Linear-Time Approximation Schemes for Clustering Problems in any Dimensions[*]

Amit Kumar
Dept of Comp Sc & Engg
Indian Institute of Technology
New Delhi-110016, India
amitk@cse.iitd.ernet.in

Yogish Sabharwal
IBM Research - India
Plot 4, Block C,
Vasant Kunj Inst. Area
New Delhi-110070, India
ysabharwal@in.ibm.com

Sandeep Sen
Dept of Comp Sc & Engg
Indian Institute of Technology
New Delhi-110016, India
ssen@cse.iitd.ernet.in

September 23, 2009

Abstract

We present a general approach for designing approximation algorithms for a fundamental class of geometric clustering problems in arbitrary dimensions. More specifically, our approach leads to simple randomized algorithms for the k-means, k-median and discrete k-means problems that yield $(1+\varepsilon)$ approximations with probability $\geq 1/2$ and running times of $O(2^{(k/\varepsilon)^{O(1)}}dn)$. These are the first algorithms for these problems whose running times are linear in the size of the input (nd for n points in d dimensions) assuming k and ε are fixed. Our method is general enough to be applicable to clustering problems satisfying certain simple properties and is likely to have further applications.

1 Introduction

The problem of clustering a group of data items into similar groups is one of the most widely studied problems in computer science. Clustering has applications in a variety of areas, for example, data mining, information retrieval, image processing, and web search ([5, 9, 22, 11]). Given the wide range of applications, many different definitions of clustering exist in the literature ([10, 4]). Most of these definitions begin by defining a notion of distance (similarity) between two data items and then try to form clusters so that data items with small distance between them get clustered together.

Often, clustering problems arise in a geometric setting, i.e., the data items are points in a high dimensional Euclidean space. In such settings, it is natural to define the distance between two points as the Euclidean distance between them. Two of the most popular definitions of clustering are the *k-means clustering problem* and the *k-median clustering problem*. Given a set of points P, the k-means clustering problems seeks to find a set K of k centers, such that $\sum_{p\in P} d(p,K)^2$ is minimized, whereas the k-median clustering problems seeks to find a set K of k centers, such that $\sum_{p\in P} d(p,K)$ is minimized. Note that the points in K can be arbitrary points in the Euclidean space. Here $d(p,K)$ refers to the distance between p and the closest center in K. We can think of this as each point in P gets assigned to the closest center. The points that get assigned to the same center form a cluster. These problems are NP-hard for even $k=2$ (when dimension is not

[*] Preliminary versions of the results have appeared earlier in IEEE Symposium on Foundations of Computer Science, 2004[17] and International Colloquium on Automata, Languages and Programming, 2005[18].

fixed) [7]. Interestingly, the center in the optimal solution to the 1-mean problem is the same as the center of mass of the points. However, in the case of the 1-median problem, also known as the Fermat-Weber problem, no such closed form is known. We show that despite the lack of such a closed form, we can obtain an approximation to the optimal 1-median in $O(1)$ time (independent of the number of points). There are many useful variations to these clustering problems, for example, in the discrete versions of these problems, the centers that we seek should belong to the input set of points.

1.1 Related work

A lot of research has been devoted to solving these problems exactly (see [14] and the references therein). Even the best known algorithms for the k-median and the k-means problem take at least $\Omega(n^d)$ time. Recently, more attention has been devoted to finding $(1+\varepsilon)$-approximation algorithm for these problems, where ε can be an arbitrarily small constant. This has resulted in algorithms with substantially improved running times. Further, if we look at the applications of these problems, they often involve mapping subjective features to points in the Euclidean space. Since there is an error inherent in this mapping, finding a $(1+\varepsilon)$-approximate solution is within acceptable limits for the actual applications.

The fastest exact algorithm for the k-means clustering problem was proposed by Inaba et al. [14]. They observed that the number of Voronoi partitions of k points in $\Re^d$ is $O(n^{kd})$ and so the optimal k-means clustering could be determined exactly in time $O(n^{kd+1})$. They also proposed a randomized $(1+\varepsilon)$-approximation algorithm for the 2-means clustering problem with running time $O(n/\varepsilon^d)$. Matoušek [19] proposed a deterministic $(1+\varepsilon)$-approximation algorithm for the k-means problem with running time $O(n\varepsilon^{-2k^2d}\log^k n)$.

By generalizing the technique of Arora [1], Arora et al. [2] presented a $O(n^{O(1/\varepsilon)+1})$ time $(1+\varepsilon)$-approximation algorithm for the k-median problem where points lie in the plane. This was significantly improved by Kolliopoulos et al. [16] who proposed an algorithm with a running time of $O(\varrho n \log n \log k)$ for the discrete version of the problem, where the medians must belong to the input set and $\varrho = exp[O((1+\log 1/\varepsilon)/\varepsilon)^{d-1}]$.

Recently, Badoiu et al. [3] proposed a $(1+\varepsilon)$-approximation algorithm for k-median clustering with a running time of $O(2^{(k/\varepsilon)^{O(1)}} d^{O(1)} n \log^{O(k)} n)$. Their algorithm can be extended to k-means with some modifications. de la Vega et al. [8] proposed a $(1+\varepsilon)$-approximation algorithm for the k-means problem which works well for points in high dimensions. The running time of this algorithm is $O(g(k,\varepsilon) n \log^k n)$ where $g(k,\varepsilon) = exp[(k^3/\varepsilon^8)(\ln(k/\varepsilon)\ln k]$.

Recently, Har-Peled and Mazumdar [13] proposed $(1+\varepsilon)$-approximation algorithms for the k-median, discrete k-median and the k-means clustering in low dimensions. They obtained a running time of $O(n + \varrho k^{O(1)} \log^{O(1)} n)$ for the k-median problem, $O(n + \varrho k^{O(1)} \log^{O(1)} n)$ for the discrete k-median problem and $O(n + k^{k+2} \varepsilon^{-(2d+1)k} \log^{k+1} n \log^k \frac{1}{\varepsilon})$ for the k-means problem. For approximating the 1-median of a set of points, Indyk [15] proposed an algorithm that finds a $(1+\varepsilon)$ approximate 1-median in time $O(n/\varepsilon^2)$ with constant probability.

Table 1.1 summarizes the recent results for the problems, in the context of $(1+\varepsilon)$-approximation algorithms. Some of these algorithms are randomized with the expected ruining time holding good for any input.

1.2 Our results and techniques

The general algorithm we present solves a large class of clustering problems satisfying a set of conditions (cf. Section 3). We show that the k-means problem, k-median problem and the discrete

Problem	Result	Reference
1-median	$O(n/\varepsilon^2)$	Indyk [15]
k-median	$O(n^{O(1/\varepsilon)+1})$ for $d = 2$	Arora [1]
	$O(\varrho n \log n \log k)$ (discrete only)	Kolliopoulos et al. [16]
	$O(2^{(k/\varepsilon)^{O(1)}} d^{O(1)} n \log^k n)$	Badoiu et al. [3]
	$O(n + \varrho k^{O(1)} \log^{O(1)} n)$ (discrete also)	Har-Peled et al. [13]
	where $\varrho = exp[O((1 + \log 1/\varepsilon)/\varepsilon)^{d-1}]$	
k-means	$O(n\varepsilon^{-2k^2 d} \log^k n)$	Matoušek [19]
	$O(g(k,\varepsilon) n \log^k n)$	de la Vega et al. [8]
	$g(k,\varepsilon) = exp[(k^3/\varepsilon^8)(\ln(k/\varepsilon)\ln k]$	
	$O(n + k^{k+2}\varepsilon^{-(2d+1)k} \log^{k+1} n \log^k \frac{1}{\varepsilon})$ (discrete also)	Har-Peled et al. [13]

Figure 1: Summary of previous results on k-means and k-median clustering.

k-means problem, all satisfy the required conditions and therefore belong to this class of clustering problems. One important condition that the clustering problems must satisfy is the existence of an algorithm to generate a candidate set of points such that at least one of these points is a close approximation to the optimal center for $k = 1$ (one cluster). Further, the running time of this algorithm as well as the size of this candidate set should be independent of n. Based on such a subroutine, we show how to approximate all the centers in the optimal solution in an iterative manner.

The running times of $O(2^{(k/\varepsilon)^{O(1)}} nd)$ of our algorithms are better than the previously known algorithms for these problems, specially when d is very large. In fact, these are the first algorithms for the k-means, k-median and the discrete k-means clustering problems that have running time linear in the size of the input for fixed k and ε. The algorithms in this paper have the additional advantage of simplicity as the only technique involved is random sampling. Our method is based on using random sampling to identify a small set of candidate centers. In contrast, an alternate strategy [13] involves identifying significantly small sets of points, called *coresets*, such that solving the clustering problem on the coresets yields a solution for the original set of points. In a subsequent work [6], further improvements were obtained using a clever combination of the two techniques (cf. Section 8).

The main drawback of our algorithm is that the running time has exponential dependence on $\left(\frac{k}{\varepsilon}\right)^{O(1)}$. We would however like to note that Guruswamy and Indyk [12] showed that it is NP-hard to obtain a PTAS for the k-median problem for arbitrary k and $d \geq \Omega(\log n)$. Since we avoid exponential dependence on d, this implies that the exponential dependence on k is inherent. An algorithm that avoids exponential dependence on k, like Arora et al.[2] has doubly exponential dependence on d which is arguably worse in most situations.

We also present a randomized $(1+\varepsilon)$-approximation algorithm for the 1-median problem which runs in time $O(2^{1/\varepsilon^{O(1)}} d)$, assuming that the points are stored in a suitable data structure such as an array, where a point can be randomly sampled in constant time. All our algorithms yield the desired result with constant probability (which can be made as close to 1 as we wish by a constant number of repetitions).

The remaining paper is organized as follows. In Section 2, we define clustering problems. In Section 3 we present a simplified algorithm for the 2-means clustering problem. In Section 4, we describe a general approach for solving clustering problems efficiently. In subsequent sections we give applications of the general method by showing that this class of problems includes the

k-means, k-median and discrete k-means problems. In Section 5.3, we also describe an efficient approximation algorithm for the 1-median problem. In Section 7, we extend our algorithms for efficiently handling weighted point sets. We conclude by stating some open problems and some interesting developments subsequent to the publication of earlier versions of this work in Section 8.

2 Clustering Problems

In this section, we give a general definition of clustering problems.

We shall define a clustering problem by two parameters – an integer k and a real-valued cost function $f(Q, x)$, where Q is a set of points, and x is a point in an Euclidean space. We shall denote this clustering problem as $\mathcal{C}(f, k)$. The input to $\mathcal{C}(f, k)$ is a set of points in an Euclidean space.

Given an instance P of n points, $\mathcal{C}(f, k)$ seeks to partition them into k sets, which we shall denote as *clusters*. Let these clusters be $C_1, \ldots, C_k$. A solution also finds k points, which we call *centers*, $c_1, \ldots, c_k$. We shall say that c_i is the center of cluster C_i (or the points in C_i are assigned to c_i). The objective of the problem is to minimize the quantity $\sum_{i=1}^{k} f(C_i, c_i)$.

This is a fairly general definition. Let us see some important special cases.

- k-median : $f_1(Q, x) = \sum_{q \in Q} d(q, x)$.
- k-means : $f_2(Q, x) = \sum_{q \in Q} d(q, x)^2$.

We can also encompass the discrete versions of these problems, i.e., cases where the centers have to be one of the points in P. In such problems, we can make $f(Q, x)$ unbounded if $x \notin Q$.

As stated earlier, we shall assume that we are given a constant $\varepsilon > 0$, and we are interested in finding $(1 + \varepsilon)$-approximation algorithms for these clustering problems.

We now give some definitions. Let us fix a clustering problem $\mathcal{C}(f, k)$. Although we should parametrized all our definitions by f, we avoid this because the clustering problem will be clear from the context.

Definition 2.1. *Given a point set P, let $\mathtt{OPT}_k(P)$ be the cost of the optimal solution to the clustering problem $\mathcal{C}(f, k)$ on input P.*

Definition 2.2. *Given a set of points P and a set of k points C, let $\mathtt{OPT}_k(P, C)$ be the cost of the optimal solution to $\mathcal{C}(f, k)$ on P when the set of centers is C.*

3 Algorithm for 2-means Clustering

In this section we describe the algorithm for 2-means clustering. The 2-means clustering algorithm contains many of the ideas inherent in the more general algorithm. This makes it easier to understand the more general algorithm described in the next section.

Consider an instance of the 2-means problem where we are given a set P of n points in $\Re^d$. We seek to find $\mathcal{C}(f_2, 2)$ where f_2 corresponds to the k-means cost function as defined in Section 2.

We first look at some properties of the 1-means problem.

Definition 3.1. *For a set of points P, define the* centroid, *$c(P)$, of P as the point $\frac{\sum_{p \in P} p}{|P|}$.*

Claim 3.1. *For any point $x \in \Re^d$,*

$$f_2(P, x) = f_2(P, c(P)) + |P| \cdot d(c(P), x)^2. \tag{1}$$

Proof.

$$\begin{aligned} f_2(P,x) &= \sum_{p\in P} ||p-x||^2 \\ &= \sum_{p\in P} ||p-c(P)+c(P)-x||^2 \\ &= \sum_{p\in P} ||p-c(P)||^2 + \sum_{p\in P} ||c(P)-x||^2 \\ &= f_2(P,c(P)) + |P|\cdot d(c(P),x)^2 \end{aligned}$$

where the second last equality follows from the fact that $\sum_{p\in P} ||p-c(P)|| = 0$. □

From this we can make the following observation.

