Machine Learning, 56, 61–87, 2004

A k-Median Algorithm with Running Time Independent of Data Size

ADAM MEYERSON* awm@cs.ucla.edu
Department of Computer Science, University of California, Los Angeles

LIADAN O'CALLAGHAN† loc@cs.stanford.edu
SERGE PLOTKIN‡ plotkin@theory.stanford.edu
Department of Computer Science, Stanford University

Editor: Shai Ben-David

Abstract. We give a sampling-based algorithm for the k-Median problem, with running time $O(k(\frac{k^2}{\epsilon}\log k)^2 \log(\frac{k}{\epsilon}\log k))$, where k is the desired number of clusters and ϵ is a confidence parameter. This is the first k-Median algorithm with fully polynomial running time that is independent of n, the size of the data set. It gives a solution that is, with high probability, an $O(1)$-approximation, if each cluster in some optimal solution has $\Omega(\frac{n\epsilon}{k})$ points. We also give weakly-polynomial-time algorithms for this problem and a relaxed version of k-Median in which a small fraction of outliers can be excluded. We give near-matching lower bounds showing that this assumption about cluster size is necessary. We also present a related algorithm for finding a clustering that excludes a small number of outliers.

Keywords: clustering, sampling, sublinear

1. Introduction

Clustering, or grouping data into representative groups, is a tool commonly used in data analysis, and can often make the manipulation of large data sets simpler. For example, in a large customer database a cluster might represent customers with similar characteristics. In a vision application, a cluster might be the group of pixels that form a single object in the viewed image. In the context of the web, a cluster could be a set of web pages with similar content. In census data a cluster might represent a population center. In each of these applications, computing a clustering makes the data easier to manipulate, and easier for a human to parse. For example, finding clusters in a customer database might facilitate targeted advertising or allow compression of the data set to a summary of large customer segments. Clustering algorithms have been the subject of a great deal of research (Bradley, Fayyad, & Reina, 1998; Charikar et al., 1997; Duda & Hart, 1972; Feder &

*Research supported by ARO Grant DAAG55-98-1-0170, NSF Grant CCR-0122581, and the University of California.
†Research supported in part by NSF Grants EIA-0137761 and IIS-0118173 and by an ARCS scholarship.
‡Research supported by NSF Grant CCR-0113217.

Greene, 1988; Gonzalez, 1985; Hartigan, 1975; Marroquin & Girosi, 1993; Pelleg & Moore, 1999; Zhang, Ramakrishnan, & Livny, 1996).

Data sets in the applications mentioned are growing at an exponential rate. The web is the most striking example of this expansion, but exponential growth has been observed in many other domains as well. As the data sets grow, the need for fast clustering algorithms becomes more and more apparent. Most of the theoretical work in clustering has focused on producing polynomial-time approximation algorithms for NP-Hard clustering problems. While these algorithms work very well for moderately sized data sets, the pattern of exponential growth suggests that even a *linear*-time algorithm may soon be too slow in practice, and so we consider the possibility of *sublinear*-time clustering algorithms that maintain provable approximation bounds. We cannot read the entire data set within such a running time, and so we must make several reasonable assumptions about the nature of the data (essentially, that the best clustering cannot be greatly influenced by the presence of a relatively small number of points). We show that these assumptions are necessary and that our running times are within a polylogarithmic factor of the best possible. Since without reading the entire data set we run the (small) risk that the points we do examine will not be representative of the entire data set, we will only give a high-probability guarantee on the quality of the clustering we find.

Different notions of clustering. The general term "clustering" covers many different types of problems. All involve a subdivision of a data set into groups of similar elements, but there are many measures of similarity, many ways of measuring how well the subdivision reflects the character of the data, and even various concepts of subdivision. Many clustering algorithms assume that the points exist in some metric space, often real space under the Euclidean (or another $\mathcal{L}_p$) norm; the similarity is then defined simply by the distance metric (Charikar & Guha, 1999; Meyerson, 2001). In the field of document clustering, the "cosine distance," which is nearly a metric (it obeys a "relaxed triangle inequality" although not the triangle inequality itself), is often used (Steinbach, Karypis, & Kumar, 2000). For some clustering problems, the distance function need not obey the triangle inequality, be symmetric, be real-valued, or even be defined on all pairs of points (Bansal, Blum, & Chawla, 2002; Lin & Vitter, 1992; Young, 2000).

Just as there are different notions of similarity, there are different notions of what clusterings are legal, and what constitutes a good clustering. Under some definitions of clustering, a point can belong fractionally to each of several groups in the subdivision; this type of clustering is sometimes called "soft," as distinct from a "hard" clustering, in which a partition is imposed on the data so each point is assigned to exactly one group. For example, if the data are considered to be drawn from a distribution, such as a mixture of k Gaussians, we may use an EM algorithm to learn the distribution (Dempster, Laird, & Rubin, 1984; Redner & Walker, 1984); in this case, for every data point x and each of the learned Gaussians $F_i = N(\mu_i, \sigma_i)$, there will be some probability (possibly very small) that F_i generated x. On the other hand, k-Means (Bradley, Fayyad, & Reina, 1998) is an example of an algorithm that makes a "hard" assignment of points to clusters. k-Means aims to find k centroids such that, if every point in the data set is "assigned" to its closest centroid, the sum of assignment distances is minimized, where, typically, the points come from a real space $\mathbb{R}^d$, and the

assignment distance of a point is defined as the squared Euclidean distance from the point to its nearest centroid.

In most clustering problems, the quality of a particular solution is given by some (usually NP-Hard) objective function defined over the set of possible solutions. For example, given a set of n points in $\mathbb{R}^d$, one may wish to find the k Gaussians that were most likely to have generated this set (Dasgupta, 1999). In the case of the *k-Center* problem, the goal is to choose k members of the data set, called "centers," so as to minimize the maximum assignment distance (where assignment distance is, as before, the distance from a data point to its nearest center) (Hochbaum & Shmoys, 1985). The *Correlation Clustering* problem is, given a complete graph on n vertices (which can be thought of as data points), each of whose edges is labelled "+" or "−," find a partition of the vertices into subsets, to minimize the number of "+" edges between subsets, plus the number of "−" edges within subsets (Bansal, Blum, & Chawla, 2002). An exception to the general rule that clustering is NP-Hard is the following problem: given n points with a symmetric but possibly non-metric distance function, divide the points into clusters so as to maximize $D - d$, where D is the minimum inter-cluster distance and d is the maximum cluster diameter. Hierarchical agglomerative clustering (HAC) (Pitt & Reinke, 1988) finds the optimal solution to this problem in polynomial time.

The k-median problem. In this paper, the *Metric k-Median* problem, one of the most popular and best studied clustering measures, will be discussed. An instance of the Metric k-Median problem (hereafter, simply "k-Median") consists of a data set N, a distance metric, and an integer k; we are to choose k members of N as "medians," and assign each member of N to its closest median. As before, the assignment distance of a point $x \in N$ is the distance from x to the median to which it is assigned, and the k-Median objective function, which is to be minimized, is the sum of assignment distances. This problem is related to the one that k-Means attempts to optimize (although k-Means finds a *locally* optimal solution, not necessarily a *global* optimum).

k-Median is NP-Hard: there is no polynomial-time algorithm that is guaranteed to find the optimal solution for all instances of k-Median, unless P = NP. Despite the fact that it is NP-Hard, there are polynomial-time algorithms guaranteed to find approximately optimal solutions to the problem. Given an NP-Hard optimization problem Π, let $\mathcal{I}$ denote the set of instances of Π, and let $C : \mathcal{I} \to \mathbb{R}^+ \cup \{0\}$ denote the objective function to be optimized. For convenience, assume that C is to be minimized; if C is to be maximized, the definitions are analogous. For all $I \in \mathcal{I}$, let S_I^* denote an optimal solution to I, and let $C_I^* = C(S_I^*)$ denote the cost of this solution. As Π is NP-Hard, there is no polynomial-time algorithm that guarantees to find S_I^* for each $I \in \mathcal{I}$, unless P = NP. An α-approximation algorithm A for Π is a polynomial-time algorithm that guarantees that, for any $I \in \mathcal{I}$, $C(A(I)) \leq \alpha C(S_I^*)$, where $A(I)$ is the solution found by A for I. That is, the cost of the solution found by A for I is guaranteed to be no more than α times the cost of the cheapest solution for I. (If C is to be maximized rather than minimized, then an α-approximation algorithm A, for $\alpha < 1$, would guarantee that $C(A(I)) \geq \alpha C(S_I^*)$.) In general, α may depend on the size of the problem instance (for example, α could be $\log n$); for the k-Median problem, there exist α-approximation algorithms for constant α.

It is NP-Hard to approximate k-Median within a factor of $1 + 2/e$ (Jain et al., to appear), i.e., unless P = NP, there is no α-approximation algorithm for k-Median with $\alpha < 1 + 2/e$. Nevertheless, the abundance of work on the k-Median problem has provided a number of constant-factor approximation algorithms (Charikar & Guha, 1999; Charikar et al., 2002; Guha et al., 2000; Jain & Vazirani, 1999). In the geometric setting, Arora, Raghavan, and Rao (1998) gave a polynomial-time approximation scheme (PTAS) for the k-Median problem in low dimensions; a PTAS is a family of algorithms that includes, for each fixed $\epsilon > 0$, a polynomial-time $(1 + \epsilon)$-approximation algorithm. (In general, the running time may be exponential in $1/\epsilon$.)

Like many optimization problems, k-Median can be expressed as an integer program, as follows. If the data set is denoted N, for each point $x \in N$ there is a variable $m(x)$ that is set to 1 if x is chosen as a median, and 0 otherwise. For each pair (x, y) of points there is a variable $a(x, y)$ that is set to 1 if x is assigned to y, and 0 otherwise. The constraints are:

$$\begin{aligned} \forall\, x \in N, m(x) &\in \{0, 1\} \\ \forall\, x, y \in N, a(x, y) &\in \{0, 1\} \\ \forall\, x \in N, \sum_{y \in N} a(x, y) &\geq 1 \\ \forall\, x, y \in N, a(x, y) &\leq m(y) \\ \sum_{x \in N} m(x) &\leq k \end{aligned}$$

so each point must be assigned to some median, and a point y must be a median if some point is assigned to it. The objective function is $\sum_{x \neq y \in N} a(x, y) \cdot d(x, y)$, where $d(x, y)$ is the distance from x to y.

Charikar et al. (2002) gave the first constant-factor approximation algorithm for the k-Median problem on general metric spaces. The algorithm employs LP-rounding: it optimally solves an LP, a linear relaxation of the above integer program, in which each $m(x)$ and each $a(x, y)$ can take on any value in $[0, 1]$. Then, if the solution found is not a feasible k-Median solution (e.g., if there are more than k points with non-zero $m(x)$), the $m(x)$ and $a(x, y)$ values are rounded so that a legal k-Median solution is obtained. The authors showed that their method of rounding produced an approximately optimal k-Median solution.

