_{i=1}^{|F|} \nu_{f'_i}(\tilde{S}_i) \leq \sum_{i=1}^{|F|} 2\nu_{f'_i}(P_i) \leq 2^{j+4} \cdot \nu_Y(P).$$

Using (4.25) and the last equation, we substitute in Lemma 4.1 $P = \tilde{S}$, $C_i = f'_i$ (for $1 \leq i \leq |F|$), $Q = Y$, and replace ε by 2ε, to obtain

$$\left|\nu_Y(\tilde{S}) - \nu_Y(S)\right| \leq 2^{j+5}\varepsilon\nu_Y(\tilde{S}). \tag{4.26}$$

We will show below that for each j-flat $f \in \mathcal{F}$ we have

$$\left|\nu_Y(\tilde{P}_f) - \nu_Y(\tilde{S}_f)\right| \leq b_{j-1}\varepsilon\nu_Y(\tilde{P}_f) \tag{4.27}$$

with probability at least $1 - \dfrac{\delta}{2\left|\mathcal{F}\right|}$. For the rest of the proof we assume that (4.27) holds simultaneously for all $f \in \mathcal{F}$, which happens with probability at least $1 -$

$\delta/2$. We then have

$$\begin{aligned}\left|\nu_Y(\tilde{P}) - \nu_Y(\tilde{S})\right| &= \left|\sum_{f\in\mathcal{F}} \nu_Y(\tilde{P}_f) - \sum_{f\in\mathcal{F}} \nu_Y(\tilde{S}_f)\right| \\ &\leq \sum_{f\in\mathcal{F}} \left|\nu_Y(\tilde{P}_f) - \nu_Y(\tilde{S}_f)\right| \\ &\leq \sum_{f\in\mathcal{F}} b_{j-1}\varepsilon\nu_Y(\tilde{P}_f) = b_{j-1}\varepsilon\nu_Y(\tilde{P}).\end{aligned} \tag{4.28}$$

By (4.23) we have

$$\nu_Y(\tilde{P}) \leq (1 + 2^{j+4}\varepsilon)\nu_Y(P) \leq 2^{j+5}\nu_Y(P).$$

Using this with (4.28) yields

$$\left|\nu_Y(\tilde{P}) - \nu_Y(\tilde{S})\right| \leq b_{j-1}\varepsilon\nu_Y(\tilde{P}) \leq b_{j-1}2^{j+5}\varepsilon\nu_Y(P). \tag{4.29}$$

Combining this with (4.23) we thus have

$$\begin{aligned}\left|\nu_Y(P) - \nu_Y(\tilde{S})\right| &\leq \left|\nu_Y(P) - \nu_Y(\tilde{P})\right| + \left|\nu_Y(\tilde{P}) - \nu_Y(\tilde{S})\right| \\ &\leq b_{j-1}2^{j+6}\varepsilon\nu_Y(P),\end{aligned} \tag{4.30}$$

which implies

$$\nu_Y(\tilde{S}) \leq \nu_Y(P) + b_{j-1}2^{j+6}\varepsilon\nu_Y(P) \leq b_{j-1}2^{j+7}\nu_Y(P). \tag{4.31}$$

By (4.26) together with the last equation we have,

$$\left|\nu_Y(\tilde{S}) - \nu_Y(S)\right| \leq 2^{j+5}\varepsilon\nu_Y(\tilde{S}) \leq b_{j-1}2^{2j+12}\varepsilon\nu_Y(P).$$

Combining this and (4.30), we get

$$\begin{aligned}|\nu_Y(P) - \nu_Y(S)| &\leq \left|\nu_Y(P) - \nu_Y(\tilde{S})\right| + \left|\nu_Y(\tilde{S}) - \nu_Y(S)\right| \\ &\leq b_{j-1}2^{j+6}\varepsilon\nu_Y(P) + b_{j-1}2^{2j+12}\varepsilon\nu_Y(P) \\ &\leq b_{j-1}2^{2j+13}\varepsilon\nu_Y(P) = b_j\varepsilon\nu_Y(P),\end{aligned} \tag{4.32}$$

by definition of b_i, and under the two assumptions that (4.27) holds for every $f \in \mathcal{F}$, and also that (4.21) holds. Since each of these assumptions holds with

probability at least $1-\delta/2$, S is a $(k, j, b_j\varepsilon)$-coreset for P with probability at least $1-\delta$. By rescaling ε as in the statement of the theorem, the asserted property follows.

It is left to prove that (4.27) holds with probability at least $1-\delta/(2\,|\mathcal{F}|)$. We argue differently in case (i) and in case (ii).

Case (i): Here $j = 1$. Hence, the set $\tilde{S}_f$ in Line 18 is a weighted facilities (k, ε)-coreset for $\tilde{P}_f$ (by Theorem 4.11). Let Y be an arbitrary set of $k' \leq k$ lines and at most $k - k'$ points in $\mathbb{R}^d$. We then have (recall that $b_0 = 1$)

$$\left|\nu_Y(\tilde{P}_f) - \nu_Y(\tilde{S}_f)\right| \leq \varepsilon\nu_Y(\tilde{P}_f) = b_0\varepsilon\nu_Y(\tilde{P}_f),$$

with probability 1, which proves (4.27) for case (i).