Fact 3.2. *Any optimal solution to the 1-means problem with respect to an input point set P chooses $c(P)$ as the center.*

We can deduce an important property of any optimal solution to the 2-means clustering problem. Suppose we are given an optimal solution to the 2-means clustering problem with respect to the input P. Let $C = \{c_1, c_2\}$ be the set of centers constructed by this solution. C produces a partitioning of the point set P into 2 clusters, namely, P_1, P_2. P_i is the set of points for which the closest point in C is c_i. In other words, the clusters correspond to the points in the Voronoi regions in $\Re^d$ with respect to C. Now, Fact 3.2 implies that c_i must be the centroid of P_i for $i = 1, 2$.

Inaba et. al. [14] showed that the centroid of a small random sample of points in P can be a good approximation to $c(P)$.

Lemma 3.3. [14] *Let T be a set obtained by independently sampling m points uniformly at random from a point set P. Then, for any $\delta > 0$,*

$$f_2(S, c(T)) \quad < \quad \left(1 + \frac{1}{\delta m}\right) \mathtt{OPT}_1(P)$$

holds with probability at least $1-\delta$.

Therefore, if we choose m as $\frac{2}{\varepsilon}$, then with probability at least 1/2, we get a $(1+\varepsilon)$-approximation to $\mathtt{OPT}_1(P)$ by taking the center as the centroid of T. Thus, a constant size sample can quickly yield a good approximation to the optimal 1-means solution.

Suppose P' is a subset of P and we want to get a good approximation to the optimal 1-mean for the point set P'. Following Lemma 3.3, we would like to sample from P'. But the problem is that P' is not explicitly given to us. The following lemma states that if the size of P' is close to that of P, then we can sample a slightly larger set of points from P and hopefully this sample would contain enough random samples from P'. Let us define things more formally first. Let P be a set of points and P' be a subset of P such that $|P'| \geq \theta|P|$, where θ is a constant between 0 and 1. Suppose we take a sample S of size $\frac{4}{\theta\varepsilon}$ from P. Now we consider all possible subsets of size $\frac{2}{\varepsilon}$ of S. For each of these subsets S', we compute its centroid $c(S')$, and consider this as a potential center for the 1-means problem instance on P'. In other words, we consider $f_2(P', c(S'))$ for all such subsets S'. The following lemma shows that one of these subsets must give a close enough approximation to the optimal 1-means solution for P'.

Lemma 3.4. *The following event happens with constant probability*

$$\min_{S':S'\subset S, |S'|=\frac{2}{\varepsilon}} f_2(P', c(S')) \leq (1+\varepsilon)\mathtt{OPT}_1(P')$$

```
Algorithm 2-clustering($(P, \varepsilon)$)
Inputs : $P$: Point set
         $\varepsilon$: approximation factor
Output : The optimal 2-means clustering of the points in $P$.

1. Let $\alpha = \varepsilon/64$
2. Sample a set $S$ of size $O(1/\alpha)$ from $P$
3. For each subset $S'$ of $S$ of size $2/\alpha$ do
   (a) Compute the mean, $c'_1$, of $S'$. This is a candidate approximate center for $C_1$.
   (b) Consider the points of $P$ in ascending order of distance from $c'_1$
   (c) For $i = 1$ to $\log(|P|) + 1$ do
       i.   Let $Q'_i$ be the last $|P|/2^{i-1}$ points (farthest from $c'_1$) in this sequence
            [ For iterations $i \geq 2$, this only requires scanning of points in $Q'_{i-1}$ ]
       ii.  Assign the points in $P - Q'_i$ to $c'_1$. Compute $f_2(P - Q'_i, c'_1)$.
            [ For iterations $i \geq 2$, this can be computed from $f_2(P - Q'_{i-1}, c'_1)$
                since $f_2(P - Q'_{i-1}, c'_1) = f_2(P - Q'_{i-1}, c'_1) + f_2(Q'_{i-1} - Q'_i, c'_1)$ ]
       iv.  Sample a set $\tilde{S}$ of size $O(1/\alpha^2)$ from $Q'_i$
       v.   For each subset $\tilde{S}'$ of $\tilde{S}$ of size $2/\alpha$ do
            - Compute the mean, $c'_2$, of $\tilde{S}'$. This is a candidate approximate center for $C_2$.
            - Assign the points in $Q'_i$ to the nearest centers in $\{c'_1, c'_2\}$
            - Compute the clustering cost for this choice of $\{c'_1, c'_2\}$ as
                $Sum = f_2(P - Q'_i, c'_1) + \mathtt{OPT}_2(Q'_i, \{c'_1, c'_2\})$.
4. Compute the clustering cost to the centroid $c(P)$ of $P$, i.e., $f_2(P, c(P))$.
5. Return the clustering which has minimum cost from those obtained in steps 4 and 5.
```

Figure 2: The 2-means clustering algorithm

Proof. With constant probability, S contains at least $\frac{2}{\varepsilon}$ points from P'. The rest follows from Lemma 3.3. □

The 2-means clustering algorithm is presented in Figure 2. In the following proofs, we use the standard notation $\mathcal{B}(p, r)$ to denote the ball of radius r around a point p.

Theorem 3.5. *Given a point set P of size n in $\Re^d$, there exists an algorithm which produces a $(1+\varepsilon)$-approximation to the optimal 2-means solution on the point set P with constant probability. Further, this algorithm runs in time $O(2^{(1/\varepsilon)^{O(1)}} dn)$.*

Proof. Let $\alpha = \varepsilon/64$. We can assume that $\mathtt{OPT}_2(P) > (1+\varepsilon/2)\mathtt{OPT}_1(P)$ otherwise the solution to the 1-mean problem for P obtained by computing the centroid of P in $O(nd)$ time has cost at most $(1+\varepsilon/2)\mathtt{OPT}_2(P)$.

Consider an optimal 2-means solution for P. Let c_1 and c_2 be the two centers in this solution. Let P_1 be the points which are closer to c_1 than c_2 and P_2 be the points closer to c_2 than c_1. So c_1 is the centroid of P_1 and c_2 that of P_2. Without loss of generality, assume that $|P_1| \geq |P_2|$.

Since $|P_1| \geq |P|/2$, Lemma 3.4 implies that if we sample a set S of size $O\left(\frac{1}{\alpha}\right)$ from P and look at the set of centroids of all subsets of S of size $\frac{2}{\alpha}$, then at least one of these centroids, call it c'_1 has the property that $f_2(P_1, c'_1) \leq (1+\alpha) f_2(P_1, c_1)$. Since our algorithm is going to cycle through all such subsets of S, we can assume that we have found such a point c'_1. Our remaining proof is based on Lemma 3.6 and Claims 3.7 and 3.8.

Let the distance between c_1 and c_2 be t, i.e., $d(c_1, c_2) = t$.

Lemma 3.6. $d(c_1, c_1') \leq t/4$.

Proof. Suppose $d(c_1, c_1') > t/4$. Equation (1) implies that

$$f_2(P_1, c_1') - f_2(P_1, c_1) = |P_1| d(c_1, c_1')^2 \geq \frac{t^2|P_1|}{16}.$$

But we also know that left hand side is at most $\alpha f_2(P_1, c_1)$. Thus we get $t^2|P_1| \leq 16\alpha f_2(P_1, c_1)$.

Applying Equation (1) once again, we see that

$$f_2(P_1, c_2) = f_2(P_1, c_1) + t^2|P_1| \leq (1 + 16\alpha) f_2(P_1, c_1).$$

Therefore, $f_2(P, c_2) \leq (1+16\alpha) f_2(P_1, c_1) + f_2(P_2, c_2) \leq (1+16\alpha)\mathtt{OPT}_2(P)$. This contradicts the fact that $\mathtt{OPT}_1(P) > (1 + \varepsilon/2)\mathtt{OPT}_2(P)$.

This completes the proof of Lemma 3.6. □

Now consider the ball $\mathcal{B}(c_1', t/4)$. The previous lemma implies that this ball is contained in the ball $\mathcal{B}(c_1, t/2)$ of radius $t/2$ centered at c_1. So $\mathcal{B}(c_1', t/4)$ contains only points in P_1. Since we are looking for the point c_2, we can delete the points in this ball and hope that the resulting point set has a good fraction of points from P_2.

This is what we prove next. Let P_1' denote the point set $P_1 - \mathcal{B}(c_1', t/4)$. Let P' denote $P_1' \cup P_2$. As we noted above P_2 is a subset of P'.

Claim 3.7. $|P_2| \geq \alpha |P_1'|$

Proof. Suppose not, i.e., $|P_2| \leq \alpha |P_1'|$. Notice that

$$f_2(P_1, c_1') \geq f_2(P_1', c_1') \geq \frac{t^2|P_1'|}{16}.$$

Since $f_2(P_1, c_1') \leq (1 + \alpha) f_2(P_1, c_1)$, it follows that

$$t^2|P_1'| \leq 16(1 + \alpha) f_2(P_1, c_1) \tag{2}$$

So,

$$\begin{aligned} f_2(P, c_1) &= f_2(P_1, c_1) + f_2(P_2, c_1) \\ &= f_2(P_1, c_1) + f_2(P_2, c_2) + t^2|P_2| \\ &\leq f_2(P_1, c_1) + f_2(P_2, c_2) + t^2\alpha|P_1'| \\ &\leq f_2(P_1, c_1) + f_2(P_2, c_2) \\ &\quad +16\alpha(1 + \alpha) f_2(P_1, c_1) \\ &\leq (1 + 32\alpha) f_2(P_1, c_1) + f_2(P_2, c_2) \\ &\leq (1 + 32\alpha)\mathtt{OPT}_2(P), \end{aligned}$$

where the second equation follows from Equation (1), while the second inequality follows from Inequality (2) and the fact that $|P_2| \leq \alpha|P_1'|$. But this contradicts the fact that $\mathtt{OPT}_1(P) > (1 + \varepsilon/2)\mathtt{OPT}_2(P)$.

This completes the proof of Claim 3.7. □

The above claim combined with Lemma 3.4 implies that if we sample $O\left(\frac{1}{\alpha^2}\right)$ points from P', and consider the centroids of all subsets of size $\frac{2}{\alpha}$ in this sample, then with constant probability we shall get a point c'_2 for which $f_2(P_2, c'_2) \leq (1+\alpha)f_2(P_2, c_2)$. Thus, we get the centers c'_1 and c'_2 which satisfy the requirements of our lemma.

The only problem is that we do not know the value of the parameter t. We will somehow need to guess this value and yet maintain the fact that our algorithm takes only linear amount of time.

We can assume that we have found c'_1 (this does not require any assumption on t). Now we need to sample from P' (recall that P' is the set of points obtained by removing the points in P distant at most $t/4$ from c'_1). Suppose we know the parameter i such that $\frac{n}{2^i} \leq |P'| \leq \frac{n}{2^{i-1}}$.

Consider the points of P in descending order of distance from c'_1. Let Q'_i be the first $\frac{n}{2^{i-1}}$ points in this sequence. Notice that P' is a subset of Q'_i and $|P'| \geq |Q'_i|/2$. Also we can find Q'_i in linear time (because we can locate the point at position $\frac{n}{2^{i-1}}$ in linear time). Since $|P_2| \geq \alpha|P'_1|$, we see that $|P_2| \geq \alpha|Q'_i|/4$. Thus, Lemma 3.3 implies that it is enough to sample $O\left(\frac{1}{\alpha^2}\right)$ points from Q'_i to locate c'_2 (with constant probability of course).

But the problem with this scheme is that we do not know the value i. One option is try all possible values of i, which will imply a running time of $O(n \log n)$ (treating the terms involving α and d as constant). Also note that we cannot use approximate range searching because preprocessing takes $O(n \text{log} n)$ time.

We somehow need to combine the sampling and the idea of guessing the value of i. Our algorithm proceeds as follows. It tries values of i in the order $0, 1, 2, \ldots$. In iteration i, we find the set of points Q'_i. Note that Q'_{i+1} is a subset of Q'_i. In fact Q'_{i+1} is the half of Q'_i which is farther from c'_1. So in iteration $(i+1)$, we can begin from the set of points Q'_i (instead of P'). We can find the candidate point c'_2 by sampling from Q'_{i+1}. Thus we can find Q'_{i+1} in time linear in $|Q'_{i+1}|$ only.

Further in iteration i, we also maintain the sum $f_2(P - Q'_i, c'_1)$. Since $f_2(P - Q'_{i+1}, c'_1) = f_2(P - Q'_i, c'_1) + f_2(Q'_i - Q'_{i+1}, c'_1)$, we can compute $f_2(P - Q'_{i+1}, c'_1)$ in iteration $i+1$ in time linear in Q'_{i+1}. This is needed because when we find a candidate c'_2 in iteration $i+1$, we need to compute the 2-means solution when all points in $P - Q'_i$ are assigned to c'_1 and the points in Q'_i are assigned to the nearer of c'_1 and c'_2. We can do this in time linear in $|Q'_{i+1}|$ if we maintain the quantities $f_2(P - Q'_i, c'_1)$ for all i.

Thus, we see that iteration i takes time linear in $|Q'_i|$. Since $|Q'_i|$'s decrease by a factor of 2, the overall running time for a given value of c'_1 is $O(2^{(1/\alpha)^{O(1)}} dn)$. Since the number of possible candidates for c'_1 is $O(2^{(1/\alpha)^{O(1)}})$, the running time is as stated.

Claim 3.8. *The cost, Δ, reported by the algorithm satisfies* $\mathtt{OPT}_2(P) \leq \Delta \leq (1+\alpha)\mathtt{OPT}_2(P)$ *with constant probability.*

Proof. $\mathtt{OPT}_2(P) \leq \Delta$ is obvious as we are associating each point with one of the 2 centers being reported and accumulating the corresponding cost. Now, consider the case when we have the candidate center set where each center is a $(1+\alpha)$-approximate centroid of its respective cluster. As we are associating each point to the approximate centroid of the corresponding cluster or a center closer than it, it follows that $\Delta \leq (1+\alpha)\mathtt{OPT}_2(P)$. □

This also completes the proof of Theorem 3.5. □

3.1 General Properties of Clustering Problems

Our algorithms will work on any of clustering problems defined in Section 2 provided certain conditions are satisfied. We state these conditions in this section.