Jain and Vazirani (1999), gave a primal-dual approximation algorithm. The Jain-Vazirani algorithm does not solve an LP, but uses LP duality to find lower bounds on the cost of the optimal k-Median solution. Charikar and Guha (1999) subsequently improved the approximation ratio. The best approximation known is $3 + \epsilon$, given by Arya et al. (2001). All these algorithms require $\Omega(n^2)$ running time to compute an approximate k-Median solution on n points. Guha et al. (2000) gave a divide-and-conquer algorithm that improves the running time to $\tilde{O}(nk)$ with some loss in the approximation ratio (here, $\tilde{O}$ indicates hidden polylogarithmic factors). They also showed a $\Omega(nk)$ lower bound on the running time of any deterministic k-Median approximation algorithm. Thorup (2001) gave an $O(1)$-approximation algorithm for k-Median on sparse graph metrics (a sparse graph metric is a metric space describable by a sparse graph that represents points as vertices, and inter-point distances as the shortest-path distances between vertices); the algorithm runs in $\tilde{O}(m)$ time,

where m is the number of edges in the graph. Indyk (1999) gave a sampling-based bicriteria algorithm with running time $O(nk)$, which returns a solution with βk medians of cost at most α times the cost of the optimal solution with exactly k medians, where α and β are both greater than 1. Mettu and Plaxton (2002) gave a sampling-based algorithm that, with high probability, produces a constant-factor approximation; this algorithm assumes that the diameter of the point set does not exceed the smallest distance between distinct points by a factor of more than $2^{O(n/\log(n/k))}$. The running time of this algorithm is $O(nk)$, where k is $\Omega(\log n)$, in the case of the standard k-Median problem; in the case of the *weighted* k-Median problem, in which each point may be associated with a non-unit weight, the algorithm runs in time $O(n(k+\log n)+k^2\log^2 n)$, which is $O(nk)$ when k is $\Omega(\log n)$ and $O(n/\log^2 n)$. Mettu and Plaxton also showed that *any* randomized k-Median algorithm guaranteeing an approximation ratio of α with nonnegligible probability (say, at least $1/100$) requires $\Omega(nk)$ running time, under one assumption: that the ratio between the diameter of the point set and the smallest distance between two distinct points in the set, is at least $\alpha\beta n/k$, where β is a sufficiently large constant.

Running in sublinear time. These algorithms produce good approximations but all have running times that are superlinear in the number of data points. When the data set is very large, even linear time can be prohibitive. We therefore turn to sampling algorithms, which select only a subset of the data to use in computing a clustering. Mishra, Oblinger, and Pitt (2001) gave a sampling-based algorithm that, with probability at least $1-\delta$, produces a k-Median clustering of cost at most $2\alpha\text{OPT}+\epsilon n$, where OPT is the optimum k-Median cost, α is the approximation ratio guaranteed by the k-Median approximation algorithm used as a subroutine, and ϵ is an input parameter. The running time depends on $(M/\epsilon)^2$, where M is the diameter of the data set, and on $\log(n/\delta)$. In the special case of Euclidean space, the dependence of the running time on $\log(n/\delta)$ is avoided, and the clustering cost is at most $\alpha\text{OPT}+\epsilon n$ (with probability $1-\delta$). As the only previous algorithm with running time independent of n, this algorithm of Mishra et al. is the existing result most comparable to the results presented here. The running times of the algorithms we present are independent of n regardless of the metric, but two of the algorithms we will present also have weakly polynomial running time, due to a dependence on $\log M$, and the other has fully polynomial running time. Mishra et al. use results from statistics to bound the difference between the true cost and sample cost of the k-Median objective function, whereas, in this paper, a purely multiplicative guarantee is given using Chernoff Bounds. Alon et al. (2000) presented a sampling-based property-testing algorithm for the related k-Center problem, in which the aim is to minimize the maximum assignment distance, rather than the sum of assignment distances. Sampling has also proved to be a successful technique in practice for various clustering problems (Guha, Rastogi, & Shim, 1998; Joze-Hkajavi & Salem, 1998; Palmer & Faloutsos, 2000).

We immediately recognize several challenges that will arise if sampling is to be used. First, we must be able to sample the data uniformly at random. Otherwise, an adversary could generate a sample that left out significant portions of the data set and therefore lose the chance of producing a good clustering. Second, it is possible for a small number of data points to have a very significant effect on the clustering. For example, suppose there

is a very small (say $o(\sqrt{n})$) set S of points that are extremely far from all the other points. If any of these points are assigned to groups containing points *outside* of S, the clustering quality will be poor. Unfortunately, a small uniform sample will be very unlikely to include *even one* point from this set, and so it will be impossible to produce a good solution for this instance without changing the sample size or sampling method.[1] Third, this approach precludes outputting an explicit assignment of each point to a group, since performing this assignment will require at least linear time.

In order to deal with these problems, we will make several assumptions. First, we will assume that we can select a data point chosen uniformly at random in constant time. In other words, we assume that in constant time we can read the identity of a point from the point set, regardless of the size of the key giving the identity. We can think of the data as being stored on some random-access memory device, so that we can select a random location and access that data point in effectively constant time. For example, the data set could be a set of documents, distributed over a network of machines that store the documents in RAM. Given an index of a set of webpages stored in such a distributed network, for example, one could pick a random sample of the document set, and then request them, in time proportional to the size of the sample. We observe that if we do not have random access, and the data are ordered adversarially on a device that must be read in-order, then the data are effectively streaming and the best we can hope for is linear time, even if we perform no operations outside of a scan of the point set. Second, we assume that each of the clusters in the *optimum* solution, against which we compare our clustering, is large. The "average" size of a cluster would be n/k, and we will assume each cluster to have $\Omega(n/k)$ size. Equally well, we can allow our solution to disregard a small fraction of "outlying" points. Third, we will output k medians that are "representative" of the data set. We consider this output sufficient without assigning each point explicitly to its nearest median (indeed, if the data set is large, it is probably better to output the median set rather than an explicit assignment of every point).

The assumption that some optimal solution has relatively "large" clusters, or that some portion of the data set can be regarded as outlying noise has been treated earlier in clustering literature. For example, Dasgupta (1999) assumes that the data points are generated from k Gaussians each with weight $\Omega(1/k)$, and there are various papers on clustering with outliers (Alon et al., 2000; Charikar et al., 2001). The assumption is motivated by the fact that the goal of our algorithms is to produce several broad classes that are representative of most data points, and that clustering a small number of outliers is not as important. Often the data set itself has "noise" due to observational or recording errors; this noise may lead to a small number of extreme outliers which should in fact be disregarded.

Overview. In Section 2, we will mention a few results from previous work that we will refer to throughout the paper. In Section 3 we will summarize our results. In Section 4 we will present our first k-Median algorithm. This algorithm assumes that in some optimal k-Median solution, the clusters are of at least a given size. In Section 5 we will present a simpler algorithm for the same problem, and in Section 6 we will present a related algorithm for the version of k-Median in which the solution may exclude a small number of outliers, but there is no assumption on the sizes of clusters in optimal solutions. Finally, in Section 7

we show some lower bounds that prove that, under certain assumptions, our sample sizes are as small as possible.

2. Preliminaries

In this section we will establish a few facts that we will use throughout the paper. First we will point out a simple property of the k-Median problem, and then we will give the running times of two k-Median approximation algorithms that we will use as subroutines in the algorithms we will present.

Restricting the median set. For completeness, we mention a convenient fact that has often been observed in related literature.

Fact 2.1. *Let N be a set of points, with $S \subseteq N$. Let k be an integer with $0 \leq k \leq n$. Let $K \subseteq S$ be the k-element subset of S minimizing $\sum_{x \in S} d(x, K)$, where $d(\cdot, \cdot)$ is the distance function on N, and $d(x, K)$ denotes $\min_{m \in K} d(x, m)$. Then if K' is a k-element subset of N, $\sum_{x \in S} d(x, K) \leq 2 \sum_{x \in S} d(x, K')$.*

Proof: We will give an example $K'' \subseteq S$ with $|K''| = k$ such that $\sum_{x \in S} d(x, K'') \leq 2 \sum_{x \in S} d(x, K')$; by the optimality of K, the fact will follow. Let K'' be chosen in the following way: for each element p of K', K'' will contain the closest member, say $q(p)$ of S, i.e., $q(p) = \text{argmin}_{r \in S} d(p, r)$.

Consider an $x \in S$, and let m be the closest member of K' to x, so that $d(x, K') = d(x, m)$. Then $d(x, K'') \leq d(x, q(m)) \leq d(x, m) + d(m, q(m))$, which, by definition of $q(m)$, is at most $2d(m, x)$. Therefore, $d(x, K'') \leq 2d(x, m) = 2d(x, K')$. The fact follows. □

In particular, this fact implies that if we solve an instance of k-Median on a sample S of a point set N, insisting that the medians be members of S, rather than arbitary members of N, costs only a factor of two in the cost of the optimal solution.

Subroutine running times. Next we review the running times of two k-Median approximation algorithms we will be using as subroutines for our algorithms. The algorithm of Arya et al. (2001) guarantees a $(3 + 2/p)$-approximation to the k-Median problem. It uses local search, meaning it begins with an initial solution and then iteratively refines this solution by means of local improvements, each of which, in the case of this algorithm, lowers the solution cost by a factor of $(1 - \gamma/k)$ or more (where γ is a constant). If the initial solution is an $O(n)$-approximation to the optimal k-Median solution, where n is the number of points in the k-Median instance, then $O(k \log n)$ improvement steps are sufficient to guarantee that the algorithm will arrive at a constant-factor approximation. An example of such an $O(n)$-approximation is the $O(nk)$-time farthest-point k-Center algorithm of Gonzalez (1985). Each improvement step takes $O((nk)^{p+1})$, because at most $O((nk)^p)$ possible improvements will need to be evaluated, at a cost of $O(nk)$ per evaluation, before one is chosen and effected (at which time the cost is lowered and another round of searching for an improvement step begins). Therefore, the running time of the entire algorithm is

dominated by the time spent in the local-improvement stage (and not in the initial-solution stage), so the running time is $O(k(nk)^{p+1} \log n)$ for an n-point instance.

The algorithm of Charikar et al. (2001) will be used as a subroutine in two of the algorithms presented in this paper. This algorithm is a constant-factor approximation algorithm for the "robust" k-Median problem, meaning that it gives the following guarantee: Given an instance of k-Median on n points, if there is a choice K^* of k medians such that the sum of the lowest $(1-\delta)n$ assignment distances to K^* is C, then this algorithm will find a set K of k medians such that the sum of the lowest $(1-\kappa_1\delta)n$ assignment distances to K is at most $\kappa_2 C$. This algorithm, like that of Jain and Vazirani (1999), uses LP duality. The linear program used by Charikar et al. is a relaxation of the basic k-Median LP, in which not every point need be assigned to a median, but an additional cost is charged per unassigned point. The running time of this algorithm is $O(n^2 \log n \log M)$ where M is the diameter of the point set. The second and third algorithms that we will present, which use this algorithm as a subroutine, will therefore have weakly polynomial running time.