Case (ii): Let M be a finite set of points that is contained in a $(j+1)$-flat in $\mathbb{R}^d$. We prove by induction on j that LINEAR-FACILITIES-CORESET$(M, 1, j, \varepsilon, \delta)$ returns a $(1, j, b_j\varepsilon)$-coreset S for M, with probability at least $1-\delta$. By substituting $M = \tilde{P}_f$, and replacing j by $j-1$, and δ by $\delta/(2|\mathcal{F}|)$, and by noting that in this case $S = \tilde{S}_f$, we obtain (4.27). Note that this claim is a restricted version of the theorem itself, where in this version we only consider sets M that lie in a $(j+1)$-flat.

The base case $j = 1$ is an instance of case (i), whose proof (in general) has just been completed. For $j \geq 2$, inductively assume that LINEAR-FACILITIES-CORESET$(M, 1, j-1, \varepsilon, \delta/(2|\mathcal{F}|))$ returns a $(1, j-1, b_{j-1}\varepsilon)$-coreset S for M, with probability at least $1-\delta/(2\mathcal{F})$. By substituting $M = \tilde{P}_f$ and noting that in this case $S = \tilde{S}_f$, this proves (4.27) for the case where the dimension of Y is at most $(j-1)$.

We need to argue, though, that (4.27) holds for every fixed j-flat Y, with the same asserted probability. For this we make use of Lemma 4.16, given in Section 4.5 below, which replaces Y by a lower-dimensional flat which preserves distances to points in the flat containing M, up to a fixed multiplicative weight.

Let Y be a j-flat, and apply Lemma 4.15 with f and with $g = Y$. If Y contains a translation of f then Y is a translation of f and (4.27) holds trivially (all distances of points on f to Y are the same). We may thus assume that Y is not a translation of f, so Lemma 4.15 applies, and yields a j'-flat Y' with $j' \leq j-1$ and a weight w, such that, for each $p \in f$, $\operatorname{dist}(p, Y) = w \cdot \operatorname{dist}(p, Y')$. Hence $\nu_Y(\tilde{P}_f) = w \cdot \nu_{Y'}(\tilde{P}_f)$ and $\nu_Y(\tilde{S}_f) = w \cdot \nu_{Y'}(\tilde{S}_f)$.

Since (4.27) holds for Y', it also holds for Y, as claimed. Moreover, if the success probability for f to satisfy (4.27) for all flats Y of dimension $\leq j-1$ is

at least $1 - \delta/2|\mathcal{F}|$, then this is also the probability for this to hold for all flats Y of dimension $\leq j$.

Now, since (4.27) holds for every j-flat Y, we can complete the proof of the whole theorem for M and conclude that LINEAR-FACILITIES-CORESET$(M, 1, j, \varepsilon, \delta)$ does indeed return a $(1, j, b_j\varepsilon)$-coreset for M with probability at least $1 - \delta$. This establishes the induction step and thus completes the proof of the claim. Consequently, as already noted, (4.27) is established in general, and this in turn completes the proof of Theorem 4.13 (in the general non-restricted case).

For squared distances the proof is similar, if we use everywhere the cost function μ instead of ν, and use Lemma 4.1(*ii*) instead of Lemma 4.1(*i*). □

4.5 Distances to j-Flats Can be Measured From $(j-1)$-Flats

In Chapter 2 we showed that the distance between a point p on a line to another line is equal to the distance from p to a weighted point c, where the location of c and its weight depend on the two lines, but not on p; see Fig. 2.3. We used this observation to show that, given a set of points P on a line, a line query can be replaced by a weighted point query.

In this section we generalize this observation, and prove that if p lies on a Δ-flat, any j-flat query can be replaced by a weighted $(\Delta - 1)$-flat query. As in the above case for lines, the weight and the location of the flat are independent of the point p, but only depends on the two input flats; see Lemma 4.15 below. This lemma is used in the proof of Lemma 4.13 for constructing coresets that approximate the distances from points in $\mathbb{R}^d$ to a single j-flat. Lemma 4.15 was recently used in [FMSW10], in order to construct smaller coresets (of size linear or independent of d) for approximating a point set in a high-dimensional space by a single j-flat.

In this section, a Δ-flat f is represented by a matrix F of size $d \times \Delta$, whose columns are mutually orthogonal unit vectors, and by a column vector $f_0 \in \mathbb{R}^d$ that represents the translation of f from the origin. Formally, we define $\text{flat}(F, f_0) = f = \left\{F \cdot p + f_0 \mid p \in \mathbb{R}^\Delta\right\}$. The dimension of a Δ-flat f is denoted by $\dim(f) = \Delta$.