Definition 3.2. *We say that a point set P is* (k,α)-irreducible *if* $\mathtt{OPT}_{k-1}(P) \geq (1+\delta\alpha)\mathtt{OPT}_k(P)$, *where δ is a constant determined by the nature of the clustering problem (e.g. k-median, k-means). Otherwise we say that the point set is* (k,α)-reducible.

Reducibility captures the fact that if P is (k,α)-reducible for a small constant α, then the optimal solution for $\mathcal{C}(f,k-1)$ on P is close to that for $\mathcal{C}(f,k)$ on P. So if we are solving the latter problem, it is enough to solve the former one. In fact, when solving the problem $\mathcal{C}(f,k)$ on the point set P, we can assume that P is (k,α)-irreducible, where $\alpha = \frac{\epsilon}{8\delta k}$. Indeed, suppose this is not the case. Let i be the highest integer such that P is (i,α)-irreducible. Then, $\mathtt{OPT}_k(P) \leq (1+\delta\alpha)^{k-i}\mathtt{OPT}_i(P) \leq (1+\varepsilon/4)\mathtt{OPT}_i(P)$. Therefore, if we can get a $(1+\varepsilon/4)$-approximation algorithm for $\mathcal{C}(f,i)$ on input P, then we have a $(1+\varepsilon)$-approximation algorithm for $\mathcal{C}(f,k)$ on P. Thus it is enough to solve instances which are irreducible.

The first property that we want $\mathcal{C}(f,k)$ to satisfy is a fairly obvious one – it is always better to assign a point in P to the nearest center. We state this more formally as follows :

> ***Closeness Property*** **:** Let Q and Q' be two disjoint set of points, and let $q \in Q$. Suppose x and x' are two points such that $d(q,x) > d(q,x')$. Then the cost function f satisfies the following property
>
> $$f(Q,x) + f(Q',x') \geq f(Q-\{q\},x) + f(Q' \cup \{q\},x').$$

This is essentially saying that in order to find a solution, it is enough to find the set of k centers. Once we have found the centers, the actual partitioning of P is just the Voronoi partitioning with respect to these centers. It is easy to see that the k-means problem and the k-median problem (both the continuous and the discrete versions) satisfy this property.

We desire two more properties from $\mathcal{C}(f,k)$. The first property says that if we are solving $\mathcal{C}(f,1)$, then there should be a simple random sampling algorithm. The second property says that suppose we have approximated the first i centers of the optimal solution closely. Then we should be able to easily extract the points in P which get assigned to these centers. We describe these properties in more detail below.

One of the key ingredients of the 2-means clustering algorithm described in the previous section (Lemma 3.3) is the ability to derive an approximate center for a set of points using a small (constant size) random sample from the point set. The generalization of this requirement is formally presented below.

> ***Random Sampling Procedure*** **:** There exists a procedure $\mathcal{A}$ that takes as input a parameter α (a constant), and a set of points $R \in \Re^d$ of size λ_α. $\mathcal{A}$ produces as output, another set of points called $\mathtt{core}(R)$, of constant size, β_α. $\mathcal{A}$ satisfies the condition that if R is a random sample obtained from a set Q, then with constant probability there is at least one point $c \in \mathtt{core}(R)$ such that $\mathtt{OPT}_1(Q,\{c\}) \leq (1+\alpha)\mathtt{OPT}_1(Q)$. Further the time taken by $\mathcal{A}$ to produce $\mathtt{core}(R)$ from R is at most $O(\eta_\alpha \cdot dn)$, where n is the size of Q and η_α is a constant.

As described in the previous section, if we take a random sample R of size $\lambda_\alpha = 2/\alpha$ points from the point set and compute its centroid, $\mathtt{core}(R)$ of size $\beta_\alpha = 1$, then with constant probability, this centroid is a $(1+\alpha)$-approximation to the mean of point set.

In the course of our algorithm, the set Q will not be explicitly known - instead we sample from a superset $P \supseteq Q$. We will sample a slightly larger set of points from P and then we isolate a λ_α subset that consists only of points in Q and supply this to the *Random Sampling Procedure*. Although we are not directly sampling from Q, our sampling/isolation procedure must ensure that all λ_α subsets of Q are equally likely in the same way if we had directly sampled from Q.

```
Procedure Superset Sampling($P, \alpha, \theta$)
Inputs : Point set $P$, constants $\alpha, \theta$ ,$\theta < 1$, approximation ratio $\varepsilon$.
Output : A set of points called $\mathtt{CEN}(P, \alpha, \theta)$, of constant size $O(2^{(\frac{\lambda_\alpha}{\theta})^{O(1)}}\beta_\alpha)$.

1. Choose a sample $S$ of size $O(\frac{4\lambda_\alpha}{\theta})$ from $P$.
2. Consider all possible subsets of size $\lambda_\alpha$ of $S$.
There are $O(2^{(\frac{\lambda_\alpha}{\theta})^{O(1)}})$ of these.
3. For each of the subsets $S'$ do,
    Generate a candidate center set for the 1-center for the clustering problem
    using the Random Sampling Procedure, $\mathcal{A}$ (using $\varepsilon$).
4. Return the union of all these candidate center sets.
```

Figure 3: The Superset Sampling Procedure

Figure 3 describes a procedure to determine a set of candidate approximate centers using the *Random Sampling Procedure* when there are extraneous points. We call this the *Superset Sampling Procedure*.

Remark 3.1. *Note that the only property we require the clustering problem to exhibit is the existence of a Random Sampling Procedure. The Superset Sampling Procedure implicitly uses the Random Sampling Procedure for the clustering problem under consideration.*

The following lemma shows that one of the subsets considered by the *Superset Sampling Procedure* must give a close enough approximation to the optimal 1-center solution for Q.

Lemma 3.9. *(Superset Sampling Lemma) Let* $\mathtt{core}(S')$ *be the center set generated using the* Random Sampling Procedure *on sampled subset* S'*. Then, the following event happens with constant probability*

$$\min_{c' \in \mathtt{core}(S') : S' \subset S, |S'| = \lambda_\alpha} f(Q, c') \le (1+\alpha)\mathtt{OPT}_1(Q).$$

Proof. With constant probability, S contains at least λ_α points from Q, the required sample size. Clearly the set S' of λ_α points is equiprobable amongst all the point sets of the same size in Q (obtained with replacement). The rest follows from the *Random Sampling Procedure* for the clustering problem. □

We will later see that our generalized algorithm yields a running time of $O(2^{(k/\varepsilon)^{O(1)}} nd)$ when $\lambda_\alpha = O((\frac{1}{\alpha})^{O(1)})$, $\beta_\alpha = O(2^{(\frac{1}{\alpha})^{O(1)}})$, $\eta_\alpha = O(2^{(\frac{1}{\alpha})^{O(1)}})$ and $\theta = O(\frac{\alpha^{O(1)}}{k})$.

Another important property of the 2-means clustering algorithm described in the previous section is the ability to obtain a random sample of points from the smaller cluster after carefully removing some points and then performing random sampling. Once we have approximated the center of the larger cluster, we can remove enough of its points from around this center. This leaves us with a point set, such that the smaller cluster forms a constant fraction of this point set.

We now generalize this property. Consider the optimal k-clustering of a point set, P, where i centers are fixed (these correspond to the centers that have already been approximated) and $k - i$ centers are free to be selected from the centers of the optimal solution (these correspond to the centers that are yet to be determined). We refer to the clusters centered at the fixed points as fixed clusters and the clusters centered at the free centers as free clusters (see Figure 4). There exists a value r such that if we construct balls of radius r around the i fixed centers, then these balls contain enough points (S) of the fixed clusters, such that the free clusters form a constant fraction of the remaining points lying outside these balls $(P - S)$. This allows us to obtain a random sample of points from the largest free cluster of the required constant size by taking a constant size random sample from the remaining point set. This generalization is formally stated below.

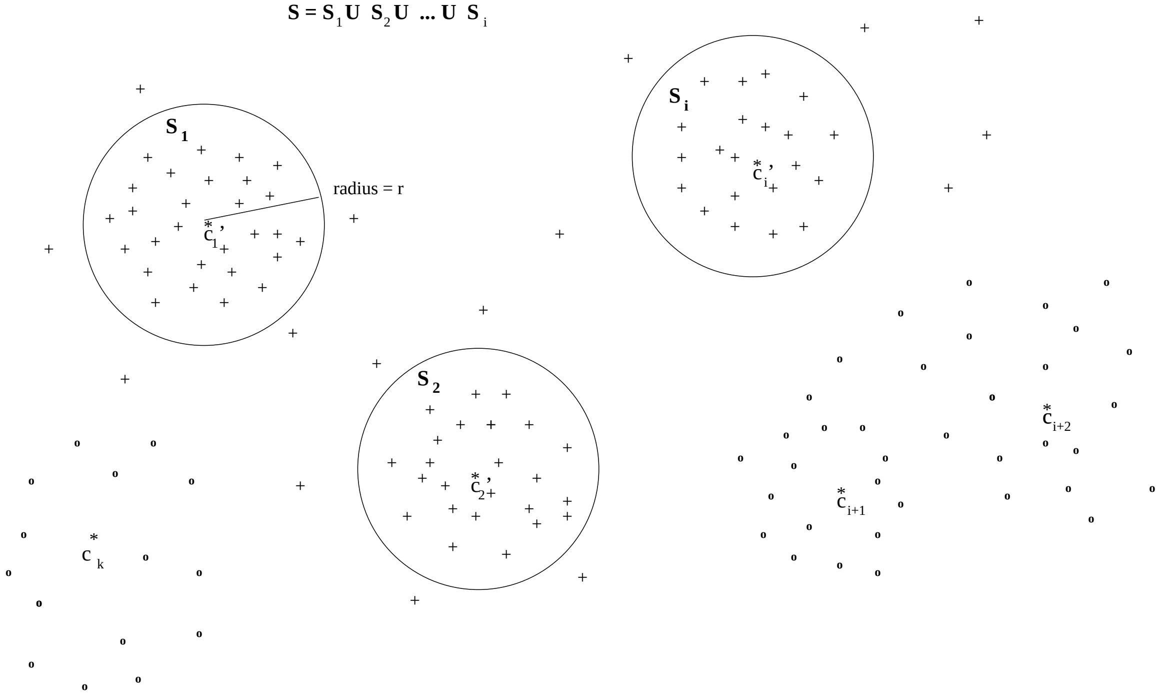

Figure 4: Tightness Property: Fixed clusters with centers $c'_1, c'_2, \ldots, c'_i$ and free clusters with centers $c_{i+1}, \ldots, c_k$

Tightness Property **:** Let P be a set of points which is (k, α)-irreducible for some constant α. Consider an optimal solution to $\mathcal{C}(f, k)$ on P and let $C = \{c_1, \ldots, c_k\}$ be the centers in this solution. Suppose we have a set of i points $C'_i = \{c'_1, \ldots, c'_i\}$, such that $\texttt{OPT}_k(P, \widetilde{C}_i) \leq (1 + \alpha/k)^i \texttt{OPT}_k(P)$, where $\widetilde{C}_i = \{c'_1, \ldots, c'_i, c_{i+1}, \ldots, c_k\}$. Let $P'_1, \ldots, P'_k$ be the partitioning of P if we choose $\widetilde{C}_i$ as the set of centers (in other words this is the Voronoi partitioning of P with respect to $\widetilde{C}_i$). We assume w.l.o.g. that P'_{i+1} is the largest cluster amongst $P'_{i+1}, \ldots, P'_k$. Then there exists a set of points S such that the following conditions hold :

(a) S is contained in $P'_1 \cup \ldots \cup P'_i$.

(b) Let $x \in S, x' \in P - S$. Then, $d(x, \{c'_1, \ldots, c'_i\}) \leq d(x', \{c'_1, \ldots, c'_i\})$.

(c) $P - S$ contains at most $\frac{|P'_{i+1}|}{\alpha^{O(1)}}$ points of $P'_1 \cup \ldots \cup P'_i$.

We show the existence of the tightness property for the k-means clustering problems.

Lemma 3.10. *The* k-means *and* discrete k-means *clustering problems satisfy the tightness property.*

Proof. We essentially need to show the existence of the desired set S, described in the definition above.

Recall that in the discrete version of the problem, $f_2(Q, x) = \infty$ when x is not a point of the input point set. Consider the closest pair of centers between the sets $C' \backslash C'_i$ and C'_i – let these centers be c_l and c'_r respectively. Let $t = d(c_l, c'_r)$. Let S be the set of points $\mathcal{B}(c'_1, t/4) \cup \cdots \cup \mathcal{B}(c'_i, t/4)$.

Clearly, S is contained in $P'_1 \cup \cdots \cup P'_i$. This shows (a). Also, for any $x \in S, x' \in P - S$, $d(x, \{c'_1, \ldots, c'_i\}) \leq d(x', \{c'_1, \ldots, c'_i\})$. This proves (b).