3. Our results

Our first k-Median algorithm will use the algorithm of Arya et al. (2001) as a subroutine. The running time of our algorithm will be $O(k(\frac{k^2}{\epsilon} \log k)^{p+1} \log(\frac{k}{\epsilon} \log k))$, where ϵ is a real number between 0 and 1 such that, in some optimum solution, each cluster contains at least $\frac{\epsilon n}{k}$ data points. This running time is *independent* of n, the number of data points. Our algorithm guarantees that, with high probability, the cost of the solution we produce is within $O(1)$ of the cost of some optimum solution in which each cluster contains at least $\frac{\epsilon n}{k}$ points. Our methods also provide a technique for evaluating candidate clusterings in sublinear time, under the same assumptions about the nature of the data set.

Next we give a simpler k-Median algorithm that uses the algorithm of Charikar et al. (2001) as a subroutine. This algorithm runs in $O((\frac{k}{\epsilon} \log \frac{k}{\epsilon})^2 \log(\frac{k}{\epsilon} \log \frac{k}{\epsilon}) \log M)$ time and will, with high probability, produce a k-Median solution whose cost is at most a constant times the optimal solution cost. Here, M is the diameter of the point set; we note that this algorithm is only weakly polynomial.

Our third algorithm is nearly identical to the first, but works for the case that it is acceptable to leave a small fraction of the points out of the clustering solution. In particular, if the optimal k-Median solution that assigns $(1-\phi)n$ points has a cost of C (i.e., on the $(1-\phi)n$ points of lowest assignment distance), our algorithm will, with high probability, produce a median set that, if it can expand the fraction of points excluded by a constant factor, will have a total assignment cost that is $O(1)$ times C. The running time of this algorithm is $O((\frac{k}{\phi}[\log k^3 + \log \frac{k^2}{\phi}])^2 \log(\frac{k}{\phi}[\log k^3 + \log \frac{k^2}{\phi}]) \log M)$.

We also give near-matching lower bounds relating the size of a uniform sample to the smallest minimum cluster size that is allowable if a constant-factor approximation is to be produced; in particular, we show that our sample size is at most $(\log(k/\epsilon))$ times as large as the smallest uniform sample we would need to guarantee a constant-factor approximation with non-negligible probability. We note that Thorup (2001) and Mettu and Plaxton (2002) show how to sample several times, using information from previous samples so that smaller samples can be taken later from a reduced point set.

4. k-Median algorithm

The first k-Median algorithm we will present consists of two steps. In the first step, described in Section 4.2, we generate several candidate median sets. To generate each set, we draw a sample uniformly with replacement from the data and compute an approximately-optimal clustering for the sample using the algorithm of Arya et al. (2001). We show that, with constant probability, the cost of a candidate set is within $O(1)$ of that of the best k-Median solution. By increasing the constants in the approximation factor and sample size, we could produce an algorithm that would with high probability produce a good clustering; the required increases would be unacceptably large, however. We can obtain better results by repeatedly sampling and clustering, and taking the best of our candidate median sets.

Determining the best set of candidate medians is a nontrivial problem. The straightforward approach to testing would take linear time. Thus, the second part of our algorithm is a method for determining the best of several candidate clusterings. We show, given a number of candidates, that we are able to select one whose cost is within $O(1)$ of the cost of the best candidate. This process can be performed, again using sampling, in time independent of n. The basic technique is to draw a number of random samples and measure the quality of each set of candidate medians on each random sample. We will return the set of medians that have the best observed quality (lowest minimum value over all samples). This evaluation of candidates is discussed in Section 4.3. As mentioned earlier, both steps of our algorithm require that the clusters in some optimal solution be large, and that we can select a data point uniformly from the input in constant time.

Combining the steps outlined above gives us an algorithm that produces constant-factor approximations to the best k-Median solution, in running time independent of n, the size of the data set (under certain assumptions about the nature of the input). We present this overall result in Section 4.4, and near-matching lower bounds in Section 7.

Some data sets contain "noise" or "outliers" that make the data inherently difficult to cluster, in the following sense: the best possible k-Median solution is "much more expensive" than the best solution that excludes a small number of points (i.e., does not group them and does not pay their assignment costs). Using a natural extension of the algorithms already described, we can produce a constant-factor approximation to the best clustering that excludes ϵn outliers, provided our algorithm is allowed to exclude a slightly higher ($O(1)$ times ϵn) number of outliers. The approach used is identical, except that we need to use the algorithm of Charikar et al. (2001) instead of that of Arya et al. (2001) to cluster the sample. We discuss this extension in Section 6.

4.1. Notation

Before giving the full description and analysis of the algorithm, we first introduce some notation.

Definition 1. N is the data set, with $n = |N|$.

Definition 2. S and T are samples, with $s = |S|$ and $t = |T|$.

Definition 3. k is the desired number of medians; K^* represents an optimum median set, and K will represent a candidate set of medians. Here $|K| = |K^*| = k$.

Definition 4. ϵ is such that in some optimum solution every cluster contains at least $\frac{\epsilon n}{k}$ points.

Definition 5. We will use $d(\cdot, \cdot)$ to denote the distance function. For points x and y, $d(x, y)$ will denote the distance between x and y. For a point x and a set T of points, we will use $d(x, T)$ to denote the distance from x to the set T, meaning $\min_{y \in T} d(x, y)$.

Definition 6. $f^N(K)$ represents the cost of the median set K as a clustering of the data set N. That is, $f^N(K) = \sum_{x \in N} d(x, C(x))$, where $C(x)$ is the closest member of K to x. $f^S(K)$ is the quality of the same median set K when used to cluster the sample S.

4.2. *Generation of a good candidate set*

We will draw a sample S of s points, by selecting s points uniformly at random, with replacement, from N, where s is $\Omega(\frac{k}{\epsilon} \log k)$. We solve the k-Median problem to obtain a g-approximation on this sample using the algorithm[2] of Arya et al. (2001). The value of g depends on the version of Arya's algorithm we use; by modifying the running time of the algorithm we can obtain any $3 < g \leq 5$.

Theorem 1. *With constant probability we will produce medians K such that $f^N(K)$ is $O(1) f^N(K^*)$.*

Proof: The main idea of the proof is to show that each of the optimum medians has one of our medians nearby. This claim depends upon various probabilistic events, and requires the assumption that the optimum clusters are large.

Consider an optimal median set $K^* = \{K_1^*, \ldots, K_k^*\}$. For $1 \leq i \leq k$, let C_i^* be the cluster of points about K_i^*. Let $n_i = |C_i^*|$, so $n_i \geq \frac{\epsilon n}{k}$. Let n_i^S be the number of points in the ith optimal cluster that were also picked by the sample. For all $x \in N$, let $C^*(x)$ denote the closest median in K^* and $C(x)$ the closest median in K. Let $Q_i = \sum_{x \in C_i^*} d(x, C^*(x))$ and $R_i = \sum_{x \in C_i^*} d(x, C(x))$. That is, Q_i is the total assignment cost under the optimal centers K^* for the points in C_i^* and R_i is the total assignment cost of the same points to our centers K. We immediately observe that $\sum_{1 \leq i \leq k} Q_i = f^N(K^*)$ and that $\sum_{1 \leq i \leq k} R_i = f^N(K)$. Let Q_i^S and R_i^S represent the same summations as Q_i and R_i, but restricted to cover only the points in the sample: $Q_i^S = \sum_{x \in C_i^* \cap S} d(x, C^*(x))$ and $R_i^S = \sum_{x \in C_i^* \cap S} d(x, C(x))$.

We observe $\sum_{1 \leq i \leq k} Q_i^S = f^S(K^*)$ and $\sum_{1 \leq i \leq k} R_i^S = f^S(K)$. The triangle inequality ensures a bound of

$$\min_{x \in C_i^* \cap S} (d(K_i^*, x) + d(x, C(x)))$$

on the distance from optimum median K_i^* to the nearest K_j. This quantity is at most

$$\frac{1}{n_i^S}\left(\sum_{x\in C_i^*\cap S} d(x, C^*(x)) + d(x, C(x))\right) = \frac{1}{n_i^S}\left(Q_i^S + R_i^S\right)$$

We can then bound the cost of using our medians to cluster the entire data set N. The distance from a point $x \in N$ to the nearest median in K is bounded by the sum of the distance to its optimum median and the distance from this optimum median to the nearest median in K. $f^N(K)$ is therefore at most

$$\sum_{x\in N} d(x, C^*(x)) + \sum_{i=1}^{k}\sum_{x\in C_i^*}\frac{1}{n_i^S}\left(Q_i^S + R_i^S\right) = \sum_{x\in N} d(x, C^*(x)) + \sum_{i=1}^{k}\frac{n_i}{n_i^S}\left(Q_i^S + R_i^S\right).$$

By Chernoff Bounds, $\forall\, i$ $\mathbf{Pr}[n_i^S < Bn_i(s/n)] < \exp(-sn_i(1-B)^2/(2n))$, where $0 < B < 1$ is a chosen parameter that trades the sample size against the approximation factor. We would like this probability to be smaller than $\frac{1}{k}$ so that there is a constant probability that *all* clusters are well represented in the sample. After some algebra, and making use of our assumption that $n_i \geq \frac{\epsilon n}{k}$, we can determine that $s = \frac{2}{(1-B)^2}\frac{k}{\epsilon}\log(32k)$ is sufficient to guarantee that $\mathbf{Pr}[\exists i\ n_i^s < Bn_i(s/n)] < \frac{1}{32}$. The important point is that, with constant probability, *all* of the optimum clusters will be well represented in the sample. Provided all optimum clusters are well represented, we can simplify the earlier expression given for the quality of clustering using the medians derived from the sample, obtaining

$$f^N(K) \leq f^N(K^*) + \frac{n}{sB}\sum_{i=1}^{k}\left(Q_i^S + R_i^S\right) = f^N(K^*) + \frac{n}{sB}(f^S(K) + f^S(K^*)).$$

Because the median set K was found by the algorithm of Arya et al., it is guaranteed that for all k-element median sets $K' \subseteq S$, $f^S(K) \leq gf^S(K')$. Fact 2.1 guarantees, then, that for all k-element median sets $K' \subseteq N$, $f^S(K) \leq 2gf^S(K')$. In particular, $f^S(K) \leq 2gf^S(K^*)$. It follows that $f^N(K) \leq f^N(K^*) + \frac{n}{sB}(1+2g)f^S(K^*)$. The final step is estimating $f^S(K^*)$, the cost of K^* on the sample. The expected value is $E[f^S(K^*)] = \frac{s}{n}f^N(K^*)$. Of course, there is some possibility that our sample has a much higher cost, but we can apply Markov's Inequality to bound the probability of this event. We observe that $\mathbf{Pr}[f^S(K^*) \leq \frac{16}{15}\frac{s}{n}f^N(K^*)] \geq \frac{1}{16}$. We conclude that $\mathbf{Pr}[f^N(K) \leq 1 + (1+2g)\frac{16}{15B}f^N(K^*)] \geq 1 - \frac{1}{32} - \frac{15}{16} = \frac{1}{32}$. The required sample size was $s = \frac{2}{(1-B)^2}\frac{k}{\epsilon}\log(32k)$. If we consider $B = \frac{4}{5}$ we will obtain an approximation ratio of $1 + (1+2g)\frac{4}{3}$ with probability at least $\frac{1}{32}$, and a sample size of $50\frac{k}{\epsilon}\log(32k)$. By modifying the constants involved, we can make our approximation ratio arbitrarily close to $2 + 2g$. □

We have shown that a given candidate set, generated as described, has constant probability of being a good approximation to k-Median. One approach to clustering would be to change the constants so as to guarantee a high probability of success. We observe, however, that our

proof depends upon the claim that $f^S(K^*) = \frac{s}{n} f^N(K^*)$. This claim holds in an expected sense but not with high probability. It might be the case that a small number of points contribute significantly to the cost of clustering; imagine a set of $\frac{n}{s}$ points which account for nearly all the cost of the optimum clusters on the set N. Our sample expects to see only one of these points, but there is reasonably good probability we will see two, thus doubling the value of $f^S(K^*)$ relative to $f^N(K^*)$. Guaranteeing a good result with high probability would therefore cost us at least a factor of two in the approximation ratio.