Theorem 4.14 (Singular Value Decomposition [Pea01])**.** *Let A be any matrix of size $d \times \Delta$, for some $d \geq \Delta \geq 1$. Then there are two unitary matrices $U_{d\times d}$, and*

$V_{\Delta\times\Delta}$, and a matrix $D_{d\times\Delta}$, such that the following properties hold.

(i) $A = UDV^T$.

(ii) D *is a diagonal matrix. That is,* $D_{i,j} = 0$ *for every* $i \neq j$, $1 \leq i \leq d$, $1 \leq j \leq \Delta$.

(iii) $D_{1,1} \geq \cdots \geq D_{\Delta,\Delta} \geq 0$.

Lemma 4.15 ([FL08])**.** *Let f be a Δ-flat in $\mathbb{R}^d$ for some $1 \leq \Delta \leq d-1$. Let g be a flat in $\mathbb{R}^d$ of any dimension such that g does not contain a translation of f. Let (h, w) denote the flat h and the constant $w > 0$ which are the output of the algorithm* WEIGHTED-FLAT(f, g)*; see Fig 4.7. Then h is a flat of dimension at most $\Delta - 1$, and for each $p \in f$ we have*

$$\operatorname{dist}(p, g) = w \cdot \operatorname{dist}(p, h).$$

Proof. We define all the variables in this proof in the same way as in Fig. 4.7. If $\dim(g) \leq \Delta - 1$ then Lemma 4.15 trivially holds; see Line 1–2 in Fig. 4.7. We therefore assume that $\dim(g) \geq \Delta$ for the rest of this proof. By the assumption in the lemma, g does not contain a translation of f, thus $F \neq GG^TF$, i.e, $D_{1,1} > 0$. Therefore, h and h_i, for $1 \leq i \leq \Delta$, are well defined; see Lines 12, 17, and 18 in Fig. 4.7.

Since V is a unitary matrix, $v_1, \ldots, v_\Delta$ are mutually orthogonal unit vectors. Also, by construction, F^TF is the $\Delta \times \Delta$ identity matrix. It follows that

$$f_i^T f_j = (Fv_i)^T(Fv_j) = v_i^T F^T F v_j = v_i^T v_j = 0, \text{ for } 1 \leq i < j \leq \Delta. \tag{4.33}$$

We also have, for every $1 \leq i \leq \Delta$, $\|f_i\| = \|Fv_i\| = \|v_i\| = 1$. By (4.33) and the last equation, we conclude that $f_1, \ldots, f_\Delta$ are mutually orthogonal unit vectors that span $\operatorname{flat}(F, \vec{0})$. Put $f' = \operatorname{flat}(F, \vec{0})$, $g' = \operatorname{flat}(G, \vec{0})$. Let p be any point on f, and $p' = p - f_0$. Since $p' \in f'$, there is a vector $\alpha = (\alpha_1, \ldots, \alpha_\Delta)^T \in \mathbb{R}^\Delta$ such that $p' = \sum_{i=1}^{\Delta} \alpha_i f_i = \sum_{i=1}^{\Delta} \alpha_i F v_i = FV\alpha$. For evert $x \in \mathbb{R}^d$ we have $\operatorname{proj}(x, g') = GG^Tx$. Using Theorem 4.14, it follows that

$$\begin{aligned}
\operatorname{dist}(p', g') &= \|p' - \operatorname{proj}(p, g')\| = \left\|p' - GG^Tp'\right\| \\
&= \left\|FV\alpha - GG^TFV\alpha\right\| = \left\|(F - GG^TF)V\alpha\right\| \\
&= \left\|UDV^TV\alpha\right\| = \|D\alpha\| = \sqrt{\sum_{i=1}^{\delta}(D_{i,i}\alpha_i)^2}.
\end{aligned} \tag{4.34}$$

Algorithm WEIGHTED-FLAT(f, g)

Input. A Δ-flat $f = \text{flat}(F, f_0)$, for some $1 \leq \Delta \leq d-1$, and a flat $g = \text{flat}(G, g_0)$ of any dimension, such that g does not contain a translation of f.

Output. A pair (h, w), where h is a flat of dimension at most $\Delta - 1$, and $w \geq 0$ is a constant, such that for each $p \in f$ we have $\text{dist}(p, g) = w \cdot \text{dist}(p, h)$.