Suppose $P - S$ contains more than $|P_l|/\alpha^2$ points of $P'_1 \cup \cdots \cup P'_i$. In that case, these points are assigned to centers at distance at least $t/4$. It follows that $\mathtt{OPT}_k(P, C')$ is at least $\frac{t^2|P_l|}{16\alpha^2}$. This implies that $t^2|P_l| \leq 16\alpha^2 \mathtt{OPT}_k(P, C')$. Let m_l be the centroid of P_l. Further, let T_l be the average cost paid by P_l when the center is its centroid, i.e., $T_l = \frac{\sum_{p \in P_l} d(p, m_l)^2}{|P_l|}$. Observe that $f_2(P_l, c_l) = |P_l|(T_l + d(c_l, m_l)^2)$. Therefore, if we assign the points in P_l from c_l to c'_r, the increase in cost is

$$\begin{aligned} |P_l| \left(d(c'_r, m_l)^2 - d(c_l, m_l)^2\right) &\leq |P_l| \left((d(c'_r, c_l) + d(c_l, m_l))^2 - d(c_l, m_l)^2\right) \\ &\leq |P_l| \left(t^2 + 2td(c_l, m_l)\right) \end{aligned}$$

We know that the first term above, i.e., $|P_l|t^2$ is at most $16\alpha^2 \mathtt{OPT}_k(P, C')$. We now need to bound the second term only. We consider two cases

- $t \leq \alpha d(c_l, c_m)$: In this case, $|P_l| \cdot 2td(c_l, m_l) \leq 2\alpha d(c_l, m_l)^2 |P_l| \leq 2\alpha f_2(P_l, c_l) \leq 2\alpha \mathtt{OPT}_k(P, C')$.
- $t > \alpha d(c_l, c_m)$: In this case, $|P_l| \cdot 2td(c_l, m_l) \leq \frac{2t^2|P_l|}{\alpha} \leq 32\alpha \mathtt{OPT}_k(P, C')$.

Thus, in either case, the cost increases by at most

$$48\alpha \mathtt{OPT}_k(P, C') \leq 48\alpha(1 + \alpha/k)^i \mathtt{OPT}_k(P) \leq 48\alpha(1 + \alpha/k)^k \mathtt{OPT}_k(P) \leq 144\alpha \mathtt{OPT}_k(P).$$

But this contradicts the fact that P is (k, α)-irreducible for $\delta = 144$. This proves the tightness property. □

We now make some important observations about the *Tightness Property* in the case when there are multiset (coincident) points. These observations are important in solving the weighted versions of the clustering problems efficiently (cf. Section 7).

Observation 3.1. *In a multiset clustering problem, given a set of k centers, there always exists an optimal clustering (for these k centers) in which all the points that share the same coordinates are assigned to the same center.*

By the *Closeness Property*, every point is assigned to its closest center. Now, if there are two centers equidistant from a point, it does not matter which center they get assigned to as this does not change the cost of the solution. Therefore we can always modify an optimal clustering to get another clustering with the same cost in which all the points that share the same coordinates are assigned to the same center.

Observation 3.2. *In a multiset clustering problem, the* Tightness Property *can be extended to ensure that all coincident points either belong to S or to $P - S$, i.e., they are not split between S and $P - S$.*

This follows from the fact that the optimal clustering is determined by the Voronoi partitioning of the point set (*Closeness Property*) and Observation 3.1 above.

4 A General Algorithm for Clustering

We show that if a clustering problem $\mathcal{C}(f,k)$ satisfies the conditions stated in the previous section, then there is an algorithm which for any fixed $\varepsilon > 0$, produces with constant probability, a solution within $(1+\varepsilon)$ factor of the optimal cost. The running time of this algorithm is $O(2^{\left(\frac{k}{\varepsilon}\right)^{O(1)}} \cdot dn)$.

Fix a clustering problem $\mathcal{C}(f,k)$. Fix an instance consisting of a set P of n points in $\Re^d$. Suppose we are given a constant $\varepsilon > 0$. We present a brief outline of the algorithm.

4.1 Outline

We can assume that the solution is irreducible, i.e., removing one of the centers does not create a solution which has cost within a small factor of the optimal solution.

We start with k optimal (unknown) centers. In each iteration, we will consider the optimal clustering formed by i currently known centers and $k-i$ optimal (unknown) centers. Call this the optimal clustering of the current iteration. Our goal will be to approximate the next largest cluster, so that the resulting clustering is an approximation to the optimal clustering of the current iteration. We will bound the overall approximation factor to within a factor of $(1+\varepsilon)$ of the optimal clustering.

One of the main challenges is to accurately estimate the centers of the (optimal) clustering that may have widely varying sizes [3, 21]. In order to obtain a linear time algorithm, this must be done efficiently.

Suppose we have found centers $c'_1, \ldots, c'_i$. As the clustering problem satisfies the tightness property, we know that there exists a value r such that the points at a distance of at most r from $\{c'_1, \ldots, c'_i\}$ (set S of the tightness property) get assigned to $c_1, \ldots, c_i$ by the optimal solution induced by the centers of the optimal solution $\{c_1, \ldots, c_k\}$. So, we can delete these points. Without loss of generality, we assume that the largest cluster from amongst those centered around the unknown clusters $\{c_{i+1}, \ldots, c_k\}$ is centered around c_{i+1}. Let P'_{i+1} be this cluster. Now we can show that among the remaining points, the size of P'_{i+1} is significant. Therefore, we can use random sampling to obtain a center c'_{i+1} which is a pretty good estimate of c_{i+1}. Of course we do not know the value of r, and so a naive implementation of this idea gives an $O(n(\log n)^k)$ time algorithm.

To obtain a linear time algorithm, we reason as follows. As mentioned above, we can not guess the parameter r. So we try to guess the size of the point set obtained by removing the points in the balls of radius r centered at $\{c_1, \ldots, c_i\}$, i.e. we try to guess the size of $P-S$. So we work with the remaining point set with the hope that the time taken for this remaining point set will also be small and so the overall time will be linear. Now, we describe the actual clustering algorithm.

4.2 The Algorithm

The algorithm is described in Figures 5 and 6. Figure 5 is the main algorithm. The inputs are the point set P, k and an approximation factor ε. Let α denote $\frac{\varepsilon}{8\delta k}$, where δ is a suitable constant depending on the nature of the clustering problem, as determined by the irreducibility of the clustering problem (see Definition 3.2). The algorithm **k-clustering**(P,k,ε) tries to find the highest i such that P is (i,α)-irreducible. In that case it is enough to find i centers only. Since we do not know this value of i, the algorithm tries all possible values of i.

We now describe the algorithm **Irred-k-clustering**$(Q, m, k, C, \alpha, \mathrm{Sum})$. We have found a set C of $k-m$ centers already. The points in $P-Q$ have been assigned to C. We need to assign the remaining points in Q. The case $m=0$ is clear. In Step 2, we try to find a new center that is a $(1+\alpha/k)$-approximation to the 1-center of the next largest cluster using the *Superset Sampling*

```
Algorithm k-clustering($P, k, \varepsilon$)
Inputs : Point set $P$, number of clusters $k$, approximation ratio $\varepsilon$.
Output : $k$-clustering of $P$.

1. For $i = 1$ to $k$ do
       Obtain the clustering Irred-k-clustering($P, i, i, \phi, \varepsilon/(8\delta k), 0$).
2. Return the clustering which has minimum cost.
```

Figure 5: The k-clustering Algorithm

Procedure and the *Random Sampling Procedure* for the clustering problem. This will work provided a good fraction of the points in Q do not get assigned to C. If this is not the case then in Step 3, we assign half of the points in Q to C and call the algorithm recursively with this reduced point set. For the base case, when $|C| = 0$, as P_1 is the largest cluster, we can obtain the candidate centers by invoking $\texttt{CEN}(Q, \alpha/k, 1/k)$ instead. This is tackled in Step 2. Step 3 is not performed in this case, as there are no centers.

4.3 Analysis and Proof of Correctness

We can now show that if we have the *Random Sampling Procedure* described above, then we can get a $(1+\varepsilon)$-approximation algorithm for the clustering problem with constant probability. Further the running time of the algorithm is $O(2^{(\frac{k}{\varepsilon})^{O(1)}} dn)$.

Theorem 4.1. *Suppose a point set P is (k, α)-irreducible. Then the algorithm* **Irred-k-clustering**$(P, k, k, \emptyset, \alpha, 0)$ *returns a solution to the clustering problem $\mathcal{C}(f, k)$ on input P of cost at most $(1+\alpha)\texttt{OPT}_k(P)$ with probability γ^k, where γ is a constant.*

Proof. Consider an optimal solution to $\mathcal{C}(f, k)$ on input P. Let the centers be $\mathcal{K} = \{c_1, \ldots, c_k\}$ and let these partition the point set P into clusters $P_1, \ldots, P_k$ respectively. The only source of randomization in our algorithm is the invocations to the *Superset Sampling Procedure* (see Lemma 3.9). Recall that the desired event in the superset sampling lemma happens with constant probability. For ease of exposition, we shall assume that this desired event in fact always happens when we invoke this procedure. At the end of this proof, we will compute the actual probability with which our algorithm succeeds. Thus, unless otherwise stated, we assume that the desired event in the superset sampling lemma always happens.

Observe that when we call **Irred-k-clustering** with input $(P, k, k, \emptyset, \alpha, 0)$, it gets called recursively again several times (although with different parameters). Let $\mathcal{W}$ be the set of all calls to **Irred-k-clustering** when we start it with input $(P, k, k, \emptyset, \alpha, 0)$. Let $\mathcal{W}_i$ be those calls in $\mathcal{W}$ in which the parameter C (i.e., the set of centers already found) has size i.

For all values of i, our algorithm shall maintain the following invariant :

> **Invariant :** The set $\mathcal{W}_i$ contains a call in which the list of parameters $(Q, m, k, C, \alpha, \mathrm{Sum})$ has the following properties :
>
> (1) If the optimal solution, $\mathcal{K}$, is (k, α)-irreducible and $C'_i = \{c'_1, \ldots, c'_i\}$ is a set of i known centers then there exists a set $C''_i = \{c_{i+1}, \ldots, c_k\}$ of $k - i$ unknown centers, $C''_i \subseteq \mathcal{K}$, such that $\texttt{OPT}_k(P, \widetilde{C}_i) \leq (1 + \alpha/k)^i \texttt{OPT}_k(P)$, where $\widetilde{C}_i = C'_i \cup C''_i$.

```
Algorithm Irred-$k$-clustering($Q, m, k, C, \alpha, \mathrm{Sum}$)
Inputs : $Q$: Remaining point set, $m$: number of cluster centers yet to
        be found, $k$: total number of clusters,
        $C$: set of $k - m$ cluster centers found so far,
        $\alpha$: approximation factor, $Sum$: the cost of assigning points in $P - Q$
            to the centers in $C$
Output : The clustering of the points in $Q$ in $k$ clusters.

1. If $m = 0$
        Assign the points in $Q$ to the nearest centers in $C$.
        Sum = Sum + The clustering cost of $Q$.
        Return the clustering.
2. Use the Superset Sampling procedure and for each center $c \in \mathtt{CEN}(Q, \alpha/k, \alpha^{O(1)}/k)$
        Obtain the clustering Irred-$k$-clustering($Q, m-1, k, C \cup \{c\}, \alpha, \mathrm{Sum}$).
3. (a) Consider the points in $Q$ in ascending order of distance from $C$.
   (b) Let $U$ be the first $|Q|/2$ points in this sequence.
   (c) Assign the points in $U$ to the nearest centers in $C$.
   (d) Sum = Sum + The clustering cost of $U$.
   (e) Compute the clustering Irred-$k$-clustering($Q - U, m, k, C, \alpha, \mathrm{Sum}$).
4. Return the clustering which has minimum cost.
```

Figure 6: The irreducible k-clustering algorithm

(2) Let $P'_1, \ldots, P'_k$ be the partitioning of P if we choose $\widetilde{C}_i$ as the set of centers (in other words this is the Voronoi partitioning of P with respect to $\widetilde{C}_i$), where $\widetilde{C}_i$ is as defined above. Then the set $P - Q$ is a subset of $P'_1 \cup \cdots \cup P'_i$.

Clearly, if we show that the invariant holds for $i = k$, then we are done. It holds trivially for $i = 0$. Suppose the invariant holds for some fixed i. We shall show that the invariant holds for $(i + 1)$ as well. We assume w.l.o.g. that P'_{i+1} is the largest cluster amongst $P'_{i+1} \cup \cdots \cup P'_k$. Our algorithm approximates the center of P'_{i+1} and we show that this center when added to C'_i forms the required set of centers C'_{i+1} for the next iteration.

As the invariant holds for i, there exist parameter lists in $\mathcal{W}_i$ which satisfy the invariant properties mentioned above. Consider any such parameter list. As the conditions of the *Tightness Property* are met, there exists a set S contained in $P'_1 \cup \cdots \cup P'_i$ such that $P - S$ contains at most $|P'_{i+1}|/\alpha$ points of $P'_1 \cup \cdots \cup P'_i$. Let $\bar{P}$ denote $P - S$.

We show in Claim 4.2 that the invariant properties imply that Q contains all the points of $P'_{i+1} \cup \cdots \cup P'_k$. Further, in Claim 4.3, we show that P'_{i+1} forms a constant fraction of the points of $\bar{P}$, i.e., $|P'_{i+1}| \geq \frac{\alpha^{O(1)}}{k}|\bar{P}|$. It follows that $|P'_{i+1}| \geq \frac{\alpha^{O(1)}}{k}|\bar{P} \cap Q|$. So, if we knew $\bar{P}$, then we could get a point c'_{i+1} which is a $(1 + \alpha/k)$ approximation to c_{i+1} (as the 1-center of the cluster P'_{i+1}) by sampling $O((\lambda_{\alpha/k} k/\alpha^{O(1)}))$ points from $\bar{P} \cap Q$, and generating the candidate center set of size $O(2^{(\frac{k}{\alpha}\lambda_{\alpha/k})^{O(1)}}\beta_{\alpha/k})$ as described by the *Random Sampling Procedure* and the *Superset Sampling Procedure*. But of course we do not know $\bar{P}$.

Note that we actually only require a constant factor approximation to $\bar{P} \cap Q$. Amongst the parameter lists in $\mathcal{W}_i$ satisfying the invariant conditions mentioned above, choose a list $(Q, m, k, C, \alpha, \mathrm{Sum})$ for which $|Q|$ is smallest.