Instead, we will independently generate m candidate sets, where m is a constant. The probability that at least one of them produces a good approximation tends towards $1-(\frac{31}{32})^m$. This process allows us to guarantee with high probability that we can generate a set of candidates such that *one of them* is a good approximation to k-Median. That is, if the candidate sets are $K_1, \ldots, K_m$, and K_{BEST} is the candidate set with the lowest f^N cost, then with probability at least $1-(31/32)^m$, $f^N(K_{BEST}) \leq (1+\frac{4}{3}(1+2g)) f^B(K^*)$. Next we give a procedure that will identify a good median set from $K_1, \ldots, K_m$.

4.3. *Close estimation of candidate quality*

Suppose we are given m sets of candidate medians K_1 through K_m. We would like to determine which of these sets minimizes $f^N(K_i)$, in sublinear time. Clearly it is possible to measure $f^N(K_i)$ exactly for each i, by calculating, for each K_i, the assignment distances of *all* n points in N to the medians K_i. This process would take $\Omega(nkm)$ time, however, defeating the purpose of using a sampling-based algorithm. Instead, we will *estimate* each $f^N(K_i)$.

The estimation procedure will be as follows. l independent estimation samples $T_1, \ldots, T_l$, with $|T_j| = t$ for all $1 \leq j \leq l$, will be drawn uniformly at random from N. Each of these samples will consist of t points drawn uniformly at random, with replacement. The number l and the size of these samples will be discussed shortly. On each of these estimation samples T_j, for each K_i, $f^{T_j}(K_i)$ will be computed *exactly*. For each K_i, the estimated cost $f^{EST}(K_i)$ will be calculated as $\min_{1\leq j\leq l} (\frac{n}{t} f^{T_j}(K_i))$. The candidate set with minimum f^{EST} will be returned as the final k-Median solution.

First we will show that K_{BEST}, the candidate set with lowest f^N cost, will with high probability have $f^{EST}(K_{BEST}) \leq \alpha f^N(K_{BEST})$, for some small constant α. It will follow that with high probability there will be some K_i with a small $f^{EST}(K_i)$, and therefore that the median set returned by the algorithm will at least have a low *estimated* cost. Finally we will show the soundness of the estimation procedure: that for every candidate set K_i it will be unlikely that $f^{EST}(K_i)$ is much *lower* than $f^N(K_i)$. It will follow that the median set returned has not only a low *estimated* cost but also a low *actual* (f^N) cost.

We introduce two new parameters α and c. The parameter α trades off the number of samples against the accuracy of the estimation, while c trades off the number of samples against the probability of failure. We independently draw $l = \frac{c\alpha}{\alpha-1}$ samples T_j from N, each of size $t = \Theta(\frac{k}{\epsilon} \log \frac{k}{\epsilon})$. For each median set i, we will compute $f^{EST}(K_i) = \min_{1\leq j\leq l} (\frac{n}{t} f^{T_j}(K_i))$. We will show that the medians that minimize f^{EST} produce a good approximation to the minimum f^N.

Low-estimate property of candidate sets. We will first show that, for any candidate set K, our estimated cost $f^{EST}(K)$ is unlikely to be much higher than the actual cost $f^N(K)$ on the whole data set.

Lemma 1. *For any candidate set K, if we compute $f^{EST}(K)$ using $\frac{c\alpha}{\alpha-1}$ samples ($\alpha \geq 2$), then with probability at least $1 - e^{-c}$, $f^{EST}(K) \leq \alpha f^N(K)$.*

Proof: The proof follows directly from Markov's Inequality, and the fact that for any random sample T_j, $\mathbf{Pr}[f^{T_j}(K) > \alpha f^N(K)]$ is at most $\frac{\mu}{\alpha f^N(K)}$, where μ is the expected value of $f^{T_j}(K)$. Therefore, this probability is at most $\frac{t}{n\alpha} \leq \frac{1}{\alpha}$. The probability that $f^{EST}(K)$ exceeds $\alpha f^N(K)$ is the probability that for *all* $T_j f^{T_j}(K) > \alpha f^N(K)$, which, because the samples were drawn independently, is at most $\frac{1}{\alpha}^{\frac{c\alpha}{\alpha-1}} \leq e^{-c}$. □

As $f^N(K_{BEST})$ is with high probability close to $f^N(K^*)$, with high probability $f^{EST}(K_{BEST})$ is close to $f^N(K^*)$. Therefore, with high probability some candidate set will have an f^{EST} close to $f^N(K^*)$.

Soundness of the estimation method. For each candidate median set K we can consider the ordering of all the points $x \in N$ by their assignment distances under K[3]. Further, imagine dividing all the points of N into $b = \frac{kD}{\epsilon}$ "bins" each containing $\frac{\epsilon n}{kD}$ points, where $D \geq 2$ is a constant (see figure 1). The first bin will have the "cheapest" points (i.e., those with the smallest assignment distances) and the last bin will have the most expensive points. We will denote by COST(i) the cost of the ith bin, that is, the sum of assignment distances of points in this bin.

By selecting our sample size to be sufficiently large, we will guarantee that with high probability, for each median set, each bin is "well-represented" in each estimation sample. We will then show how to bound the cost $f^N(K)$ in terms of $f^{EST}(K)$ under the assumption that all bins are well-represented.

We will set $t = \frac{4b}{(1-\beta)^2} \log(mb\frac{c\alpha}{\alpha-1})$, which is $\Theta(\frac{k}{\epsilon} \log \frac{k}{\epsilon})$. Here β is a parameter trading off between the sample size and the accuracy of the estimation method, m is the number of samples, $b = \frac{kD}{\epsilon}$ is the number of bins, and α and c are the parameters defining the number of samples.

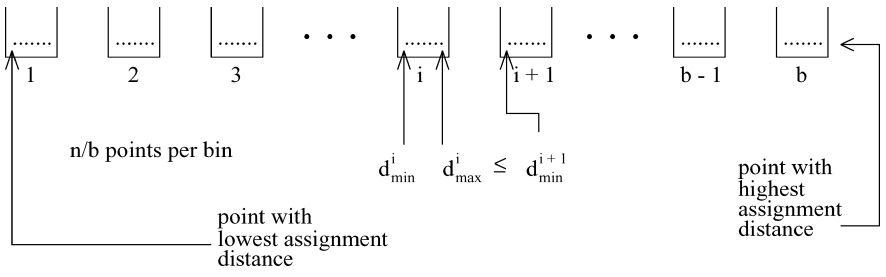

Figure 1. Ordering of points by assignment distances under a candidate median set K.

Lemma 2. *With high probability, for every triple of* (*bin*, *median set*, *sample*), *we have at least* $\frac{\beta t}{b}$ *points from this bin in the sample.*

Proof: The expected number of points is $\frac{t}{b}$ since points from N are equally divided among the bins. Using Chernoff Bounds, the probability that a particular bin for a particular median set has at most $\beta t/b$ points in a particular sample is at most $e^{\frac{-t(1-\beta)^2}{2b}}$. Substituting the value of t yields a probability of at most $(mb\frac{c\alpha}{\alpha-1})^{-2}$. There are m median sets, b bins, and $\frac{c\alpha}{\alpha-1}$ samples, so the probability that there exists a bad triple remains very small (and can be reduced by increasing the constants). □

Lemma 3. *With high probability, for all median sets* K, $f^{EST}(K) \geq \beta \sum_{i=1}^{b-1} \text{COST}(i)$.

Proof: Let us denote by $d_i^{\max}$ the assignment distance of the last point, i.e., the point with largest assignment distance, in bin i. Let $d_i^{\min}$ be the assignment distance of the first point in bin i. (See figure 1.)

Recall that

$$f^{EST}(K) = \min_{1 \leq j \leq l} \left(\frac{n}{t} f^{T_j}(K) \right),$$

where $l = c\alpha/(\alpha - 1)$ is the number of estimation samples, and T_j is the jth estimation sample. By Lemma 2, with high probability, all T_j have at least $\beta t/b$ points that fall into bin i for median set K, and that will therefore contribute at least $d_i^{\min}$ apiece to $f^{T_j}(K)$. Therefore, with high probability, $\forall\, j\ 1 \leq j \leq l\ f^{T_j}(K) \geq \sum_{i=1}^{b} \frac{\beta b}{t} d_i^{\min}$, and so, since $f^{EST}(K)$ $= \min_{1 \leq j \leq l} (\frac{n}{t} f^{T_j}(K))$, we have

$$f^{EST}(K) \geq \frac{n}{t} \sum_{i=1}^{b} \frac{\beta t}{b} d_i^{\min}.$$

$\frac{n}{t} \sum_{i=1}^{b} \frac{\beta t}{b} d_i^{\min} \geq \frac{\beta n}{b} \sum_{i=1}^{b-1} d_i^{\max}$, since, for all $1 \leq i \leq b$, $d_i^{\min} \geq d_{i-1}^{\max}$. $\frac{\beta n}{b} \sum_{i=1}^{b-1} d_i^{\max} \geq$ $\beta \sum_{i=1}^{b-1} \text{COST}(i)$, because $\text{COST}(i)$ is the sum of assignment costs of the n/b members of bin i, for medians K, and these costs are each at most $d_i^{\max}$. □

Lemma 4. *For all candidate median sets* K, *the cost of the most expensive bin*, $\text{COST}(b)$, *is at most* $f^N(K^*) + \frac{1}{D-1} \sum_{i=1}^{b-1} \text{COST}(i)$.