if $\dim(g) \leq \Delta - 1$
  then return $(g, 1)$
$(U, D, V) \leftarrow$ A tuple of three matrices that satisfy Theorem 4.14, with the $d \times \Delta$ matrix $A = F - GG^T F$
$\delta \leftarrow \left|\left\{1 \leq i \leq \Delta \mid D_{i,i} > 0\right\}\right|$
for $i \leftarrow 1$ **to** $\Delta + \delta$
  if $i \leq \Delta$
    then $v_i \leftarrow$ the ith column of V.
      $f_i \leftarrow F \cdot v_i$
    else $f_i \leftarrow$ an arbitrary unit vector in $\mathbb{R}^d$ that is orthogonal to f_j for all $1 \leq j < i$.
      $\triangleright$ There exists such a vector f_i, as explained in the proof of Lemma 4.15.
for $i \leftarrow 2$ **to** Δ
    if $i \leq \delta$
      then $h_{i-1} \leftarrow f_i \cdot \sqrt{1 - \left(\frac{D_{i,i}}{D_{1,1}}\right)^2} + f_{i+\Delta-1} \cdot \frac{D_{i,i}}{D_{1,1}}$
      else $h_{i-1} \leftarrow f_i$
$H \leftarrow$ A matrix of size $d \times (\Delta - 1)$ whose ith column is h_i, for every $1 \leq i \leq \Delta - 1$.
for each $i \leftarrow 1$ to d
  if $i \leq \delta$
    then $y_i \leftarrow \frac{\left[U^T(f_0 - GG^T f_0 - g_0 + GG^T g_0)\right]_i}{D_{i,i}}$

      $\triangleright$ For a vector x, we denote by $[x]_i$ the ith entry of x.

    else $y_i \leftarrow \frac{\left[U^T(f_0 - GG^T f_0 - g_0 + GG^T g_0)\right]_i}{D_{1,1}}$
$h_0 = f_0 - HH^T f_0 - \sum_{i=1}^{\Delta} y_i(f_i - HH^T f_i) - f_{\Delta+\delta}\sqrt{\sum_{i=\delta+1}^{d} y_i^2}$
$h \leftarrow \text{flat}(H, h_0)$
return $(h, D_{1,1})$

Fig. 4.7: The algorithm WEIGHTED-FLAT

It follows that $p' \in f' \cap g'$ if and only if $\alpha_i = 0$ for every $1 \leq i \leq \delta$. Since $p' = \sum_{i=1}^{\Delta} \alpha_i f_i$, this implies $\dim(f' \cap g') = \Delta - \delta$. We assumed $\dim(g') = \dim(g) \geq \Delta$, thus

$$d \geq \dim(f') + \dim(g') - \dim(f' \cap g') \geq 2\Delta - (\Delta - \delta) = \Delta + \delta. \tag{4.35}$$

Combining (4.33) and (4.34), we conclude that there are δ vectors $f_{\Delta+1}, \ldots, f_{\Delta+\delta}$, such that $f_1, \ldots, f_{\Delta+\delta}$ are mutually orthogonal unit vectors. This proves the claim in the comment for Line 9 of the algorithm WEIGHTED-FLAT; see Fig. 4.7.

It is left to prove that $\operatorname{dist}(p, g) = w \cdot \operatorname{dist}(p, h)$. Similarly to (4.34), we have

$$\begin{aligned}
\operatorname{dist}(p, g) &= \operatorname{dist}\big(p - g_0, g'\big) = \big\|p - g_0 - GG^T(p - g_0)\big\| \\
&= \big\|p - GG^T p - g_0 + GG^T g_0\big\| \\
&= \big\|f_0 + p' - GG^T(f_0 + p') - g_0 + GG^T g_0\big\| \\
&= \big\|f_0 + FV\alpha - GG^T(f_0 + FV\alpha) - g_0 + GG^T g_0\big\| \\
&= \big\|(F - GG^T F)V\alpha + f_0 - GG^T f_0 - g_0 + GG^T g_0\big\| \\
&= \big\|UDV^T V\alpha + f_0 - GG^T f_0 - g_0 + GG^T g_0\big\| \\
&= \big\|D\alpha + U^T(f_0 - GG^T f_0 - g_0 + GG^T g_0)\big\| .
\end{aligned} \tag{4.36}$$

For every $0 \leq i \leq \Delta$, the vector $f_i - HH^T f_i = f_i - \operatorname{proj}(f_i, H)$ is orthogonal to H. Moreover, since f_1 is "used" in the construction of H (see Lines 10–13 of Fig. 4.7), f_1 is orthogonal to H. Since $f_{\Delta+\delta}$ is also orthogonal to H (for the same reason), we have $HH^T h_0 = \operatorname{proj}(h_0, H) = 0$. It follows that

$$\begin{aligned}
\operatorname{dist}(p, h) &= \operatorname{dist}\big(p - h_0, \operatorname{flat}(H, \vec{0})\big) = \big\|p - h_0 - HH^T(p - h_0\big\| \\
&= \big\|p - h_0 - HH^T p\big\| = \big\|(f_0 + p') - h_0 - HH^T(p' + f_0)\big\| \\
&= \left\| p' - HH^T p' + \sum_{i=1}^{\Delta} y_i(f_i - HH^T f_i) + f_{\Delta+\delta}\sqrt{\sum_{i=\delta+1}^{d} y_i^2} \right\| \\
&= \left\| \sum_{i=1}^{\Delta} \alpha_i f_i - HH^T \sum_{i=1}^{\Delta} \alpha_i f_i + \sum_{i=1}^{\Delta} y_i\big(f_i - HH^T f_i\big) + f_{\Delta+\delta}\sqrt{\sum_{i=\delta+1}^{d} y_i^2} \right\| \\
&= \left\| \sum_{i=1}^{\Delta} (\alpha_i + y_i)\big(f_i - HH^T f_i\big) + f_{\Delta+\delta}\sqrt{\sum_{i=\delta+1}^{d} y_i^2} \right\| .
\end{aligned} \tag{4.37}$$