Note that condition (b) of the tightness property suggests that when we consider the points in

P (and therefore Q) in order of distance from the centers in C_i', then the points of S are closer than the points of $\bar{P}$. Therefore, we can eliminate half the remaining points of Q in order of distance from these centers until we get to a factor 2 approximation to $\bar{P} \cap Q$. This is done recursively in Step 3 of the algorithm. In fact, a call to the algorithm **Irred-k-clustering** that corresponds to Q being a factor 2 approximation to $\bar{P} \cap Q$ corresponds to a parameter list $(Q, m, k, C, \alpha, \mathrm{Sum})$ in $\mathcal{W}_i$, satisfying the invariant conditions, for which $|Q|$ is smallest. We prove this in Lemma 4.4.

Note that this is similar to the process of eliminating points in order of distance from the approximate center of the larger cluster till we get to a constant fraction of the points in the second cluster in the 2-means clustering algorithm described in Section 3.

We now formally prove the above claims.

Claim 4.2. *$P_{i+1}' \cup \cdots \cup P_k'$ is contained in $\bar{P} \cap Q$.*

Proof. We already know that S is contained in $P_1' \cup \cdots \cup P_i'$. Therefore, $P_{i+1}' \cup \cdots \cup P_k'$ is contained in $\bar{P}$. Moreover, from invariant (2), we have that $P_{i+1}' \cup \cdots \cup P_k' \subseteq Q$. Claim 4.2 follows. □

Claim 4.3. $|P_{i+1}'| \geq \frac{\alpha^{O(1)}}{k}|\bar{P}|$.

Proof. By the tightness property, we know that there are at most $|P_{i+1}'|/\alpha$ elements of $P_1' \cup \ldots \cup P_i'$ in $\bar{P}$. Therefore, since $P_{i+1}', \ldots, P_k'$ are the clusters associated with the centers in C_i'' and P_{i+1}' is the largest of these clusters, we have $|\bar{P}| \leq |P_{i+1}'|/\alpha^{O(1)} + |P_{i+1}'| + \ldots + |P_k'| \leq |P_{i+1}'|/\alpha^{O(1)} + k|P_{i+1}'| \leq \frac{k}{\alpha^{O(1)}}|P_{i+1}'|$.

This proves Claim 4.3. □

Recall that we are considering the parameter list $(Q, m, k, C, \alpha, \mathrm{Sum})$ in $\mathcal{W}_i$, satisfying the invariant conditions, for which $|Q|$ is smallest.

Lemma 4.4. *$|\bar{P} \cap Q| \geq |Q|/2$.*

Proof. Suppose not, i.e., $|\bar{P} \cap Q| \leq |Q|/2$.

Claim 4.5. *Consider the points in Q sorted in ascending order of the distance from C. Let U be the first $|Q|/2$ points in this order. Then U does not contain a point of $\bar{P} \cap Q$.*

Proof. Follows from condition (b) of the *Tightness Property* for the clustering problem and the assumption that $|\bar{P} \cap Q| \leq |Q|/2$. □

So, if U is as defined in the claim above, then $\bar{P} \cap Q$ is a subset of $Q - U$. Since $P_{i+1}' \cup \cdots \cup P_k'$ is contained in $\bar{P} \cap Q$ (because of Claim 4.2 and the fact that Q is in the parameter list which satisfies the invariant for i), it follows that $P_{i+1}' \cup \cdots \cup P_k'$ is a subset of $Q - U$. Thus, the parameter list $(Q - U, C, k, m, \alpha, \mathrm{Sum})$ which is formed in Step 3(e) of the algorithm satisfies the invariant for i as well, i.e., it is in $\mathcal{C}_i$. But this violates the fact that $(Q, C, k, m, \alpha, \mathrm{Sum})$ was the parameter list satisfying the invariant for i in $\mathcal{C}_i$ for which $|Q|$ is smallest.

This proves Lemma 4.4. □

The lemma above implies that $|\bar{P} \cap Q| \geq |Q|/2$. Combined with Claim 4.3, we get $|P_{i+1}'| \geq \frac{\alpha^{O(1)}|Q|}{4k}$. The superset sampling lemma combined with the claim above imply that by sampling $O(\lambda_{\alpha/k} k/\alpha^{O(1)})$ points from Q and generating the candidate center set as described by the *Random Sampling Procedure*, $\mathcal{A}$, for the clustering problem, we shall get a point c_{i+1}' such that $f(P_{i+1}', c_{i+1}') \leq (1 + \alpha/k) f(P_{i+1}', c_{i+1})$, where $c_{i+1} \in C_i''$ is the center of P_{i+1}' in the optimal clustering induced by $\widetilde{C}_i$. This is done in the *Superset Sampling Procedure*. This is the case handled by the Step 2 in the algorithm **Irred-k-clustering**. In this case the algorithm is called again with parameters $(Q, m-1, k, C \cup \{c'_{i+1}\}, \alpha, \mathrm{Sum})$. It is easy to see now that this parameter list satisfies the invariant for $i+1$. The set of known centers C'_{i+1} for the next iteration is $C'_i \cup \{c'_{i+1}\}$ and the set of unknown centers C''_{i+1} is $C''_i \backslash \{c_{i+1}\}$. Since $f(P'_{i+1}, c'_{i+1}) \leq (1+\alpha/k) f(P'_{i+1}, c_{i+1})$ and the clustering problem satisfies the closeness property, it follows that $\mathtt{OPT}_k(P, C'_{i+1} \cup C''_{i+1}) \leq (1+\alpha/k)\mathtt{OPT}_k(P, \widetilde{C}_i) \leq (1+\alpha/k)^{i+1}\mathtt{OPT}_k(P)$. Thus we have shown that the invariant holds for all values of i.

As we mentioned earlier, a parameter list $(Q, m, k, C, \alpha, \mathrm{Sum})$ which satisfies the invariant for $i = k$ has the desired centers in C.

It is easy to verify that the cost reported by the algorithm $\mathtt{OPT}_k(P, C)$ satisfies

$$\mathtt{OPT}_k(P) \leq \mathtt{OPT}_k(P, C) \leq (1+\alpha/k)^k \mathtt{OPT}_k(P) \leq (1+2\alpha)\mathtt{OPT}_k(P) \leq (1+\varepsilon/4)\mathtt{OPT}_k(P).$$

This proves the correctness of our algorithm. We just need to calculate the probability with which the algorithm is called with such a parameter list.

Note that the only source of randomness in **Irred-k-clustering** is in the Step 2(a). The sampling gives the desired result with constant probability (according to Lemma 3.9). Further each time we execute Step 2, we decrease m by 1. So, in any sequence of successive recursive calls, there can be at most k invocations of Step 2. Now, we have just shown that there is a parameter list in $\mathcal{W}_k$ for which C contains a set of centers close to the optimal clusters. Let us look at the sequence of recursive calls which have resulted in this parameter list. In these sequence of calls, as we mentioned above, there are k invocations of the random sampling. Each of these work correctly with constant probability. Therefore, the probability that we actually see this parameter list during the execution of this algorithm is γ^k for some constant γ.

This completes the proof of Theorem 4.1 □

Now we establish the running time of our algorithm.

Theorem 4.6. *The algorithm* **Irred-k-clustering** *when called with parameters* $(P, k, k, \emptyset, \alpha, 0)$ *runs in time* $O(2^{(k/\alpha)^{O(1)}} dn)$*, where* $n = |P|$.

Proof. Let $T(n, m)$ be the running time of our algorithm on input $(Q, m, k, C, \alpha, \mathrm{Sum})$ where $n = |Q|$. Then in the invocation of *Superset Sampling Procedure* in Step 2, we have $u(k, \alpha)$ subsets of the sample, where $u(k, \alpha) = O(2^{(\lambda_{\alpha/k} \frac{k}{\alpha})^{O(1)}})$. Computation of the candidate center set from any set S' takes $O(\eta_{\alpha/k} \cdot nd)$ time. Steps 3(a)-(d) take $O(nd)$ time. Therefore we get the recurrence

$$T(n, m) = O(u(k, \alpha) \cdot \beta_{\alpha/k}) T(n, m-1) + T(n/2, m) + O(u(k, \alpha) \cdot \eta_{\alpha/k} \cdot nd).$$

Let $\lambda_\alpha = O(1/\alpha^{O(1)})$, $\beta_\alpha = O(2^{(1/\alpha)^{O(1)}})$ and $\eta_\alpha = O(2^{(1/\alpha)^{O(1)}})$. Choose $c = O(2^{(k/\alpha)^\gamma})$ to be large enough, for a suitable constant γ, such that

$$T(n, m) \leq c \cdot T(n, m-1) + T(n/2, m) + c \cdot nd.$$

We claim that $T(n, m) \leq c^m \cdot 2^{3m^2} \cdot nd$. The proof is by induction. Consider the base cases. $T(n, 0) = knd$ as there are no more centers to be determined and the points only need to be assigned to the closest center. Also, $T(0, m) = 0$ as there are no points and therefore it holds vacuously. For the inductive step, suppose that the claim holds for $T(n', m')$ $\forall n', \forall m' < m$ and it holds for $T(n', m')$ $\forall n' < n, \forall m'$. Then, we are required to show that

$$c^m \cdot 2^{3m^2} \cdot nd \geq c \cdot c^{m-1} \cdot 2^{3(m-1)^2} \cdot nd + c^m \cdot 2^{3m^2} \cdot \frac{n}{2} d + c \cdot nd.$$

For this, it suffices to show that $2^{3m^2} \geq 2^{3(m-1)^2} + 2^{3m^2-1} + 1$ which clearly holds for $m \geq 1$.

It follows that $T(n,k)$ is $O(2^{(k/\alpha)^{O(1)}} dn)$ when $\lambda_\alpha = O(1/\alpha^{O(1)})$, $\beta_\alpha = O(2^{(1/\alpha)^{O(1)}})$ and $\eta_\alpha = O(2^{(1/\alpha)^{O(1)}})$. □

We can now state our main Theorem.

Theorem 4.7. *For a clustering problem satisfying the* Closeness Property*,* Tightness Property *and for which there exists a* Random Sampling Procedure*, a $(1+\varepsilon)$-approximate solution for a point set P in $\Re^d$ can be found in time $O(2^{(k/\varepsilon)^{O(1)}} dn)$, with constant probability.*

Proof. We can run the algorithm **Irred-k-clustering** c^k times for some constant c to ensure that it yields the desired result with constant probability. This still keeps the running time $O(2^{(k/\alpha)^{O(1)}} dn)$. So let us assume this algorithm gives the desired solution with constant probability.

Notice that the running time of our main algorithm in Figure 5 is also $O(2^{(k/\alpha)^{O(1)}} dn)$. We just have to show that it is correct.

Let i be the highest index for which P is (i,α)-irreducible. So, it follows that

$$\mathtt{OPT}_i(P) \leq (1+\delta k\alpha)\mathtt{OPT}_{i+1}(P) \leq \cdots \leq (1+\delta k\alpha)^{k-i}\mathtt{OPT}_k(P) \leq (1+\varepsilon/4)\mathtt{OPT}_k(P).$$

Further, we know that the algorithm **Irred-k-clustering** on input $(P,i,i,\emptyset,\alpha,0)$ yields a set of i centers C for which $\mathtt{OPT}_k(P,C) \leq (1+\varepsilon/4)\mathtt{OPT}_i(P)$. Therefore, we get a solution of cost at most $(1+\varepsilon/4)(1+\varepsilon/4)\mathtt{OPT}_k(P) \leq (1+\varepsilon)\mathtt{OPT}_k(P)$. This proves the Theorem. □

We now give applications to various clustering problems. We show that these clustering problems satisfy the tightness property and admit a random sampling procedure as described in the previous section.

For the k-means clustering problem, the random sampling procedure follows from Lemma 3.3 shown by Inaba et al [14], and the tightness property follows from Lemma 3.10. This leads to the following Corollary to Theorem 4.7.

Corollary 4.8. *Given a point set P of n points in $\Re^d$, a $(1+\varepsilon)$-approximate solution to the k-means clustering problem can be found in time $O(2^{(k/\varepsilon)^{O(1)}} dn)$, with constant probability.*

5 k-median Clustering

As described earlier, the clustering problem $\mathcal{C}(f,k)$ is said to be the k-median problem if $f(Q,x) = \sum_{q\in Q} d(q,x)$. We now exhibit the Random Sampling Procedure and the Tightness Property for this problem leading to the following Corollary to Theorem 4.7.

Corollary 5.1. *Given a point set P of n points in $\Re^d$, a $(1+\varepsilon)$-approximate solution to the k-median clustering problem can be found in time $O(2^{(k/\varepsilon)^{O(1)}} dn)$, with constant probability.*

5.1 Random Sampling Procedure

Badoiu et al. [3] showed that a small random sample can be used to get a close approximation to the optimal 1-median solution. Given a set of points P, let $\mathtt{AvgMed}(P)$ denote $\frac{\mathtt{OPT}_1(P)}{|P|}$, i.e., the average cost paid by a point towards the optimal 1-median solution.

Lemma 5.2. *[3] Let P be a set of points in $\Re^d$, and ε be a constant between 0 and 1. Let X be a random sample of $O(1/\varepsilon^3 \log 1/\varepsilon)$ points from P. Then with constant probability, the following two events happen: (i) The flat span(X) contains a point x such that $\mathtt{OPT}_1(P,\{x\}) \leq (1+\varepsilon)\mathtt{OPT}_1(P)$. and (ii) X contains a point y at distance at most $2\mathtt{AvgMed}(P)$ from x.*

We now show that if we can upper and lower bound $\texttt{AvgMed}(P)$ up to constant factors, then we can construct a small set of points such that at least one of these is a good approximation to the optimal center for the 1-median problem on P.