Proof: We will show an upper bound on the cost of the most expensive bin, by showing a way of "moving" points in this bin to their centers.[4] We will call a point "bad" if it is in the most expensive bin, and we will call it "good" if it is not in any of the $D - 1$ most expensive bins. Figure 2 shows the bins re-labelled as to whether they contain good, bad, or non-good points.

Consider the optimal clusters $C_1^*, C_2^*, \ldots, C_k^*$ (those induced by the optimal medians K^*). Some of these contain bad points. We will imagine moving each bad point first to

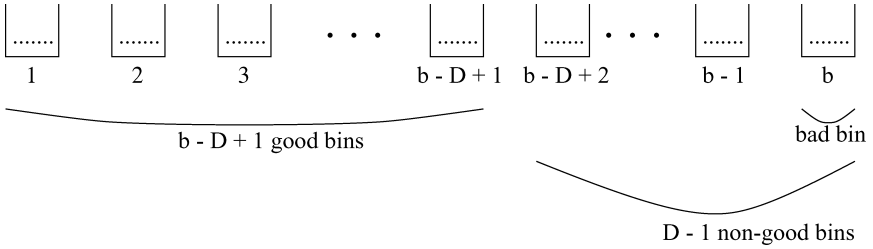

Figure 2. Good, bad, and non-good bins.

its optimal median and then to the location of a good point in the same optimal cluster, in such a way that each bad point has a unique new (good) location and so that the sum of new assignment distances of the (formerly) bad points is bounded. For each optimal cluster containing one or more bad points, we can find for each bad point a unique good point in the same cluster (because each optimal cluster contains at least $\frac{\epsilon n}{k}$ points, and there are at most $(D-1)\frac{\epsilon n}{kD} = \frac{D-1}{D}\frac{\epsilon n}{k}$ points in N which are not good). If we move each bad point to the location of a good point in the same optimal cluster, then we know that the sum of the movement distances is at most $f^N(K^*)$. This bound holds because each bad point is moved at most $d_1 + d_2$ where d_1 is its own assignment distance under the optimal centers, and d_2 is the assignment distance of a different point in that optimal cluster; adding up a d_1 and a d_2 for each bad point, we get a sum of only some of the optimal-median assignment distances, which is then at most $f^N(K^*)$. Figure 3 illustrates this process.

Now that the bad points are all at good locations, we need to show that the points at these good locations can be assigned to medians in K for a total assignment cost of at

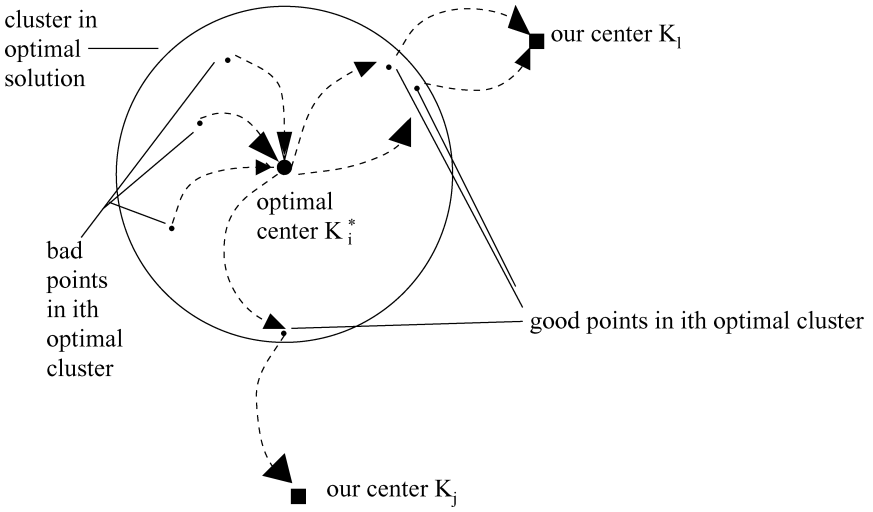

Figure 3. Assignment distances of bad points.

most $\frac{1}{D-1}\sum_{i=1}^{b}(\text{COST}(i))$; the lemma will follow. Recall that each bad point was assigned to a unique good point, and so, in the worst case, the bad points have been moved so that they occupy exactly the locations of the points from bin $b - D + 1$ (the "worst" good bin). In other words, the total remaining distance that the bad points need to travel from their new locations to medians in K is at most $\text{COST}(b - D + 1)$. Since the points were arranged in bins by non-decreasing assignment distances, bin $b - D + 1$ accounts for at most a $1/(D - 1)$ fraction of the costs of the $D - 1$ bins from $b - D + 1$ through $b - 1$. Therefore, $\text{COST}(b - D + 1) \leq \frac{1}{D-1}\sum_{i=b-D+1}^{b-1}\text{COST}(i) \leq \frac{1}{D-1}\sum_{i=1}^{b-1}\text{COST}(i)$. The lemma follows. □

Theorem 2. *With high probability,* $f^N(K)$ *is at most* $f^N(K^*) + \frac{1}{\beta}\frac{D}{D-1}f^{EST}(K)$.

Proof: This follows directly from Lemmas 3 and 4. □

By adjusting the constants, we can cause the above expression to converge so that $f^N(K)$ is bounded by an expression arbitrarily close to $f^N(K^*) + f^{EST}(K)$. Combining this result with Lemma 1, we have $f^N(K) - f^N(K^*) \leq f^{EST}(K) \leq \alpha f^N(K)$.

4.4. Combining generation and estimation

We will restate the algorithm that we have built up over the last several sections, and state and prove its approximation and probability guarantees. The k-Median algorithm is as follows:

Algorithm **Sample-k-Median**

1. For $i = 1$ to m, draw a sample S_i of $s = 50\frac{k}{\epsilon}\log(32k)$ points uniformly at random, with replacement, from N. m is a constant.
2. For $i = 1$ to m, solve k-Median to a g-approximation on S_i, using the algorithm of Arya et al., to produce a median set K_i.
3. For $j = 1$ to $l = \frac{c\alpha}{\alpha-1}$, draw a sample T_j of $t = \frac{4b}{(1-\beta)^2}\log(mb\frac{c\alpha}{\alpha-1})$ points uniformly at random, with replacement, from N. $\alpha \geq 2$, $\beta < 1$, and $c > 1$ are constants; b is kD/ϵ for constant $D \geq 2$.
4. For each pair (K_i, T_j), calculate $f^{T_j}(K_i)$, and for each K_i let $f^{EST}(K_i) = \min_{1\leq j\leq l}(f^{T_j}(K_i))$.
5. Return K_i such that $f^{EST}(K_i)$ is minimum, as a k-Median solution for N.

The running time of the algorithm is equal to the time required for sampling plus the time required for clustering and estimating clustering cost. The amount of time spent on sampling is $O(\frac{k}{\epsilon}\log\frac{k}{\epsilon})$. Since the algorithm of Arya et al. clusters n points in $O(k(nk)^{p+1}\log n)$ time, the time spent clustering is $O(k(\frac{k^2}{\epsilon}\log k)^{p+1}\log(\frac{k}{\epsilon}\log k))$. If A is a set of size a, it takes $O(ak)$ time to estimate $f^A(K)$, so the amount of time spent estimating the costs of the median sets is $O(\frac{k^2}{\epsilon}\log\frac{k}{\epsilon})$. The overall running time, then, is asymptotically equal to the time spent clustering the samples S_1 through S_m, or

$$O\left(k\left(\frac{k^2}{\epsilon}\log\left(m\frac{c\alpha}{\alpha-1}\frac{k}{\epsilon}\right)\right)^{p+1}\log\left(\frac{k}{\epsilon}\log\left(m\frac{c\alpha}{\alpha-1}\frac{k}{\epsilon}\right)\right)\right).$$

Theorem 3. *Given an instance of k-Median, a data set N of n points with metric $dist$, and an integer k, with optimal solution K^* having all clusters of size at least $\frac{\epsilon n}{k}$, we can with high probability produce a $(9 + O(\frac{1}{p}))$-approximation (where $p \in \{1, \ldots, k\}$), in running time $O(k(\frac{k^2}{\epsilon}\log(m\frac{c\alpha}{\alpha-1}\frac{k}{\epsilon}))^{p+1}\log(\frac{k}{\epsilon}\log(m\frac{c\alpha}{\alpha-1}\frac{k}{\epsilon})))$. The probability of success is at least $1-(\frac{31}{32})^m - e^{-c} - (mb\frac{c\alpha}{\alpha-1})^{-2}$.*

Proof: We first generate a large number of candidate median sets. By Theorem 1 the approximation ratio of the best candidate set is with high probability bounded above by a quantity arbitrarily close to $2+2g$. We then perform the testing step, selecting a candidate K whose estimated cost $f^{EST}(K)$ is (by Theorem 2) with high probability between $f^N(K) - f^N(K^*)$ and $\alpha f^N(K)$. Therefore, the true approximation factor of the final medians is $1+\alpha(2+2g)$, with high probability. g is the approximation ratio of the k-Median algorithm used to find the candidates; the algorithm of Arya et al. yields $g = 3 + 2/p$.

The probability of failure is at most the sum of the failure probabilities, which are $(\frac{31}{32})^m$ (the probability that no median set will have a low enough cost on N); e^{-c} (the probability that the candidate median set with the lowest cost on N will not have a low enough estimated cost); and $(mb\frac{c\alpha}{\alpha-1})^{-2}$ (the probability that the candidate set with the lowest estimated cost will have a high cost on N).

The theorem follows. □

5. A simpler algorithm

In this section we will give a simpler algorithm, involving just a single random sample of size $O(\frac{k}{\epsilon}\ln\frac{k}{\epsilon})$, and a single invocation of a k-Median approximation algorithm, that, with high probability finds an approximately optimal k-Median solution in the case that the optimal clusters are not too small.

We again have a set N of n points, and a parameter k, so that, in some optimal k-Median clustering of N, each cluster contains at least $\frac{\epsilon n}{k}$ points. We will use as a subroutine the algorithm of Charikar et al. (2001) that will produce a $(1-\kappa_1\delta)$-solution of cost at most κ_2 times the optimal $(1-\delta)$-solution cost (where κ_1 and κ_2 are constants). Here for $0 \le \gamma \le 1$, a $(1-\gamma)$-solution K is a set of k medians that leave γn points unassigned. We will ignore the γn members of N farthest from K in evaluating the cost of K as a $(1-\gamma)$-solution; in particular, we will say that the $(1-\gamma)$-cost of K on N, which we will denote $f^N_{1-\gamma}(K)$, is the sum of the assignment distances of the $(1-\gamma)n$ members of N that are closest to the medians K.

We will pick a constant κ_1, and let $\delta = (0.375\epsilon)/((\kappa_1+0.3)k)$, $\delta' = \kappa_1\delta$, and D be a constant no less than $(\kappa_1+0.3)/(0.3)$.