By Lines 12–13, for every $1 \leq i \leq \Delta$ there is $\gamma_i \in \mathbb{R}$ such that $HH^T f_i = \gamma_i f_i$. Also, $HH^T f_1 = 0$. Using (4.33), we thus have that $f_2 - HH^T f_2, \ldots, f_\Delta - HH^T f_\Delta$, $f_1, f_{\Delta+\delta}$ are mutually orthogonal vectors. Equation (4.37) can thus be rewritten as

$$\begin{aligned}
\operatorname{dist}(p,h) &= \left\| (\alpha_1 + y_1) f_1 + \sum_{i=2}^{\Delta} (\alpha_i + y_i)(f_i - HH^T f_i) + f_{\Delta+\delta} \sqrt{\sum_{i=\delta+1}^{d} y_i^2} \right\| \\
&= \sqrt{(\alpha_1 + y_1)^2 \|f_1\|^2 + \sum_{i=2}^{\Delta} (\alpha_i + y_i)^2 \left\| f_i - h_{i-1} h_{i-1}^T f_i \right\|^2 + \|f_{\Delta+\delta}\|^2 \sum_{i=\delta+1}^{d} y_i^2} \\
&= \sqrt{(\alpha_1 + y_1)^2 + \sum_{i=2}^{\Delta} (\alpha_i + y_i)^2 \left\| f_i - h_{i-1} h_{i-1}^T f_i \right\|^2 + \sum_{i=\delta+1}^{d} y_i^2}. \qquad (4.38)
\end{aligned}$$

Fix i for some $2 \leq i \leq \Delta$, and define $x = D_{i,i}/D_{1,1}$. Hence, $h_{i-1} = f_i \cdot \sqrt{1-x^2} + f_{i+\Delta-1} \cdot x$. It follows that

$$\begin{aligned}
\left\| f_i - h_{i-1} h_{i-1}^T f_i \right\| &= \left\| f_i - h_{i-1}[(\sqrt{1-x^2} f_i + x f_{i+\Delta-1}) \cdot f_i] \right\| \\
&= \left\| f_i - \sqrt{1-x^2} \cdot h_{i-1} \right\| \\
&= \left\| f_i - \sqrt{1-x^2} \cdot (f_i \sqrt{1-x^2} + f_{i+\Delta-1} \cdot x) \right\| \\
&= \left\| f_i - (1-x^2) \cdot f_i - f_{i+\Delta-1} \cdot x\sqrt{1-x^2} \right\| \\
&= \left\| f_i \cdot x^2 - f_{i+\Delta-1} \cdot x\sqrt{1-x^2} \right\| \\
&= \sqrt{x^4 + x^2(1-x^2)} = x = \frac{D_{i,i}}{D_{1,1}}.
\end{aligned}$$

Substituting the last equation in (4.38) for every $2 \leq i \leq \Delta$, and using the values y_i as in Lines 17–18 of Fig 4.7, we get

$$\begin{aligned}
D_{1,1} \cdot \operatorname{dist}(p,h) &= D_{1,1} \sqrt{(\alpha_1 + y_1)^2 + \sum_{i=2}^{\Delta} (\alpha_i + y_i)^2 \left(\frac{D_{i,i}}{D_{1,1}} \right)^2 + \sum_{i=\delta+1}^{d} y_i^2} \\
&= \sqrt{\sum_{i=1}^{\Delta} (D_{i,i}\alpha_i + D_{i,i} y_i)^2 + \sum_{i=\delta+1}^{d} (D_{1,1} y_i)^2} \\
&= \left\| D\alpha + U^T (f_0 - GG^T f_0 - g_0 + GG^T g_0) \right\|.
\end{aligned}$$

Combining this with (4.36) yields $\mathrm{dist}(p, g) = D_{1,1} \cdot \mathrm{dist}(p, h)$. Since $w = D_{1,1}$ by Line 21 of WEIGHTED-FLAT, this concludes the proof of the lemma. □

Chapter 5

Conclusion and Open Problems

In this thesis we described approximation algorithms and the construction of coresets for projective clustering. The main open problem in this field is to compute a $(1+\varepsilon)$-approximation to this problem (with exactly k j-flats) in linear or near-linear time in n where $k, j, d > 1$ are constants. This is unknown for either the sum, sum of squares, or maximum of the distances from the input points to the output flats, and this is the case even for $d = 3$. The only exception is the result of Har-Peled [HP04b] for $d = 3$, $j = k = 2$, where the cost is the distance of the farthest point from the two output planes.

In Theorem 3.1 we suggested a (bicriteria) $2^{O(j)}$-approximation for this problem using $\mathrm{poly}(\log n)$ j-flats. A step towards an efficient PTAS for the projective clustering problem would be to reduce these two criteria. Our algorithm uses random sampling and its properties that relate to ε-approximation and ε-nets (see [HW87]). A natural question would be to de-randomize the algorithm using deterministic ε-approximations for the corresponding problems, and to generalize the algorithm to other approximation problems.