Lemma 5.3. *Let P be a set of points in $\Re^d$ and X be a random sample of size $O(1/\varepsilon^3 \log 1/\varepsilon)$ from P. Suppose we know numbers a and b such that $a \leq \texttt{AvgMed}(P) \leq b$. Then, we can construct a set Y of $O(2^{(1/\varepsilon)^{O(1)}} \log(b/\varepsilon a))$ points such that with constant probability there is at least one point $z \in X \cup Y$ satisfying $\texttt{OPT}_1(P, \{z\}) \leq (1+2\varepsilon)\texttt{OPT}_1(P)$. Further, the time taken to construct Y from X is is $O(2^{(1/\varepsilon)^{O(1)}} \log(b/\varepsilon a)d)$.*

Proof. Our construction is similar to that of Badoiu et al. [3]. We can assume that the result stated in Lemma 5.2 holds (because this happens with constant probability). Let x and y be as in Lemma 5.2.

We will carefully construct candidate points around the points of X in *span*(X) in an effort to get within close distance of x.

For each point $p \in X$, and each integer i in the range $[\lfloor \log \frac{\varepsilon}{4} a \rfloor, \lceil \log b \rceil]$ we do the following – let $t = 2^i$. Consider the grid $G_p(t)$ of side length $\varepsilon t/(4|X|) = O(t\varepsilon^4 \log(1/\varepsilon))$ in *span*(X) centered at p. We add all the vertices of this grid lying within distance at most $2t$ from p to our candidate set Y. This completes the construction of Y. The number of vertices in a grid $G_p(t)$ is $O(2t/(t\varepsilon^4 \log(1/\varepsilon)))^{O(1/\varepsilon^3 \log 1/\varepsilon)} = O(2^{(1/\varepsilon)^{O(1)}})$. The number of such grids considered is $O((1/\varepsilon^3 \log 1/\varepsilon) \cdot \log(4b/\varepsilon a))$. Hence the total size of Y is $O(2^{(1/\varepsilon)^{O(1)}} \log(b/\varepsilon a))$. The time taken to construct Y from X is proportional to the number of points in Y and hence $O(2^{(1/\varepsilon)^{O(1)}} \log(b/\varepsilon a)d)$.

We now show the existence of the desired point $z \in X \cup Y$. Consider the following cases:

1. $d(y,x) \leq \varepsilon \texttt{AvgMed}(P)$: Using triangle inequality, we see that

$$f(P,y) \leq f(P,x) + |P| d(y,x) \leq (1+2\varepsilon)\texttt{OPT}_1(P).$$

   Therefore y itself is the required point.

2. $d(y,x) > \varepsilon \texttt{AvgMed}(P)$: Consider the value of i such that $2^{i-1} \leq \texttt{AvgMed}(P) \leq 2^i$ – while constructing Y, we must have considered this value of i for all points in X. Let $t = 2^i$. Clearly, $t/2 \leq \texttt{AvgMed}(P) \leq t$.

   Observe that $d(y,x) \leq 2\texttt{AvgMed}(P) \leq 2t$. Therefore, in the manner by which we have constructed $G_y(t)$, there must be a point $p \in G_y(t)$ for which $d(p,x) \leq \varepsilon t/2 \leq \varepsilon \texttt{AvgMed}(P)$. This implies that

$$f(P,p) \leq f(P,x) + |P| d(x,p) \leq (1+2\varepsilon)\texttt{OPT}_1(P).$$

   Hence p is the required point.

This completes the proof of Lemma 5.3. □

We now show the existence of the random sampling procedure.

Theorem 5.4. *Let P be a set of n points in $\Re^d$, and let ε be a constant, $0 < \varepsilon < 1/12$. There exists an algorithm which given a random sample, R, of $O((\frac{1}{\varepsilon})^{O(1)})$ points from P constructs a set of points $\texttt{core}(R)$ such that with constant probability there is a point $x \in \texttt{core}(R)$ satisfying $f(P,x) \leq (1+O(\varepsilon))\texttt{OPT}_1(P)$. Further, the time taken to construct $\texttt{core}(R)$ from R is $O(2^{(1/\varepsilon)^{O(1)}} d)$.*

Proof. Consider the optimal 1-median solution for P – let c be the center in this solution. Let T denote $\texttt{AvgMed}(P)$. Consider the ball B_1 of radius T/ε^2 around c. Let P' be the points of P contained in B_1. It is easy to see that $|P'| \geq (1-\varepsilon^2)n$.

Let R be a random sample of size $1/\varepsilon + 1$ points from P. We split R into two parts. Let p be a random point of R and Q be the remaining $1/\varepsilon$ points. Then clearly p and Q are random samples of P having size 1 and $1/\varepsilon$ respectively.

With constant probability, p lies in P'. With constant probability, the points of Q also lie in P'. So we assume that these two events happen. Let $v = \sum_{q \in Q} d(q,p)$. We want to show that v is actually close to $\texttt{AvgMed}(P)$.

Let B_2 denote the ball of radius εT centered at p. One of the following two cases must happen:

- There are at least $2\varepsilon|P'|$ points of P' outside B_2 : In this case, with constant probability, the sample Q contains a point outside B_2. Therefore, $v \geq \varepsilon T$. Also notice that any two points in B_1 are at distance at most $2T/\varepsilon^2$ from each other. So, $v \leq 2T|Q|/\varepsilon^2$. We choose $a = \frac{v\varepsilon^2}{2|Q|}$ and $b = v/\varepsilon$. Notice that b/a is $O(1/\varepsilon^{O(1)})$. We can now use the Lemma 5.3 to construct the desired core set.

- There are at most $2\varepsilon|P'|$ points of P' outside B_2 : Suppose $d(p,c) \leq 4\varepsilon T$. In this case $f(P,p) \leq (1+O(\varepsilon))\texttt{OPT}_1(P)$ and we are done. So assume this is not the case. Note that the number of points outside B_2 is at most $|P - P'| + 2\varepsilon|P'| \leq \varepsilon^2 n + 2\varepsilon(1-\varepsilon^2)n \leq 3\varepsilon n$. Now suppose we assign all points of P from c to p. Let us see the change in cost. The distance the points in B_2 have to travel decreases by at least $d(c,p) - 2\varepsilon R$. The increase in the distance for points outside B_2 is at most $d(c,p)$. So the overall decrease in cost is at least

$$|B_2|(d(c,p) - 2\varepsilon R) - (n - |B_2|)d(c,p) > 0$$

if we use $|B_2| \geq n(1-3\varepsilon)$ and $d(c,p) \geq 4\varepsilon R$. This yields a contradiction because c is the optimal center. Thus we are done in this case as well.

This proves Theorem 5.4. □

Thus we have shown the existence of the random sampling procedure.

5.2 Tightness Property

We now show the tightness property.

Lemma 5.5. *The k-median clustering problem, having cost function $f_1(Q,x) = \sum_{q \in Q} d(q,x)$ satisfies the tightness property.*

Proof. We need to show the existence of the desired set S.

Consider the closest pair of centers between the sets $\widetilde{C}_i \backslash C'_i$ and C'_i – let these centers be c_l and c'_r respectively. Let $t = d(c_l, c'_r)$. Let S be the set of points $\mathcal{B}(c'_1, t/4) \cup \cdots \cup \mathcal{B}(c'_i, t/4)$, i.e., the points which are distant at most $t/4$ from $C'_i = \{c'_1, \ldots, c'_i\}$.

Clearly, S is contained in $P'_1 \cup \cdots \cup P'_i$. This shows (a). Also, for any $x \in S, x' \in P - S$, $d(x, \{c'_1, \ldots, c'_i\}) \leq d(x', \{c'_1, \ldots, c'_i\})$. This proves (b).

Suppose $P - S$ contains more than $|P_l|/\alpha$ points of $P'_1 \cup \cdots \cup P'_i$. In that case, these points are assigned to centers at distance at least $t/4$. It follows that $\texttt{OPT}_k(P, \widetilde{C}_i)$ is at least $\frac{t|P_l|}{4\alpha}$. This implies that $t|P_l| \leq 4\alpha\texttt{OPT}_k(P, \widetilde{C}_i)$. But then if we assign all the points in P_l to c'_r, the cost increases by at most

$$|P_l|t \leq 4\alpha\texttt{OPT}_k(P, \widetilde{C}_i) \leq 4\alpha(1+\alpha/k)^i\texttt{OPT}_k(P) \leq 4\alpha(1+\alpha/k)^k\texttt{OPT}_k(P) \leq 12\alpha\texttt{OPT}_k(P).$$

But this contradicts the fact that P is (k, α)-irreducible for $\delta = 12$. This proves the tightness property. □

5.3 Applications to the 1-median Problem

In this section, we present an algorithm for the 1-median problem. Given a set of n points in $\Re^d$, the algorithm with constant probability produces a solution of cost at most $(1+\varepsilon)$ of the optimal cost for any constant $\varepsilon > 0$. The running time of the algorithm is $O(2^{1/\varepsilon^{O(1)}} d)$, assuming that it is possible to randomly sample a point in constant time.

Our algorithm is based on the following idea presented by Indyk [15].

Lemma 5.6. *[15] Let X be a set of n points in $\Re^d$. For a point $a \in \Re^d$ and a subset $Q \subseteq X$, define $S_Q(a) = \sum_{x \in Q} d(a, x)$ and $S(a) = S_X(a)$. Let ε be a constant, $0 \le \varepsilon \le 1$. Suppose a and b are two points such that $S(b) > (1+\varepsilon)S(a)$. Then, for a random sample Q obtained from X,*

$$Pr\left(\sum_{x \in Q} d(a, x) \geq \sum_{x \in Q} d(b, x)\right) < e^{-\varepsilon^2 |Q|/64}.$$

We now show the existence of a fast algorithm for approximating the optimal 1-median solution.

Theorem 5.7. *Let P be a set of n points in $\Re^d$, and let ε be a constant, $0 < \varepsilon < 1$. There exists an algorithm which randomly samples a set R of $O((\frac{1}{\varepsilon})^{O(1)})$ points from P. Using this sample only, it finds a point p such that $f(P, x) \le (1 + O(\varepsilon))\mathtt{OPT}_1(P)$ with constant probability (independent of ε). The time taken by the algorithm to find such a point p from R is $O(2^{(1/\varepsilon)^{O(1)}} d)$.*

Proof. We first randomly sample a set R_1 of $O((\frac{1}{\varepsilon})^{O(1)})$ points from P and using Theorem 5.4, construct a set $\mathtt{core}(R_1)$ of $O(2^{(1/\varepsilon)^{O(1)}})$ points such that with constant probability, there is a point $x \in \mathtt{core}(R_1)$ satisfying $f(P, x) \le (1 + O(\varepsilon))\mathtt{OPT}_1(P)$.

Now we randomly sample a set R_2 of $O((1/\varepsilon)^{O(1)})$ points and find the point $p \in \mathtt{core}(R_1)$ for which $S_{R_2}(p) = f_1(R_2, p)$ is minimum. By Lemma 5.6, p is with constant probability a $(1 + O(\varepsilon))$-approximate median of P.

Clearly, the time taken by the algorithm is $O(2^{(1/\varepsilon)^{O(1)}} d)$. □

Also note that we can boost the success probability to an arbitrarily small constant by selecting a large enough (yet constant) sample R.

6 Discrete k-means Clustering

This is the same as the k-means problem with the additional constraint that the centers must be chosen from the input point set only. The tightness property follows from Lemma 3.10. We now exhibit the Random Sampling Procedure for this problem leading to the following Corollary to Theorem 4.7.

Corollary 6.1. *Given a point set P of n points in $\Re^d$, a $(1+\varepsilon)$-approximate solution to the discrete k-means clustering problem can be found in time $O(2^{(k/\varepsilon)^{O(1)}} dn)$, with constant probability.*

6.1 Random Sampling Procedure

We first show that given a good approximation to the center of the optimal (continuous) 1-means problem, we can get a good approximation to the center of the optimal discrete 1-means problem. Let P be a set of n points in $\Re^d$. Let c be the center of the optimal solution to the (continuous) 1-means problem on P.

Lemma 6.2. *Let α be a constant, $0 < \alpha < 1$, and c' be a point in $\Re^d$ such that $\sum_{p \in P} d(p, c')^2 \leq (1+\alpha) \sum_{p \in P} d(p, c)^2$. Let x' be the point of P closest to c'. Then $\texttt{OPT}_1(P, \{x'\}) \leq (1 + O(\sqrt{\alpha}))\texttt{OPT}_1(P)$.*

Proof. Let x be the center of the optimal discrete 1-means solution, i.e., $\texttt{OPT}_1(P, \{x\}) = \texttt{OPT}_1(P)$. Let T be the average cost paid by the points of P in the optimal 1-means solution, i.e., $T = \frac{\sum_{p \in P} d(p,c)^2}{|P|}$.

Then $\texttt{OPT}_1(P) = |P|(T + d(c, x)^2)$ and $\texttt{OPT}_1(P, \{x'\}) = |P|(T + d(c, x')^2)$. From the definition of c', we know that $d(c, c')^2 \leq \alpha T$.

Notice that

$$d(c, x') \leq d(c, c') + d(c', x') \leq d(c, c') + d(c', x) \leq 2d(c, c') + d(c, x).$$

We know that $f_2(P, x) = |P|(T + d(c, x)^2)$ and $f_2(P, x') = |P|(T + d(c, x')^2)$. So

$$\begin{aligned} f_2(P, x') - f_2(P, x) &= |P|(d(c, x')^2 - d(c, x)^2) \\ &\leq |P| \left((2d(c, c') + d(c, x))^2 - d(c, x)^2)\right) \\ &\leq 4|P| \left(d(c, c')^2 + d(c, c')d(c, x)\right) \\ &\leq 4|P| \left(\alpha T + \sqrt{\alpha T} d(c, x)\right) \\ &\leq 4|P| \left(\alpha T + \sqrt{\alpha}(T + d(c, x)^2)\right) \\ &\leq O(\sqrt{\alpha})\texttt{OPT}_1(P). \end{aligned}$$

□

We now show the existence of the random sampling procedure.