We will show that, with probability at least $1-\frac{1}{k^2}$, the following algorithm, **OneSample**, will produce a median set K with $f^N(K) \le (1+\frac{(1+\kappa_2)D}{0.3(D-1)})f^N(K^*)$.

Algorithm **OneSample**

1. Draw a sample S of $s = \frac{ckD}{\epsilon}\ln(\frac{kD}{\epsilon})$ points uniformly at random, with replacement, from N. (The constant c will be specified shortly.)

2. Using the algorithm of Charikar et al., find median set $K = \{K_1, \ldots, K_k\}$ whose $(1-\delta')$-cost on S is at most κ_2 times the cost of the best $(1-\delta)$-solution for S.
3. Return K.

The running time of this algorithm is the amount of time spent computing the clustering, which, since the algorithm of Charikar et al. runs in $O(n^2 \log n \log M)$ time on a diameter-M point set of n points, is equal to

$$O\left(\left(\frac{k}{\epsilon}\log\frac{k}{\epsilon}\right)^2 \log\left(\frac{k}{\epsilon}\log\frac{k}{\epsilon}\right)\log M\right).$$

We claim that **OneSample** will, with high probability, produce an approximately optimal clustering of N. The following definitions will facilitate our proof:

Definition 7. As before, let $K^* = \{K_1^*, \ldots, K_k^*\}$ denote some optimal set of medians for N; for each $1 \le l \le k$, let C_l^* again denote the members of N assigned to K_l^*, and let $|C_l^*| = n_l$.

Definition 8. Given a point set X, a median set Y for X, and an integer $z < |X|$ (for simplicity we assume z divides $|X|$), let "bins" $W_1, \ldots, W_z \subseteq X$ be defined as follows. The elements of X, in increasing order of $d(\cdot, Y)$, are assigned to bins in the order $W_1, \ldots, W_z$, so that each bin receives $|X|/z$ points. We will call the bins $W_1, \ldots, W_z$ a *z-bin division of X with respect to Y*.

Definition 9. Let $B_1, \ldots, B_b$ be a b-bin division of N with respect to K^*, where $b = \frac{kD}{\epsilon}$.

Definition 10. Let $\hat{C}_l = C_l^* \setminus B_b$; let $\hat{n}_l = \hat{C}_l$; let $S' = S \setminus B_b$.

Fact 5.1. $\hat{n}_l \ge \frac{D-1}{D} n_l$.

Proof: Recall that $n_l = |C_l^*|$, and $\hat{n}_l = |\hat{C}_l|$. By the definitions of $\hat{C}_l$ and C_l^*, $\hat{n}_l \ge n_l - n/b = n_l - (n\epsilon)/(kD) \ge n_l - n_l/D = \frac{D-1}{D} n_l$. □

Lemma 5. *If $s \ge \max\{35b\ln(4k^2b), \frac{35b}{D-1}\ln(4k^3)\}$, then with probability at least $1-k^{-2}$ the following are both true*:
1. (*Condition I*) *For all* $1 \le i \le b$, $|S \cap B_i| \in [\frac{0.75s}{b}, \frac{1.25s}{b}]$.
2. (*Condition II*) *For all* $1 \le l \le k$, $|S' \cap C_l^*| = |S \cap \hat{C}_l| \in [\frac{0.75\hat{n}_l s}{n}, \frac{1.25\hat{n}_l s}{n}]$.

Proof: The proof follows by a straightforward application of Chernoff Bounds. If $s \ge 35b\ln(4k^2b)$, it follows that for a particular bin B_i the probability that $|S \cap B_i| < \frac{0.75s}{b}$ is at most $\frac{1}{4k^2b}$, and the probability that $|S \cap B_i| > \frac{1.25s}{b}$ is also at most $\frac{1}{4k^2b}$. Therefore, the probability that there exists a bin B_i with $|S \cap B_i| < \frac{0.75s}{b}$ is at most $\frac{1}{4k^2}$, and the probability that there exists a bin B_i with $|S \cap B_i| > \frac{1.25s}{b}$ is also at most $\frac{1}{4k^2}$, so the probability that

there exists a bin with $|S \cap B_i| \notin [\frac{0.75s}{b}, \frac{1.25s}{b}]$ is at most $\frac{1}{2k^2}$. It can similarly be shown, using Fact 5.1, that if $s \geq \frac{35b}{D-1} \ln(4k^3)$ the probability that there exists $1 \leq l \leq k$ for which $|S \cap \hat{C}_l| \notin [\frac{0.75\hat{n}_l s}{n}, \frac{1.25\hat{n}_l s}{n}]$ is at most $\frac{1}{2k^2}$. The lemma follows. □

Therefore, we will assume from here on that both Conditions I and II hold. Now we can prove an upper bound on $f^N(K)$, the cost of the medians output by **OneSample**.

Lemma 6.

$$f^N(K) \leq f^N(K^*) + \frac{nD}{0.75s(D-1)}(f^{S'}(K^*) + f^{S'}(K)).$$

Proof: As before, we observe that the cost of sending all members of N to the medians K is the cost of moving each point to the closest member of K^*, and then moving each point from its optimal median to the nearest member of K. That is,

$$f^N(K) \leq f^N(K^*) + \sum_{l=1}^{k} n_l d(K_l^*, K).$$

Using an argument similar to the one from the proof of Theorem 1, we see that

$$\begin{aligned}
\sum_{l=1}^{k} n_l d(K_l^*, K) &\leq \sum_{l=1}^{k} \left(\frac{n_l}{|C_l^* \cap S'|} \sum_{x \in C_l^* \cap S'} (d(K_l^*, x) + d(x, K)) \right) \\
&\leq \sum_{l=1}^{k} \left(\frac{n_l n}{0.75\hat{n}_l s} \sum_{x \in C_l^* \cap S'} (d(K_l^*, x) + d(x, K)) \right) \\
&\leq \frac{nD}{0.75s(D-1)}(f^{S'}(K^*) + f^{S'}(K))
\end{aligned}$$

The second inequality follows from Condition II, and the third from Fact 5.1. □

Fact 5.2. $f^{S'}(K^*) \leq \frac{1.25s}{n} f^N(K^*)$.

Proof: Recall that S' contains no points from the highest bin B_b, and that (by Condition I) it contains at most $1.25s/b$ points from each bin B_i with $i < b$. By arguments similar to those in the proof of Lemma 3, the result follows. □

Lemma 7. $f^{S'}(K) \leq f^{S'}(K^*) + 2\kappa_2 f_{1-\delta}^S(K^*)$.

Proof: Let $P_1, \ldots, P_{b'}$ be a b'-bin division of S with respect to K, where $b' = 1/\delta'$. Note that by Condition II, each cluster C_l^* contains at least $\frac{0.75\hat{n}_l s}{n}$ points from S'.

We claim that $2s/b' \leq \frac{0.75\hat{n}_l s}{n}$, or, equivalently, that $\frac{0.375\kappa_1\epsilon}{(\kappa_1+0.3)k} = \delta' \leq \frac{0.375\hat{n}_l}{n}$. It is then sufficient to show that $\frac{\epsilon n}{k}\frac{\kappa_1}{(\kappa_1+0.3)} \leq \hat{n}_l$. Note that by definition of D, the assumption that $n_l \geq \epsilon n/k$, and Fact 5.1, $\frac{\epsilon n}{k}\frac{\kappa_1}{(\kappa_1+0.3)} \leq \frac{\epsilon n}{k}\frac{D-1}{D} \leq n_l\frac{D-1}{D} \leq \hat{n}_l$. Therefore, each cluster C_l^* contains at least $2s/b'$ points from S'. By definition of the b'-bin division, S' contains at most s/b' points from $S' \cap P_b$, and so we can map each $x \in S' \cap P_b$ to a unique $g(x)$ in $S' \setminus P_b$, that is in the same optimal cluster C_l^* as x. We can then account for the cost of $f^{S'\cap P_{b'}}(K)$ as the cost of moving each $x \in S' \cap P_b$ to the closest member of K^*, then to $g(x)$, and then to the closest member of K to $g(x)$. Then we have

$$\begin{aligned}
f^{S'\cap P_{b'}}(K) &\leq f^{S'\cap P_{b'}}(K^*) + f^{S'\setminus P_{b'}}(K^*) + f^{S'\setminus P_{b'}}(K) \\
&= f^{S'}(K^*) + f^{S'\setminus P_{b'}}(K) \\
f^{S'}(K) &= f^{S'\cap P_{b'}}(K) + f^{S'\setminus P_{b'}}(K) \\
&\leq f^{S'}(K^*) + 2f^{S'\setminus P_{b'}}(K) \\
&= f^{S'}(K^*) + 2f^{S}_{1-\delta'}(K) \\
&\leq f^{S'}(K^*) + 2\kappa_2 f^{S}_{1-\delta}(K^*)
\end{aligned}$$

The last inequality follows from the use of the algorithm of Charikar et al., which guarantees a $(1-\delta')$ solution for S whose cost is at most κ_2 times the cost of the best $(1-\delta)$ solution, which in turn cannot exceed $f^S_{1-\delta}(K^*)$. □

Theorem 4. $f^N(K) \leq (1 + \frac{(1+\kappa_2)D}{0.3(D-1)})f^N(K^*)$.

Proof: First we show that $|B_b \cap S| \leq \delta s$, i.e., that the top bin B_b of N contains at most δs members of S. First, by Condition I, $|B_b| \leq 1.25s/b$, so it suffices to show that $1.25/b \leq \delta$. Recall that $\delta = \frac{0.375\epsilon}{(\kappa_1+0.3)k}$, which, by definition of b, is $\frac{0.375D}{(\kappa_1+0.3)b}$; by choice of D this quantity is at least $0.375/(0.3b) = 1.25/b$. Therefore, excluding the δs members of S that are furthest from K^* means excluding all the members of B_b. Since by Condition I S contains at most $1.25s/b$ points from each bin B_i, we can use the same arguments from the proofs of Lemma 3 and Fact 5.2 to show that $f_{1-\delta}(K^*) \leq \frac{1.25s}{n} f^N(K^*)$. Combining this bound with the results of Lemma 6, Fact 5.2, and Lemma 7, we get:

$$\begin{aligned}
f^N(K) &\leq f^N(K^*) + \frac{nD}{0.75s(D-1)}(f^{S'}(K^*) + f^{S'}(K)) \\
&\leq f^N(K^*) + \frac{nD}{0.75s(D-1)}\left(f^{S'}(K^*) + f^{S'}(K^*) + 2\kappa_2 f^S_{1-\delta}(K^*)\right) \\
&\leq f^N(K^*) + \frac{nD}{0.75s(D-1)}\left(2f^{S'}(K^*) + \frac{2.5\kappa_2 s}{n} f^N(K^*)\right) \\
&\leq f^N(K^*) + \frac{nD}{0.75s(D-1)}\left(\frac{2.5s}{n} f^N(K^*) + \frac{2.5\kappa_2 s}{n} f^N(K^*)\right)
\end{aligned}$$

$$\leq f^N(K^*) + \frac{D}{0.75s(D-1)}(1+\kappa_2)2.5sf^N(K^*)$$
$$= \left(1 + \frac{(1+\kappa_2)D}{0.3(D-1)}\right) f^N(K^*)$$

□

6. Clustering with outliers

Instead of assuming all the optimum clusters are large, we can instead allow a clustering that discards some fraction ϕ of the demand points. These points are considered outliers (or noise) and their distances are not included in the cost of the solution. Recall that if K is a set of k medians for a set N of n points, we call the cost of K on N, not counting the ϕn highest assignment distances, the $(1-\phi)$-cost of the solution K; as before, this cost will be denoted $f^N_{1-\phi}(K)$. For a point set N, a set of medians that are considered in terms of their $(1-\phi)$-cost on N will be called a $(1-\phi)$-solution for N; for simplicity, the $(1-\phi)$-cost of a solution already designated as a $(1-\phi)$-solution will simply be called the cost of the solution (unless this term is ambiguous). Finally, a median set K^* is an optimal $(1-\phi)$-solution if it is the set of medians with the lowest $(1-\phi)$-cost on N.