In practical applications, one usually assume that there are outliers in the data. That is, for a given $m \geq 1$, we want to minimize the sum of distances (or the other cost functions) from the input points to the output flats, while ignoring the m farthest points from the flats. Currently, results in this direction are known only for the case of points ($j = 0$); see [CKMN01, Che08].

Although it is not clear whether a $(1+\varepsilon)$-approximation for the projective clustering problem with $k, j > 1$ can be computed in near-linear time, it was proved that there is no coreset for this problem. Here, "coreset" means a small (sub-linear) set of points that approximates the sum of distances (or the other cost functions) to *any* query set of k flats. However, it is still an open problem

whether there is a way to represent the set of n input points using $O(n)$ bits so that such a query can be answered in sub-linear time, or to construct a coreset that approximates a restricted class of k j-flats. Lower bounds for this kind of questions are usually obtained using tools from communication and information theory (where no assumptions are made on the "type" of the ouput coreset, but only on its representation length).

In this work we constructed coresets for a static set of n points. Open problems are whether such a coresets can be constructed for a dynamic (under insertion/deletions) or kinetic (moving) set of points. For the case $k = 1$, recent results provide construction of coresets of size only linear in d [FMSW10, DDH$^+$08]. It is still an open problem to compute a coreset for a set of k lines $(j = 1)$ in high dimensional space, or to solve the corresponding optimization problems.

Bibliography

[ABG06] E. Angel, E. Bampis, and L. Gourvès. Approximation algorithms for the bi-criteria weighted MAX-CUT problem. *Discrete Applied Mathematics*, 154(12):1685–1692, 2006.

[AGGR98] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. ACM-SIGMOD Int. Conf. on Management of Data*, volume 27(2) of *SIGMOD Record*, pages 94–105. ACM Press, 1998.

[AHPV04] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. *Journal of the ACM*, 51(4):606–635, 2004.

[AHPV05] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Geometric approximations via coresets. *Combinatorial and Computational Geometry - MSRI Publications*, 52:1–30, 2005.

[AJMP02] P. K. Agarwal, M. Jones, T. M. Murali, and C. M. Procopiuc. A Monte Carlo algorithm for fast projective clustering. In *Proc. ACM-SIGMOD Int. Conf. on Management of Data*, pages 418–427, 2002.

[AM04] P. K. Agarwal and N. H. Mustafa. k-means projective clustering. In *Proc. 23rd ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems (PODS)*, pages 155–165, 2004.

[AP00] P. K. Agarwal and C. M. Procopiuc. Approximation algorithms for projective clustering. In *Proc. 11th Annu. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 538–547, 2000.

[APV02] P. K. Agarwal, C. M. Procopiuc, and K. R. Varadarajan. Approximation algorithms for k-line center. In *Proc. 10th Annu. European*

Symp. on Algorithms (ESA), volume 2461 of *Lecture Notes in Computer Science*, pages 54–63. Springer, 2002.

[APW+99] C. C. Aggarwal, C. M. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park. Fast algorithms for projected clustering. In *Proc. ACM-SIGMOD Int. Conf. on Management of Data*, pages 61–72, 1999.

[AY00] C. Aggarwal and P. Yu. Finding generalized projected clusters in high dimensional spaces. In *Proc. ACM-SIGMOD Int. Conf. on Management of Data*, pages 544–555, 2000.

[BFLS07] L. S. Buriol, G. Frahling, S. Leonardi, and C. Sohler. Estimating clustering indexes in data streams. In *Proc. 15th Annu. European Symp. on Algorithms (ESA)*, volume 4698 of *Lecture Notes in Computer Science*, pages 618–632. Springer, 2007.

[BHPI02] M. Badoiu, S. Har-Peled, and P. Indyk. Approximate clustering via coresets. In *Proc. 34th Annu. ACM Symp. on Theory of Computing (STOC)*, pages 250–257, 2002.

[BMS99] V. Boltyanski, H. Martini, and V. Soltan. *Geometric Methods and Optimization Problems*. Kluwer Academic Publishers, The Netherlands, 1999.

[Bre96] T. M. Breuel. Finding lines under bounded error. *Pattern Recognition*, 29(01):167–178, 1996.

[CC01] R. M. Cesar and L. F. Costa. *Shape Analysis and Classification*. CRC Press, Boca Raton, 2001.

[CEF+05] A. Czumaj, F. Ergun, L. Fortnow, A. Magen, I. Newman, R. Rubinfeld, and C. Sohler. Approximating the weight of the Euclidean minimum spanning tree in sublinear time. *SIAM Journal on Computing*, 35:91–109, 2005.

[Cha04] T. M. Chan. Faster coreset constructions and data stream algorithms in fixed dimensions. In *Proc. 20th Annu. ACM Symp. on Computational Geometry (SoCG)*, pages 152–159, 2004.