Theorem 6.3. *Let α be a constant, $0 < \alpha < 1$. There exists an algorithm which given a random sample,R, of $O\left(\frac{1}{\alpha}\right)$ points from P, finds a singleton set $\texttt{core}(R)$ such that with constant probability the point $x \in \texttt{core}(R)$ satisfies $f(P, x) \leq (1+O(\sqrt{\alpha}))\texttt{OPT}_1(P)$. Further, the time taken to construct $\texttt{core}(R)$ from R is $O((\frac{1}{\alpha} + n)d)$.*

Proof. Using Lemma 3.3, we can get a point c' using the random sample R, such that $\sum_{p \in P} d(p, c')^2 \leq (1+\alpha) \sum_{p \in P} d(p, c)^2$. As mentioned in the lemma, we do this by by taking the centroid of a random sample of $O(1/\alpha)$ points of P. This takes time $O(\frac{1}{\alpha} \cdot d)$.

The rest follows from the previous lemma. □

7 Weighted Clustering

In this section, we consider the situation in which each point p has an integral weight w_p associated with it. Let W be the total sum of all the weights and n be the number of distinct points.

Let us look at the cost function for some weighted clustering problems.

- Weighted k-median : $f_1(Q, x) = \sum_{q \in Q} w_q \cdot d(q, x)$.

- Weighted k-means : $f_2(Q, x) = \sum_{q \in Q} w_q \cdot d(q, x)^2$.

For a set of points S, let $W_S = \sum_{s \in S} w_s$ be the weighted sum of the points in S.

An important observation is that the solution to the above weighted problems is the same as the solution to the corresponding unweighted version of the problems where a point p with weight w is replaced with w points (of unit weight). The algorithm and proofs of the unweighted version extend with little or no change by virtue of this observation. We outline these extensions below.

The *Irreducibility* definition and the *Closeness Property* extend to the weighted version without any change. The *Random Sampling Procedure* is modified as follows.

> ***Weighted Random Sampling Procedure* :** There exists a procedure $\mathcal{A}$ that takes as input a parameter α (a constant), and a set of points $R \in \Re^d$ of size λ_α. $\mathcal{A}$ produces as output, another set of points called $\mathtt{core}(R)$, of constant size, β_α. $\mathcal{A}$ satisfies the condition that if R is a random sample obtained from a weighted set Q, then with constant probability there is at least one point $c \in \mathtt{core}(R)$ such that $\mathtt{OPT}_1(Q, \{c\}) \leq (1+\alpha)\mathtt{OPT}_1(Q)$. Further the time taken by $\mathcal{A}$ to produce $\mathtt{core}(R)$ from R is at most $O(\eta_\alpha \cdot dn)$, where n is the size (number of distinct points) of Q and η_α is a constant.

Note that, as in the unweighted case, when we actually apply the *Weighted Random Sampling Procedure*, the set Q will not be explicitly known - instead a superset $P \supseteq Q$ will be given. We will sample a slightly larger set of points from P and then isolate a subset of points of Q to be supplied to the *Weighted Random Sampling Procedure*. Our sampling/isolation procedure must additionally satisfy the condition that any point set obtained from the underlying set Q after sampling/isolation must be equiprobable amongst all the point sets of the same size in Q_m (obtained with replacement) where Q_m is the multiset obtained by replacing each weighted point $p \in Q$ by w_p points.

We now extend the *Superset Sampling Procedure* to the weighted problem, provided that the total weight of the subset Q is a constant fraction of the total weight of P. To do this we perform weighted sampling, i.e., while sampling every point, a point p is picked with probability w_p/W. We then isolate a subset of Q by enumerating all subsets of the sample.

> ***Weighted Superset Sampling Procedure* :** This procedure takes as input a parameter α (a constant), another parameter $0 < \theta \leq 1$ (also a constant) and a set of points $P \in \Re^d$, such that $W_Q \geq \theta W_P$, where $|Q| \subseteq P$ is fixed but not explicitly specified. It produces as output another set of points called $\mathtt{WCEN}(P, \alpha, \theta)$, of constant size $O(2^{(\frac{\lambda_\alpha}{\theta})^{O(1)}} \beta_\alpha)$, such that with constant probability there is at least one point $c \in \mathtt{WCEN}(P, \alpha, \theta)$ such that $\mathtt{OPT}_1(Q, \{c\}) \leq (1+\alpha)\mathtt{OPT}_1(Q)$.
>
> This Procedure works as follows. It takes a weighted sample S of size $O(\frac{4\lambda_\alpha}{\theta})$ from P. It then considers all possible subsets of size λ_α of S. There are $O(2^{(\frac{\lambda_\alpha}{\theta})^{O(1)}})$ of these. For each of these subsets S', it generates a candidate center set for the 1-center for the clustering problem using the *Weighted Random Sampling Procedure*, $\mathcal{A}$. It returns the union of all these candidate center sets.

The following lemma shows that one of the subsets considered by the *Weighted Superset Sampling Procedure* must give a close enough approximation to the optimal 1-center solution for Q.

Lemma 7.1. *(Weighted Superset Sampling Lemma) Let* $\mathtt{core}(S')$ *be the center set generated using the* Weighted Random Sampling Procedure *on sampled subset* S'. *Then, the following event happens with constant probability*

$$\min_{c' \in \mathtt{core}(S') : S' \subset S, |S'| = \lambda_\alpha} f(Q, c') \leq (1+\alpha)\mathtt{OPT}_1(Q).$$

Proof. With constant probability, S contains at least λ_α points from Q, the required sample size. Moreover this set S' of λ_α points is equiprobable amongst all the point sets of the same size in Q_m (obtained with replacement) where Q_m is the corresponding multiset obtained by replacing a weighted point $p \in Q$ with w_p points. The rest follows from the *Weighted Random Sampling Procedure* for the clustering problem. □

For the *Tightness Property*, the size of the set is substituted with the weight of the set. We restate this property below.

> ***Weighted Tightness Property*:** Let P be a set of points which is (k, α)-irreducible for some constant α. Consider an optimal solution to $\mathcal{C}(f, k)$ on P – let $C = \{c_1, \ldots, c_k\}$ be the centers in this solution. Suppose we have a set of i points $C'_i = \{c'_1, \ldots, c'_i\}$, such that $\mathtt{OPT}_k(P, \widetilde{C}_i) \leq (1 + \alpha/k)^i \mathtt{OPT}_k(P)$, where $\widetilde{C}_i = \{c'_1, \ldots, c'_i, c_{i+1}, \ldots, c_k\}$. Let $P'_1, \ldots, P'_k$ be the partitioning of P if we choose $\widetilde{C}_i$ as the set of centers (in other words this is the Voronoi partitioning of P with respect to $\widetilde{C}_i$). We assume w.l.o.g. that P'_{i+1} be the largest cluster amongst $P'_{i+1}, \ldots, P'_k$. Then there exists a set of points S such that the following conditions hold :
>
> (a) S is contained in $P'_1 \cup \ldots \cup P'_i$.
>
> (b) Let $x \in S, x' \in P - S$. Then, $d(x, \{c'_1, \ldots, c'_i\}) \leq d(x', \{c'_1, \ldots, c'_i\})$.
>
> (c) $P - S$ contains at most $\frac{w_{P'_{i+1}}}{\alpha^{O(1)}}$ points of $P'_1 \cup \ldots \cup P'_i$.

It is important to note here that given a *Random Sampling Procedure* for an unweighted clustering problem, the corresponding *Weighted Random Sampling Procedure* for the weighted version of the problem can be simply obtained by performing weighted sampling as described above. Similarly, it is easy to see that the *Weighted Superset Sampling Procedure* and the *Weighted Tightness Property* translate into the *Superset Sampling Procedure* and the *Tightness Property* for the corresponding unweighted version of the problem.

7.1 The Weighted Algorithm

The algorithm extends to the weighted version as follows. The algorithm *k-clustering* remains unchanged. The algorithm *Irred-k-clustering* is modified as shown in Figure 7. We call the weighted version of this algorithm, the **Weighted-Irred-k-clustering** algorithm.

Step 2 is modified to perform weighted random sampling. Based on the *Weighted Superset Sampling Lemma*, now sampling can be performed for a set only if the remaining points have at most double the weight (instead of double the number of points). Therefore, in Step 3(b), we only eliminate points constituting half the remaining weight (instead of half the remaining points). We assign these points to the nearest centers in C and recursively compute *Irred-k-clustering* on the remaining set.

7.2 Analysis and Proof of Correctness

We will show that the weighted algorithm mimics the steps performed by the unweighted algorithm on the unweighted version of the problem, with the exception of certain optimizations that lead to improved running time.

We now prove correctness of the weighted algorithm.

```
**Algorithm Weighted-Irred-$k$-clustering**$(Q, m, k, C, \alpha, \mathrm{Sum})$
**Inputs :** $Q$: Remaining point set, $m$: number of cluster centers yet to
        be found, $k$: total number of clusters,
        $C$: set of $k - m$ cluster centers found so far,
        $\alpha$: approximation factor, *Sum*: the cost of assigning points in $P - Q$
            to the centers in $C$
**Output :** The clustering of the points in $Q$ in $k$ clusters.

1. If $m = 0$
        Assign the points in $Q$ to the nearest centers in $C$.
        Sum = Sum + The clustering cost of $Q$.
        Return the clustering.
2. Use the Weighted Superset Sampling procedure and for each center $c \in \mathtt{WCEN}(Q, \alpha/k, \alpha^{O(1)}/k)$
        Obtain the clustering **Weighted-Irred-$k$-clustering**$(Q, m-1, k, C \cup \{c\}, \alpha, \mathrm{Sum})$.
3. (a) Consider the points in $Q$ in ascending order of distance from $C$.
   (b) Let $U$ be the first $j$ points in this order such that
         $W_U \geq W_Q/2 > W_U - w_{p_j}$, where $p_j$ is the $j^{th}$ point in this sequence
   (c) Assign the points in $U$ to the nearest centers in $C$.
   (d) Sum = Sum + The clustering cost of $U$.
   (e) Compute the clustering **Weighted-Irred-$k$-clustering**$(Q - U, m, k, C, \alpha, \mathrm{Sum})$.
4. Return the clustering which has minimum cost.
```

Figure 7: The weighted irreducible k-clustering algorithm

Theorem 7.2. *Suppose a weighted point set P is (k, α)-irreducible. Then the algorithm* **Weighted-Irred-k-clustering**$(P, k, k, \emptyset, \alpha, 0)$ *returns a solution to the clustering problem $\mathcal{C}(f, k)$ on input P of cost at most $(1 + \alpha)\mathtt{OPT}_k(P)$ with probability γ^k, where γ is a constant.*

Proof. Note that the modifications in Step 2, namely weighted sampling for application of the *Weighted Superset Sampling Procedure* and the *Weighted Random Sampling Procedure* correspond to unweighted sampling for application of the *Superset Sampling Procedure* and the *Random Sampling Procedure* on the unweighted problem. Therefore the weighted algorithm mimics this step of the unweighted algorithm exactly.

In Step 3, as the *Weighted Tightness Property* implies *Tightness Property* in the corresponding unweighted problem, we can remove half the remaining weight in order to get a factor 2 approximation to the set $\bar{P} \cap Q$ as was done in the unweighted version of the problem.

However, a careful look at Step 3(b) seems to suggest that we may be removing extra points in the corresponding unweighted algorithm since we are removing all the points that have the same coordinates as the point that divides Q into two equal sized partitions in order of distances from the known centers. We prove below that we are justified in removing these extra points, in the corresponding unweighted version of the problem.

Let X_p denote the set of (multiset) points in the unweighted problem corresponding to the point p in the weighted problem. Note that $w_p = |X_p|$. Consider X_{p_j} (of Step 3(b)) in the unweighted clustering problem. Then by Observation 3.2, either $X_{p_j} \subseteq S$ or $X_{p_j} \cap S = \phi$. If $X_{p_j} \subseteq S$, then Q is already a factor 2 approximation of $\bar{P} \cap Q$, since $W_U - w_{p_j} < W_Q/2$. Hence, the next center is discovered using random sampling in Step 2 of the algorithm. Otherwise, $X_{p_j} \cap S = \phi$. In this case, clearly by eliminating the whole of X_{p_j} in Step 3(b) of the algorithm, we do not remove any point of $\bar{P} \cap Q$.

Therefore the correctness of the weighted algorithm on a weighted point set follows from the correctness of the corresponding unweighted algorithm when run on the corresponding unweighted instance of the problem. □

We now establish the running time of the weighted algorithm.