As we will show, given a set N of n points for which the median set $K^* = \{K^*_1, \ldots, K^*_k\}$ is an optimal $(1-\phi)$-solution, with high probability the following algorithm **OutlierKM** will produce a median set K whose $(1-\alpha\phi)$-cost on N (for constant α) is at most a constant times $f^N_{1-\phi}(K)$. In the following algorithm, κ_1 and κ_2 are constants.

Algorithm **OutlierKM**

1. Draw a sample S of $s = \max\left(\frac{35}{\phi}\ln\frac{4k^2}{\phi}, \frac{32k}{\phi}\ln 4k^3\right)$ points independently and uniformly at random with replacement from N.
2. Using the algorithm of Charikar et al. (2001), find a $(1-2.5\kappa_1\phi)$-solution K with $f^S_{1-2.5\kappa_1\phi}(K) \leq \kappa_2 f^S_{1-2.5\phi}(\hat{K})$, where $\hat{K}$ is the optimal $(1-2.5\phi)$-solution for S.
3. Return K as a $(1-(9+7\kappa_1)\phi)$-solution for N.

The running time of **OutlierKM** is the amount of time spent computing the clustering. This is equal to

$$O\left(\left(\frac{k}{\phi}\left[\log k^3 + \log\frac{k^2}{\phi}\right]\right)^2 \log\left(\frac{k}{\phi}\left[\log k^3 + \log\frac{k^2}{\phi}\right]\right)\log M\right).$$

We will show that with probability at least $1 - \frac{3}{4k^2}$,

$$f^N_{1-(9+7\kappa_1)\phi}(K) \leq \left[1 + \frac{10}{3}(1+\kappa_2)\right] f^N_{1-\phi}(K^*).$$

Definition 11. Let $B_1, \ldots, B_b$ be the b-bin division (where $b = 1/\phi$) of N with respect to K^*.

Definition 12. Let the median set $K^* = \{K_1^*, \ldots, K_k^*\}$ be the optimal $(1-\phi)$-solution for N; for each l with $1 \leq l \leq k$, let C_l^* denote the set of points of N assigned to K_l^* (therefore, none of the ϕn points left unassigned by the $(1-\phi)$-solution K^* is a member of a cluster C_l^*). For $1 \leq l \leq k$, let $n_l = |C_l^*|$.

Definition 13. For $1 \leq l \leq k$, call C_l^* *large* if $|C_l^*| \geq \phi n/k$; otherwise call C_l^* *small*. Note that the small clusters together comprise at most ϕn points.

Definition 14. Let A denote the subset of S that is assigned under both the following solutions: K^* as a $(1-2.5\phi)$-solution for S, and K as a $(1-2.5\kappa_1\phi)$-solution for S. That is, $x \in A$ just if $x \in S$, x is among the $(1-2.5\phi)s$ members of S closest to the median set K^*, and x is among the $(1-2.5\kappa_1\phi)s$ members of S to the median set K. $|A|$ is therefore at least $(1-2.5(1+\kappa_1)\phi)s$.

Definition 15. If C_l^* is a large cluster with $|C_l^* \cap A| \geq \frac{1}{2}|C_l^* \cap S|$, we will say C_l^* is *covered*. If a large cluster C_l^* is not covered, we will say C_l^* is *lost*. Without loss of generality, let $C_1^*, \ldots, C_{k_2}^*$ denote the covered clusters, and $C_{k_2+1}^*, \ldots, C_k^*$ denote the lost and the small clusters.

Proof overview. Lemma 8 will show that the sample size s is large enough to guarantee that S is a "good representation" of N, in that it contains approximately the expected number of representatives from each large cluster and from each bin B_i. Lemma 9 will show that $f^N_{1-(9+7\kappa_1)\phi}(K) \leq f^{N'}(K)$, where N' is a conveniently-defined subset of N. From then on, we will prove upper bounds on $f^{N'}(K)$ rather than on $f^N_{1-(9+7\kappa_1)\phi}(K)$. In Lemma 10 we will bound $f^{N'}(K)$ in terms of $f^N_{1-\phi}(K^*)$, $f^S_{1-2.5\phi}(K^*)$, and $f^S_{1-2.5\kappa_1\phi}(K^*)$ by showing that each member of K^* is close to some member of K. With Lemma 11 we will observe that the use of the algorithm of Charikar et al. in Step 2 guarantees that $f^S_{1-2.5\kappa_1\phi}(K) \leq \kappa_2 f^S_{1-2.5\phi}(K^*)$, so that the bound given in Lemma 10 can be re-written in terms of just $f^N_{1-\phi}(K^*)$ and $f^S_{1-2.5\phi}(K^*)$. In Fact 6.1 we will bound $f^S_{1-2.5\phi}(K^*)$ in terms of $f^N_{1-\phi}(K^*)$ by using the assumption that the members of S are well-distributed among the bins $B_1, \ldots, B_b$. Finally, in Theorem 5 we will show that

$$f^N_{1-(9+7\kappa_1)\phi}(K) \leq \left[1 + \frac{10}{3}(1+\kappa_2)\right] f^N_{1-\phi}(K^*)$$

by combining all the bounds.

Lemma 8. *With probability at least* $1 - \frac{3}{4k^2}$, *the following conditions both hold*:

Condition 6.1. $\forall\, i, 1 \leq i \leq b, |S \cap B_i| \in [0.75\phi s, 1.25\phi s]$.

Condition 6.2. $\forall\, l, 1 \leq l \leq k$ *with* $|C_l^*| \geq \phi n/k$, $|S \cap C_l^*| \geq 0.75 n_l s/n$ (*recall* $n_l = |C_l^*|$).

Proof: A straightforward application of Chernoff Bounds shows that Condition 6.1 holds since $s \geq \frac{35}{\phi} \ln \frac{4k^2}{\phi}$, and Condition 6.2 holds since $s \geq \frac{32k}{\phi} \ln 4k^3$. □

We will assume for Lemmas 9 through 11 and Fact 6.1 that Conditions 6.1 and 6.2 both hold.

Lemma 9. *Let $N' = N \setminus (X_{ex} \cup X_{sm} \cup X_l)$, where X_{ex} is the set of the ϕn members of N that are farthest from K^* (i.e., that are excluded by the $(1-\phi)$-solution K^*); X_{sm} is the union of all small clusters; and X_l is the union of all lost clusters. Then $f^N_{1-(9+7\kappa_1)\phi}(K) \leq f^{N'}(K)$.*

Proof: $|A| \geq (1 - 2.5(1+\kappa_1)\phi)s$, so $|S \setminus A| \leq 2.5(1+\kappa_1)\phi s$. Recall that X_l is the union of all lost clusters, and that a cluster C_l^* is lost if it is a large cluster such that fewer than half of its sample representatives (the points in $C_l^* \cap S$) are in A. Therefore for each lost cluster C_l^*, more than half the points in $C_l^* \cap S$ were in $S \setminus A$, and so $|C_l^* \cap S| \leq 2|(C_l^* \cap S) \setminus A|$. Since X_l is the union of lost clusters, it follows that $|X_l \cap S| \leq 2|X_l \cap S \setminus A| \leq 2|S \setminus A| \leq 2 \cdot 2.5(1+\kappa_1)\phi s = 5(1+\kappa_1)\phi s$. By Condition 6.2, each large C_l^* has $|C_l^* \cap S| \geq 0.75\frac{s}{n}n_l$. Thus, we can put a bound on the number of points n_l in a large cluster C_l^*, in terms of the number of representatives it has in S; in particular, $n_l \leq \frac{4}{3}\frac{n}{s}|C_l^* \cap S|$. Therefore, for X_l, which is the union of several large clusters (namely, the lost clusters), we have $X_l \leq \frac{20}{3}(1+\kappa_1)\phi n$.

It follows that, since X_{ex} and X_{sm} each contain at most ϕn points, $|N'| \geq (1-(2+\frac{20}{3}(1+\kappa_1))\phi)n$.

Since N' includes a particular set of $(1-(2+\frac{20}{3}(1+\kappa_1))\phi)n > (1-(9+7\kappa_1)\phi)n$ points, and $f^N_{1-(9+7\kappa_1)\phi}(K)$ counts the assignment distances of only the closest $(1-(9+7\kappa_1)\phi)n$ points to K, it follows that $f^{N'}(K)$ is an upper bound on $f^N_{1-(9+7\kappa_1)\phi}(K)$. □

Therefore, it suffices to prove the appropriate upper bound for $f^{N'}(K)$, and so we need only consider the covered clusters in what follows.

Lemma 10.

$$f^{N'}(K) \leq f^N_{1-\phi}(K^*) + \frac{8}{3}\frac{n}{s}\left[f^S_{1-2.5\phi}(K^*) + f^S_{1-2.5\kappa_1\phi}(K)\right]$$

Proof: First, we observe, as in the proof of Lemma 6, that the cost of assigning each member x of N' to the nearest median in K is at most the cost of moving x to the nearest median in K^*, plus the further cost of moving x from there to the nearest median in K. The cost of moving every point $x \in N'$ to the median nearest it in K^* is $f^{N'}(K^*)$, which is at most $f^N_{1-\phi}(K^*)$ since N' excludes the ϕn members of N furthest from K^*. For each (covered) C_l^*, the further cost of moving all the points from K_l^* to the nearest member of K is at most n_l times the distance from K_l^* to the nearest member of K. Therefore $f^{N'}(K) \leq f^{N'}(K^*) + \sum_{l=1}^{k_2} n_l d(K_l^*, K)$.