[Che06] K. Chen. On k-median clustering in high dimensions. In *Proc. 17th Annu. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 1177–1185, 2006.

[Che08] Ke Chen. A constant factor approximation algorithm for k -median clustering with outliers. In Shang-Hua Teng, editor, *SODA*, pages 826–835. SIAM, 2008.

[CKMN01] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *Proc. 12th Annu. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 642–651, 2001.

[Cla05] K. L. Clarkson. Subgradient and sampling algorithms for l_1-regression. In *Proc. 16th Annu. ACM-SIAM Symp. on Discrete algorithms (SODA)*, pages 257–266, 2005.

[COP03] M. Charikar, L. O'Callaghan, and R. Panigrahy. Better streaming algorithms for clustering problems. In *Proc. 35th Annu. ACM Symp. on Theory of Computing (STOC)*, pages 30–39, 2003.

[CS07] A. Czumaj and C. Sohler. Sublinear-time approximation algorithms for clustering via random sampling. *Random Struct. Algorithms (RSA)*, 30(1-2):226–256, 2007.

[DDH+08] A. Dasgupta, P. Drineas, B. Harb, R. Kumar, and M. W. Mahoney. Sampling algorithms and coresets for ℓ_p-regression. In *Proc. 19th Annu. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 932–941, 2008.

[Des07] A. J. Deshpande. *Sampling Based Algorithms for Dimension Reduction,*. Ph.D. thesis, Department of Mathematics, Massachusetts Institute of Technology, 2007.

[Dey98] T. K. Dey. Improved bounds for planar k-sets and related problems. *Discrete Comput. Geom.*, 19(3):373–382, 1998.

[DH02] Z. Drezner and H. W. Hamacher, editors. *Facility Location: Applications and Theory*. Springer Verlag, New York, 2002.

[DHS00] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, New-York, 2000.

[DM98] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. John Wiley and Sons, San Diego, 1998.

[Don08] P. Dong. Generating and updating multiplicatively weighted Voronoi diagrams for point, line and polygon features in GIS. *Computers & Geosciences*, 34(4):411–421, 2008.

[DRVW06] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via volume sampling. In *Proc. 17th Annu. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 1117–1126, 2006.

[DV06] A. Deshpande and S. Vempala. Adaptive sampling and fast low-rank matrix approximation. *Proc. 10th Int. Workshop on Randomization and Computation (RANDOM)*, pages 292–303, 2006.

[DV07] A. Deshpande and K. R. Varadarajan. Sampling-based dimension reduction for subspace approximation. In *Proc. 39th Annu. ACM Symp. on Theory of Computing (STOC)*, pages 641–650, 2007.

[Fel04] D. Feldman. Algorithms for fitting points by k lines. M.Sc. Thesis, School of Copmputer Science, Tel-Aviv university. 2004.

[FFKN09] D. Feldman, A. Fiat, H. Kaplan, and K. Nissim. Private coresets. In *Proc. 41st Annu. ACM Symp. on Theory of Computing (STOC)*, pages 361–370, 2009.

[FFS06] D. Feldman, A. Fiat, and M. Sharir. Coresets for weighted facilities and their applications. In *Proc. 47th IEEE Annu. Symp. on Foundations of Computer Science (FOCS)*, pages 315–324, 2006.

[FFSS07] D. Feldman, A. Fiat, D. Segev, and M. Sharir. Bi-criteria linear-time approximations for generalized k-mean/median/center. In *Proc. 23rd ACM Symp. on Computational Geometry (SOCG)*, pages 19–26, 2007.

[FIS08] G. Frahling, P. Indyk, and C. Sohler. Sampling in dynamic data streams and applications. *Int. J. Comput. Geometry Appl.*, 18(1/2):3–28, 2008.

[FL08] D. Feldman and M. Langberg. On approximating subspaces by subspaces. *Manuscript*, 2008.

[FMS07] D. Feldman, M. Monemizadeh, and C. Sohler. A PTAS for k-means clustering based on weak coresets. In *Proc. 23rd ACM Symp. on Computational Geometry (SoCG)*, pages 11–18, 2007.

[FMSW10] D. Feldman, M. Monemizadeh, C. Sohler, and D. P. Woodruff. Coresets and sketches for high dimensional subspace approximation problems. In *Proc. 21th Annu. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2010.

[FS05] G. Frahling and C. Sohler. Coresets in dynamic geometric data streams. In *Proc. 37th Annu. ACM Symp. on Theory of Computing (STOC)*, pages 209–217, 2005.

[Hau92] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, 1992.

[HOK01] A. Hyvärinen, E. Oja, and J. Karhunen. *Independent Component Analysis*. Wiley-Interscience, New-York, 2001.

[HP04a] S. Har-Peled. Clustering motion. *Discrete Comput. Geom.*, 31(4):545–565, 2004.

[HP04b] S. Har-Peled. No coreset, no cry. In *Proc. 24th Int. Conf. Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 3328 of *Lecture Notes in Computer Science*, pages 324–335. Springer, 2004.