Theorem 7.3. *The algorithm* **Weighted-Irred-k-clustering** *when called with parameters* $(P, k, k, \emptyset, \alpha, 0)$ *runs in time* $O(2^{(k/\alpha)^{O(1)}} dn \log^k W)$*, where* $n =$ *number of distinct points in* P *and* $W =$ *total weight of the point set* P*.*

Proof. Let $T(n, m, W)$ be the running time of the weighted algorithm on input $(Q, m, k, C, \alpha, \mathrm{Sum})$ where $n =$ number of distinct points in Q and $W =$ total weight of the point set Q. Note that in application of the *Weighted Random Sampling Procedure* in Step 2, we can perform the weighted random sampling in time $O(\lambda_{\alpha/k} \cdot \frac{k}{\alpha^{O(1)}} \cdot n)$. This can be done by computing the cumulative weights of the weighted points, Q, in an array of size n, then generating a random number in the range 1 to W_Q and picking the corresponding point based on the cumulative weights array. Then we obtain $u(k, \alpha)$ subsets of the sample, where $u(k, \alpha) = O(2^{(\lambda_{\alpha/k} \frac{k}{\alpha})^{O(1)}})$. Computation of the candidate center set from any set S' takes $O(\eta_{\alpha/k} \cdot nd)$ time. Steps 3(a)-(d) take $O(nd)$ time. Also note that in Step 3(b), at least one distinct point is removed. Therefore we get the recurrence

$$T(n, m, W) = O(u(k, \alpha) \cdot \beta_{\alpha/k}) T(n, W, m-1) + T(n-1, W/2, m) + O(u(k, \alpha) \cdot \eta_{\alpha/k} \cdot nd).$$

Let $\lambda_\alpha = O(1/\alpha^{O(1)})$, $\beta_\alpha = O(2^{(1/\alpha)^{O(1)}})$ and $\eta_\alpha = O(2^{(1/\alpha)^{O(1)}})$. Choose $c = O(2^{(k/\alpha)^\gamma})$ to be large enough, for a suitable constant γ, such that

$$T(n, m, W) \le c \cdot T(n, m-1, W) + T(n-1, m, W/2) + c \cdot nd.$$

We claim that $T(n, m, W) \le c^m \cdot 2^{3m^2} \cdot nd \cdot \log^m W$. The proof is by induction. We show the inductive step here. Suppose that the claim holds for $T(n', m', W')\ \forall n' < n, \forall m', \forall W'$, it holds for $T(n', m', W')\ \forall n', \forall m' < m, \forall W'$ and it holds for $T(n', m', W')\ \forall n', \forall m', \forall W' < W$. Then, we are required to show that

$$c^m \cdot 2^{3m^2} \cdot nd \cdot \log^m W \ge c \cdot c^{m-1} \cdot 2^{3(m-1)^2} \cdot nd \cdot \log^{m-1} W + c^m \cdot 2^{3m^2} \cdot (n-1)d \cdot (\log W - 1)^m + c \cdot nd.$$

For this, it suffices to show that

$$2^{3m^2} \log^m W \ge 2^{3(m-1)^2} \log^{m-1} W + 2^{3m^2-1} (\log W - 1)^m + 1.$$

We know that

$$a^k \ge a^{k-1} + (a-1)^k$$

(follows from the identity $a^k - b^k = (a-b)(a^{k-1} + a^{k-2}b + ... + b^{k-1})$ by setting $b = a - 1$). Therefore, we get that

$$\begin{array}{ll} & 2^{3m^2} \log^m W \ge 2^{3m^2} \log^{m-1} (W) + 2^{3m^2} (\log W - 1)^m \\ \ge & 2^{3(m-1)^2} \log^{m-1} W + 2^{3m^2-1} (\log W - 1)^m + 1 \qquad \text{for } m \ge 1 \end{array}$$

It follows that $T(n, k, W)$ is $O(2^{(k/\alpha)^{O(1)}} dn \log^k W)$ when $\lambda_\alpha = O(1/\alpha^{O(1)})$, $\beta_\alpha = O(2^{(1/\alpha)^{O(1)}})$ and $\eta_\alpha = O(2^{(1/\alpha)^{O(1)}})$. □

Using these theorems, we get the final result for the weighted problem.

Theorem 7.4. *For a clustering problem satisfying the* Closeness Property*,* Weighted Tightness Property *and for which there exists a* Weighted Random Sampling Procedure*, a* $(1+\varepsilon)$*-approximate solution for a weighted point set* P *in* $\Re^d$ *can be found in time* $O(2^{(k/\alpha)^{O(1)}} dn \log^k W)$*, with constant probability.*

Proof. The proof is along the same lines as Theorem 4.7 based on Theorems 7.2 and 7.3. □

The following corollary follows from our arguments for the extensions to the general properties for weighted clustering problems.

Corollary 7.5. *Given a point set* P *of* n *points in* $\Re^d$ *with total weight* W*,* $(1+\varepsilon)$*-approximate solutions to the weighted* k*-means clustering, weighted* k*-median clustering and the weighted discrete* k*-means clustering problems can be found in time* $O(2^{(k/\alpha)^{O(1)}} dn \log^k W)$*, with constant probability.*

8 Conclusions

We presented a generic framework that solves a large class of clustering problems satisfying certain properties in time linear in the size of the input for fixed values of k and d. We showed that the k-means clustering, the k-median clustering and the discrete k-means clustering, all satisfy the given properties and therefore admit of the linear time algorithms.

It remains open whether or not the discrete k-median clustering problem belongs to this class of clustering problems. In particular, the existence of a *Random Sampling procedure* for the discrete k-median clustering problem is not yet known.

We illustrate below by means of an example, that the strategy used in the discrete k-means clustering, of first finding an approximate (continuous) center and then selecting the closest discrete point to this approximate center does not work for the case of the discrete k-median clustering problem. More precisely, we show that for a point set P, the discrete point closest to the non-discrete 1-median may have cost arbitrary larger than the cost of the optimal discrete 1-median.

Consider a set P_1 consisting of $3n$ points placed on the vertices of an equilateral triangle with n points on each vertex. The coordinates of the vertices of the triangle are $(0, 2a/\sqrt{3})$, $(a, -a/\sqrt{3})$ and $(-a, -a/\sqrt{3})$. Consider another set P_2 consisting of 3 points placed on the vertices of another equilateral triangle with 1 point on each vertex. The coordinates of this triangle are $(0, -2a'/\sqrt{3})$, $(a', a'/\sqrt{3})$ and $(-a', a'/\sqrt{3})$ (see figure 8). The optimal 1-median for both P_1 and P_2 happens to be the origin $(0,0)$. Therefore the optimal 1-median for $P = P_1 \cup P_2$ is also the origin. Let $a' = a - \delta$, where δ is an infinitely small number. For ease of computation, we will consider $a' = a$. The reason for having a different value of a' is only to ensure that the points of P_2 are closer to the origin than the points of P_1. Now, the optimal (non-discrete) 1-median cost of P is $2\sqrt{3}a(n+1)$. The cost of the 1-median solution when the center is at a vertex corresponding to P_1 is $4an + 8a/\sqrt{3}$ and the cost of the 1-median solution when the center is at a vertex corresponding to P_2 is $4a + 8an/\sqrt{3}$. Note that the discrete points of P_2 are closer to the optimal (non-discrete) 1-median than the discrete points of P_1. Therefore if we pick the discrete point closest to the non-discrete 1-median, we can get a solution that has cost arbitrarily larger than the cost of the optimal discrete 1-median cost.

Note that the optimal 1-mean of the points set also happens to be the origin. However, the cost of the 1-means solution when we pick any center from P_1 is $4(2n+1)a^2 + 4(a-\delta)^2$ whereas for a center from P_2 it is $4a^2n + 4(n+2)(a-\delta)^2$. Thus, the cost is always lower for a point that is closer to the optimal 1-mean of the point set.

Thus, the strategy used in approximating the discrete 1-mean center, of approximating the (non-discrete) 1-center and then selecting the discrete point closest to it, does not work for the 1-median problem.

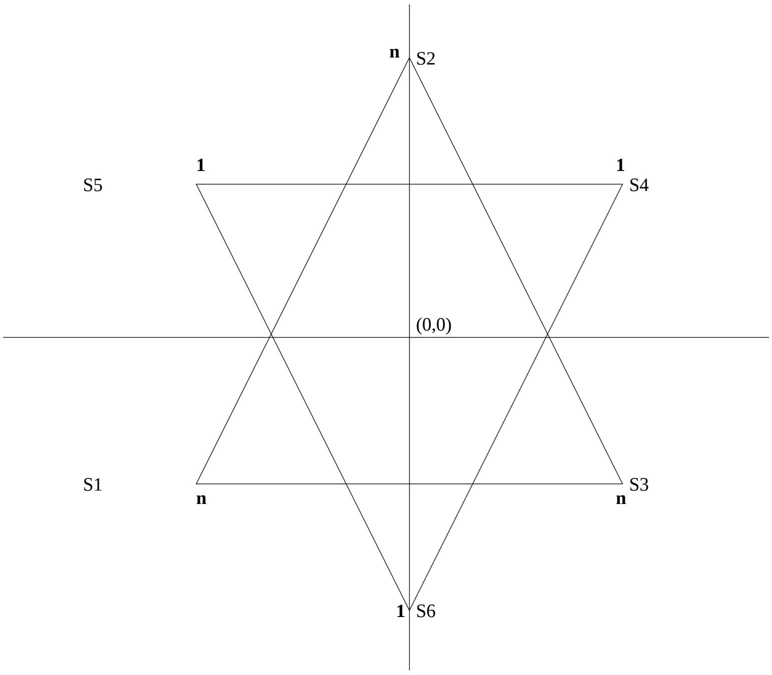

Figure 8: Bad example for discrete 1-median clustering

In a recent development, Ke Chen [6] showed that we can obtain small coresets for k-median clustering in metric spaces as well as in Euclidean spaces. Specifically, in $\Re^d$, the coresets are of size with only polynomial dependence in d. Ke Chen shows that combining these coresets with our result leads to a $(1+\varepsilon)$-approximation algorithm for k-median clustering in $\Re^d$, with running time $O(ndk + 2^{(k/\varepsilon)^{O(1)}} d^2 n^\sigma)$, for any $\sigma > 0$. An interesting open problem is whether there exist coresets for the k-median or k-means clustering problems of size independent of n and having only polynomial dependence in d.

Another interesting open problem is to find a PTAS for the k-means clustering problem, even for fixed dimensions.

We also leave open the problem of finding linear time clustering algorithm with running time $O(2^{O(k/\varepsilon)} nd)$. We would also like to note that such an algorithm was recently given by Ostrovsky et. al. [20], but their analysis requires the input instances to satisfy a crucial "separation property".

References

[1] S. Arora, *Polynomial time approximation schemes for Euclidean TSP and other geometric problems*, Journal of the ACM, 1996, pp. 2-11.

[2] S. Arora, P. Raghavan and S. Rao, *Polynomial time approximation schemes for the Euclidean k-median problem*, 30th Annual Symposium on Theory of Computing, 1998.

[3] M. Badoiu, S. Har-Peled and P. Indyk, *Approximate clustering via core-sets*, 34th Annual Symposium on Theory of Computing 2002, pp. 250-257.

[4] M. Bern and D. Eppstein, *Approximation algorithms for geometric problems*, D. S. Hauchbaum, editor, Approximating algorithms for NP-Hard problems. PWS Publishing Company, 1997.

[5] A. Broder, S. Glassman, M. Manasse and G. Zweig, *Syntactic clustering of the Web*, 6th Int'l World Wide Web Conf (WWW), 1997, pp. 391-404.

[6] Ke Chen, On k-median clustering in high dimensions. In *17th Annual Symposium on Discrete Algorithms*, 2006, pages 1177–1185.

[7] S. Dasgupta, *The hardness of k-means clustering*, Technical Report CS2007-0890, University of California, San Diego, 2007.

[8] W. F. de la Vega, M. Karpinski, C. Kenyon and Y. Rabani, *Approximation schemes for clustering problems*, 35th Annual Symposium on Theory of Computing, 2003, pp. 50-58.

[9] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas and R. A. Harshman, *Indexing by latent semantic analysis*, Journal of the Society for Information Science, 41(6):391-407, 1990.

[10] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, Wiley-Interscience, New York, 2nd edition, 2001.

[11] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic and W. Equitz, *Efficient and effective querying by image content*, Journal of Intelligent Information Systems, 3(3):231-262, 1994.

[12] V. Guruswami and P. Indyk, *Embeddings and non-approximability of geometric problems*, 14th Annual Symposium on Discrete Algorithms, 2003, pp. 537-538.

[13] S. Har-Peled and S. Mazumdar, *Coresets for k-Means and k-Median Clustering and their Applications*, 36th Annual Symposium on Theory of Computing, 2004, pp. 291-300.

[14] M. Inaba, N. Katoh and H. Imai, *Applications of Weighted Voronoi Diagrams and Randomization to Variance-Based k-Clustering*, 10th Annual ACM Symposium on Computational Geometry, 1994, pp. 332-339.

[15] P. Indyk, *High Dimensional Computational Geometry*, Ph.D. Thesis, Department of Computer Science, Stanford University, September 2004.

[16] S. Kolliopoulos and S. Rao, *A nearly linear time approximation scheme for the Euclidean k-medians problem*, SIAM Journal of Computing, 37(3), 2007, pp. 757-782.

[17] A. Kumar, Y. Sabharwal and S. Sen, *A simple linear time $(1+\varepsilon)$-approximation algorithm for k-means clustering in any dimensions*, 45th Annual IEEE Symposium on Foundations of Computer Science 2004, pp. 454-462.

[18] A. Kumar, Y. Sabharwal and S. Sen, *Linear Time Algorithms for Clustering Problems in Any Dimensions.*, International Colloquium on Automata, Languages and Programming, 2005, pp. 1374-1385.

[19] J. Matoušek, *On approximate geometric k-clustering*, Discrete and Computational Geometry, 24, 2000, pp. 61-84.

[20] R. Ostrovsky, Y. Rabani, L. J. Schulman and C. Swamy, *The Effectiveness of Lloyd-Type Methods for the k-Means Problem*, 47th Annual IEEE Symposium on Foundations of Computer Science 2006, pp. 165–176.

[21] Yogish Sabharwal and Sandeep Sen. A linear time algorithm for approximate 2-means clustering. *Comput. Geom.*, 32(2):159–172, 2005.

[22] M. J. Swain and D. H. Ballard, *Color indexing*, International Journal of Computer Vision, 7:11-32, 1991.