For a fixed l with C_l^* covered,

$$d(K_l^*, K) \leq \frac{1}{|C_l^* \cap A|} \sum_{x \in C_l^* \cap A} [d(x, K_l^*) + d(x, C(x))],$$

where $C(x)$ is the member of K nearest to x. This quantity is at most $\frac{2}{|C_l^* \cap S|} \sum_{x \in C_l^* \cap A} [d(x, K_l^*) + d(x, C(x))]$, since C_l is covered.

Define $Q_l = \sum_{x \in C_l^* \cap A} d(x, K_l^*)$ and $R_l = \sum_{x \in C_l^* \cap A} d(x, C(x))$. Then, combining these definitions with the above bounds, we have $f^{N'}(K) \leq f^{N'}(K^*) + \sum_{l=1}^{k_2} \frac{2n_l}{|C_l^* \cap S|}(Q_l + R_l)$. Recall that, by Condition 6.2, for all large C_l^*, $|C_l^* \cap S| \geq 0.75\frac{n_l s}{n}$, which implies $\frac{n_l}{|C_l^* \cap S|} \leq \frac{4}{3}\frac{n}{s}$. $f^{N'}(K) \leq f^{N'}(K^*) + \frac{8}{3}\frac{n}{s}\sum_{l=1}^{k_2}(Q_l + R_l)$.

Note that $\sum_{l=1}^{k_2} Q_l$ is the sum of $d(x, K^*)$, over all $x \in A$ that are members of covered clusters. This sum is therefore no more than $\sum_{x \in A} d(x, K^*)$, which is at most $f^S_{1-2.5\phi}(K^*)$, since all members of A are members of S assigned by the $(1 - 2.5\phi)$-solution K^* for S. By similar arguments it is possible to show that $\sum_{l=1}^{k_2} R_l \leq f^S_{1-2.5\kappa_1\phi}(K)$. We can conclude that

$$f^{N'}(K) \leq f^N_{1-\phi}(K^*) + \frac{8}{3}\frac{n}{s}\left[f^S_{1-2.5\phi}(K^*) + f^S_{1-2.5\kappa_1\phi}(K)\right]$$

□

Lemma 11.

$$f^S_{1-2.5\kappa_1\phi}(K) \leq \kappa_2 f^S_{1-2.5\phi}(K^*)$$

Proof: Let $\hat{K}$ denote the optimal $(1 - 2.5\phi)$-solution for S. Then, as guaranteed by the algorithm of Charikar et al. (2001), $f_{1-2.5\kappa_1\phi}(K) \leq \kappa_2 f_{1-2.5\phi}(\hat{K})$, but, since K^* is not necessarily the optimal $(1 - 2.5\phi)$-solution for S, we have $f_{1-2.5\phi}(\hat{K}) \leq f_{1-2.5\phi}(K^*)$, and the lemma follows. □

Fact 6.1.

$$f^S_{1-2.5\phi}(K^*) \leq 1.25\frac{s}{n} f^N_{1-\phi}(K^*)$$

Proof: By Condition 6.1, excluding the $2.5\phi s$ members of S that are furthest from K^* excludes all of $B_{b-1} \cup B_b$, since no bin contributes more than $1.25\phi s$ points to S. Using the same arguments as those used in the proofs of Lemma 3 and Fact 5.2, we can show that the cost of assigning to K^* the $(1 - 2.5\phi)s$ members of S closest to K^*, is at most $1.25\frac{s}{n} f^N_{1-\phi}(K^*)$. □

Theorem 5. *With probability at least* $1 - \frac{3}{4k^2}$,

$$f^N_{1-(9+7\kappa_1)\phi}(K) \leq \left[1 + \frac{10}{3}(1 + \kappa_2)\right] f^N_{1-\phi}(K^*)$$

Proof: The theorem follows immediately from Lemmas 8 through 11 and Fact 6.1. □

7. Lower bounds

The following example illustrates the trade-off between sample size and the number of points that must be excluded from the clustering (or the minimum cluster size). Assume we are trying to cluster using a sample of size s, and that we want a solution whose $(1-\delta)$-cost is competitive with the cost of the optimal $(1-\delta')$-solution. (We will see how δ is a function of s, but it certainly must be at least δ'.)

Consider the class of data sets consisting of:

- $k-1$ "small" clusters, each consisting of n/s co-located points.
- $\delta' n$ singleton points.
- One "big" cluster of $n - \frac{(k-1)n}{s} - \delta' n$ points that are all co-located.

Assume all inter-point distances are infinite (or very large), except distances between pairs of co-located points, which are zero.

Then the optimal $(1-\delta')$-cost is 0. We will say our sample "misses" a cluster if the cluster has no members in the sample. For each small cluster that is missed, our solution will have to exclude an additional n/s points in order to have cost competitive with the best $(1-\delta')$-cost.

We expect each small cluster to contribute one point to the sample. Therefore, each cluster has constant probability of being missed, and so we expect the sample to miss $\Omega(k)$ small clusters. Even if we repeat our sampling a constant number of times, it is still likely that every sample will miss $\Omega(k)$ small clusters.

We conclude that either δ must be at least $\delta' + \Omega(\frac{k}{s})$, or our optimum solution must disallow clusters of size $O(\frac{n}{s})$ or smaller.

Our upper bounds match these lower bounds to within a factor of $\Theta(\log \frac{k}{\epsilon})$. This factor ensures that clusters are represented in (roughly) equal proportion; this condition is required for our analysis and is naturally somewhat stronger than the lower bound condition that each cluster have at least one representative in the sample. It is straightforward to extend our results to consider samples whose size depends on n; for example if we use samples of size $\Theta(\sqrt{n}\log n)$, we can cluster a data set in which the optimum solution has no cluster of size smaller than $\sqrt{n}$.

Acknowledgments

The authors gratefully acknowledge an anonymous reviewer who made extremely helpful editing suggestions, and who recommended the inclusion of the algorithm in Section 5 and the simplifications evident in Section 6, which vastly improved the paper.

Notes

1. The $\Omega(nk)$ running-time lower bound given by Mettu and Plaxton (2002) reflects this problem, as well.
2. We could reduce running time to get a larger approximation ratio using the algorithm of Guha et al. (2000) instead.
3. This ordering and division into bins are merely analytical tools in this proof, and are not computed by the algorithm.
4. The idea of "moving" points is only a tool for analyzing the assignment distances: if there is a path over which a point x can travel to its center $C(x)$, then the length of this path is certainly an upper bound on the assignment distance of x.

References

Alon, N., Dar, S., Parnas, M., & Ron, D. (2000). Testing of clustering. In *Proc. FOCS*.

Arora, S., Raghavan, P., & Rao, S. (1998). Approximation schemes for euclidean k-medians and related problems. In *Proc. STOC*.

Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., & Pandit, V. (2001). Local search heuristics for k-median and facility location problems. In *Proc. STOC*.

Bansal, N., Blum, A., & Chawla, S. (2002). Correlation clustering. In *Proc. FOCS*.

Bradley, P. S., Fayyad, U. M., & Reina, C. (1998). Scaling clustering algorithms to large databases. In *Proc. KDD*.

Charikar, M., Chekuri, C., Feder, T., & Motwani, R. (1997). Incremental clustering and dynamic information retrieval. In *Proc. STOC*.

Charikar, M., & Guha, S. (1999). Improved combinatorial algorithms for the facility location and k-median problems. In *Proc. FOCS*.

Charikar, M., Guha, S., Tardos, É., & Shmoys, D. (2002). A constant-factor approximation algorithm for the k-median problem. *Journal of Computer and System Sciences, 65:1*, 129–149.

Charikar, M., Khuller, S., Mount, D., & Narasimhan, G. (2001). Algorithms for facility location problems with outliers. In *Proc. SODA*.

Charikar, M., & Panigrahy, R. (2001). Clustering to minimize the sum of cluster diameters. In *Proc. 33rd STOC*.

Dasgupta, S. (1999). Learning mixtures of gaussians. In *Proc. FOCS*.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1984). Maximum-likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc. Ser. B,* 39.

Duda, R. O., & Hart, P. E. (1972). *Pattern classification and scene analysis*. Wiley.

Feder, T., & Greene, D. (1988). Optimal algorithms for approximate clustering. In *Proc. STOC*.

Gonzalez, T. F. (1985). Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science, 38*, 2–3.

Guha, S. (2000). Approximation algorithms for facility location problems. Ph.D. Thesis, Stanford University.

Guha, S., Mishra, N., Motwani, R., & O'Callaghan, L. (2000). Clustering data streams. In *Proc. FOCS*.

Guha, S., Rastogi, R., & Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. In *Proc. SIGMOD*.

Hartigan, J. A. (1975). *Clustering algorithms*. John Wiley and Sons.

Hochbaum, D. S., & Shmoys, D. B. (1985). A best possible approximation algorithm for the k-center problem. *Mathematics of Operations Research, 10*, 180–184.

Indyk, P. (1999). Sublinear time algorithms for metric space problems. In *Proc. STOC*.

Jain, K., Mahdian, M., Markakis, E., Saberi, A., & Vazirani, V. (to appear). Greedy facility location algorithms analyzed using dual-fitting with factor-revealing LP. *Journal of the ACM*.

Jain, K., & Vazirani, V. (1999). Primal-dual approximation algorithms for metric facility location and k-median problems. In *Proc. FOCS*.

Joze-Hkajavi, N., & Salem, K. (1998). Two-phase clustering of large datasets. Tech. rpt. CS-98-27, Univ. of Waterloo.

Lin, J.-H., & Vitter, J. S. (1992). ϵ-Approximations with minimum packing constraint violation. In *Proc. STOC*.

Marroquin, J., & Girosi, F. (1993). Some extensions of the *k*-means algorithm for image segmentation and pattern recognition. *AI Memo 1390, MIT AI Lab.*.

Mettu, R., & Plaxton, C. G. (2002). Optimal time bounds for approximate clustering. In *Proc. 18th UAI*.

Meyerson, A. (2001). Online facility location. In *Proc. FOCS*.

Mishra, N., Oblinger, D., & Pitt, L. (2001). Sublinear time approximate clustering. In *Proc. SODA*.

Palmer, C. R., & Faloutsos, C. (2000). Density biased sampling: An improved method for data mining and clustering. In *Proc. SIGMOD*.

Pelleg, D., & Moore, A. W. (1999). Accelerating exact *k*-means algorithms with geometric reasoning. In *Proc. KDD*.

Pitt, L., & Reinke, R. E. (1988). Criteria for polynomial-time (conceptual) clustering. *Machine Learning, 2*, 4.

Redner, R. A., & Walker, H. F. (1984). Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review, 26*, 2.

Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. Tech rept. No. 00-034, Univ. of Minn.

Thorup, M. (2001). Quick *k*-median, *k*-center, and facility location for sparse graphs. In *Proc. ICALP*.

Young, N. E. (2000). *k*-medians, facility location, and the Chernoff-Wald bound. In *Proc. SODA* (pp. 86–95).

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. In *Proc. SIGMOD* (pp. 103–114).

Received February 3, 2003
Revised January 12, 2004
Accepted January 12, 2004
Final manuscript March 1, 2004