[HP06a] S. Har-Peled. Coresets for discrete integration and clustering. In *Proc. 26th Int. Conf. Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 4337 of *Lecture Notes in Computer Science*, pages 33–44. Springer, 2006.

[HP06b] S. Har-Peled. Low rank matrix approximation in linear time. *Manuscript*, 2006.

[HPK07] S. Har-Peled and A. Kushal. Smaller coresets for k-median and k-means clustering. *Discrete Comput. Geom.*, 37(1):3–19, 2007.

[HPM04] S. Har-Peled and S. Mazumdar. On coresets for k-means and k-median clustering. In *Proc. 36th Annu. ACM Symp. on Theory of Computing (STOC)*, pages 291–300, 2004.

[HPV02] S. Har-Peled and K. R. Varadarajan. Projective clustering in high dimensions using coresets. In *Proc. 18th ACM Symp. on Computational Geometry (SoCG)*, pages 312–318, 2002.

[HPV04] S. Har-Peled and K. R. Varadarajan. High-dimensional shape fitting in linear time. *Discrete Comput. Geom.*, 32(2):269–288, 2004.

[HPW04] S. Har-Peled and Y. Wang. Shape fitting with outliers. *SIAM Journal on Computing*, 33(2):269–285, 2004.

[HW87] David Haussler and Emo Welzl. ε-nets and simplex range queries. *Discrete Computational Geometry*, 2:127–151, 1987.

[Ind99] P. Indyk. Sublinear time algorithms for metric space problems. In *Proc. 31st Annu. ACM Symp. on Theory of Computing (STOC)*, pages 428–434, 1999.

[JK95] J. W. Jaromczyk and M. Kowaluk. The two-line center problem from a polar view: a new algorithm and data structure. In *Proc. 4th Int. Workshop on Algorithms and Data Structures (WADS'95)*, volume 955 of *Lecture Notes in Computer Science*, pages 13–25. Springer, 1995.

[Jol86] I. T. Jolliffe. *Principal Component Analysis*. Springer Verlag, New York, 1986.

[KA04] S. Kolenikov and G. Angeles. The use of discrete data in PCA: Theory, simulations, and applications to socioeconomic indices. Technical report, Carolina Population Center, University of North Carolina, Chapel Hill, 2004.

[KM93] N. M. Korneenko and H. Martini. Hyperplane approximation and related topics. In *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms and Combinatorics*, chapter 6, pages 135–161. Springer-Verlag, Heidelberg, 1993.

[KS90] M. Kirby and L. Sirovich. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):103–108, 1990.

[LS08] C. Lammersen and C. Sohler. Facility location in dynamic geometric data streams. In *Proc. 16th Annu. European Symp. on Algorithms (ESA)*, pages 660–671, 2008.

[MI94] T. Masuda and H. Ishii. Two machine open shop scheduling problem with bi-criteria. *DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 52, 1994.

[MS98] H. Martini and A. Schöbel. Median hyperplanes in normed spaces - a survey. *Discrete Applied Mathematics*, 89(1-3):181–195, 1998.

[MT83] N. Meggido and A. Tamir. Finding least-distance lines. *SIAM J. on Algebric and Discrete Methods*, 4:207–211, 1983.

[Mul93] K. Mulmuley. *Computational Geometry, an Introduction through Randomized Algorithms*. Prentice Hall, Englewood Cliffs, 1993.

[Pan04] R. Panigrahy. Minimum enclosing polytope in high dimensions. *Computing Research Repository, cs.CG/0407020*, 2004.

[Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, 2:559–572, 1901.

[Ric86] J. A. Richards. *Remote Sensing Digital Image Analysis: an Introduction*. Springer-Verlag, Berlin, 1986.

[SA95] M. Sharir and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, New York, 1995.

[Sar06] T. Sarlós. Improved approximation algorithms for large matrices via random projections. In *Proc. 47th IEEE Annu. Symp. on Foundations of Computer Science (FOCS)*, pages 143–152, 2006.

[Sch99] A. Schöbel. *Locating Lines and Hyperplanes: Theory and Algorithms*. Springer-Verlag, New-York, 1999.

[SV07] N. D. Shyamalkumar and K. R. Varadarajan. Efficient subspace approximation algorithms. In *Proc. 18th Annu. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 532–540, 2007.

[TT96] C. W. Tao and J. S. Taur. Medical image compression using principal component anlyasis. In *Proc. 6th IEEE Int. Conf. on Image Processing (ICIP'96)*, volume 2 (1), pages 903–906, 1996.

[Yan08] J. Yan. An improved lower bound for a bi-criteria scheduling problem. *Oper. Res. Lett*, 36:57–60, 2008.

[YKII88] P. Yamamoto, K. Kato, K. Imai, and H. Imai. Algorithms for vertical and orthogonal ℓ_1 linear approximation of points. In *Proc. 4th Annu. Symp. on Computational Geometry (SoCG)*, pages 352–361, 1988.