Stability yields a PTAS for k-Median and k-Means Clustering

Pranjal Awasthi
Carnegie Mellon University
pawasthi@cs.cmu.edu

Avrim Blum
Carnegie Mellon University
avrim@cs.cmu.edu

Or Sheffet
Carnegie Mellon University
osheffet@cs.cmu.edu

***Abstract*—We consider k-median clustering in finite metric spaces and k-means clustering in Euclidean spaces, in the setting where k is part of the input (not a constant). For the k-means problem, Ostrovsky et al. [18] show that if the optimal $(k-1)$-means clustering of the input is more expensive than the optimal k-means clustering by a factor of $1/\epsilon^2$, then one can achieve a $(1+f(\epsilon))$-approximation to the k-means optimal in time polynomial in n and k by using a variant of Lloyd's algorithm. In this work we substantially improve this approximation guarantee. We show that given only the condition that the $(k-1)$-means optimal is more expensive than the k-means optimal by a factor $1+\alpha$ for *some* constant $\alpha > 0$, we can obtain a PTAS. In particular, under this assumption, for any $\epsilon > 0$ we achieve a $(1+\epsilon)$-approximation to the k-means optimal in time polynomial in n and k, and exponential in $1/\epsilon$ and $1/\alpha$. We thus decouple the strength of the assumption from the quality of the approximation ratio. We also give a PTAS for the k-median problem in finite metrics under the analogous assumption as well. For k-means, we in addition give a randomized algorithm with improved running time of $n^{O(1)}(k \log n)^{\mathrm{poly}(1/\epsilon, 1/\alpha)}$.**

Our technique also obtains a PTAS under the assumption of Balcan et al. [4] that all $(1+\alpha)$ approximations are δ-close to a desired target clustering, in the case that all target clusters have size greater than δn and $\alpha > 0$ is constant. Note that the motivation of Balcan et al. [4] is that for many clustering problems, the objective function is only a proxy for the true goal of getting close to the target. From this perspective, our improvement is that for k-means in Euclidean spaces we reduce the distance of the clustering found to the target from $O(\delta)$ to δ when all target clusters are large, and for k-median we improve the "largeness" condition needed in [4] to get exactly δ-close from $O(\delta n)$ to δn. Our results are based on a new notion of clustering stability.

1. Introduction

Clustering is a well-studied task, arising in numerous areas from computer vision to computational biology to distributed computing. Generally speaking, the goal of clustering is to partition n given data objects into k groups that share some commonality. Operationally, clustering is often performed by viewing the data as points in a metric space and then optimizing some natural objective over them. In this paper, we consider two popular such objectives, k-median and k-means. Both measure a k-partition by choosing a special point for each cluster, called the *center*, and define the *cost* of a clustering as a function of the distances between the data points and their respective centers. In the k-median case, the cost is the sum of the distances of the points to their centers, and in the k-means case, the cost is the sum of these distances squared. The k-median objective is typically studied for data in a finite metric (complete weighted graph satisfying triangle inequality) over the n data points; k-means clustering is typically studied for n points in a (finite dimensional) Euclidean space. Both objectives are known to be NP-hard (we view k as part of the input and not a constant, though even the 2-means problem in Euclidean space was recently shown to be NP-hard [8]). For k-median in a finite metric, there is a known $(1+1/e)$-hardness of approximation result [14] and substantial work on approximation algorithms [11], [7], [2], [14], [9], with the best guarantee a $3+\epsilon$ approximation. For k-means in a Euclidean space, there is also a vast literature of approximation algorithms [17], [3], [9], [10], [12], [15] with the best guarantee a constant-factor approximation if polynomial dependence on k and the dimension d is desired.[1]

Ostrovsky et al. [18] proposed an interesting condition under which one can achieve better k-means approximations in time polynomial in n and k. They consider k-means instances where the optimal k-clustering has cost noticeably smaller than the cost of any $(k-1)$-clustering, motivated by the idea that "if a near-optimal k-clustering can be achieved by a partition into fewer than k clusters, then that smaller value of k should be used to cluster the data" [18]. Under the assumption that the ratio of the cost of the optimal $(k-1)$-means clustering to the cost of the optimal k-means clustering is at least $\max\{100, 1/\epsilon^2\}$, Ostrovsky et al. show that one can obtain a $(1+f(\epsilon))$-approximation for k-means in time polynomial in n and k, by using a variant on Lloyd's algorithm. In this paper, we substantially improve on this approximation guarantee. We show that under the much weaker assumption that the ratio of these costs is just at least $(1+\alpha)$ for *some* constant $\alpha > 0$, we can achieve a PTAS: namely, $(1+\epsilon)$-approximate the k-means optimum, for any constant $\epsilon > 0$. Our approximation scheme runs in time which is $\mathrm{poly}(n, k)$ and exponential only in $1/\epsilon$ and $1/\alpha$. Thus, we decouple the strength of the assumption from the quality of the conclusion, and in the process allow the assumption to be substantially weaker. For k-means clustering we in addition give a randomized algorithm with improved running time $n^{O(1)}(k \log n)^{\mathrm{poly}(1/\epsilon, 1/\alpha)}$.

Balcan et al. [4], motivated by the fact that objective functions are often just a proxy for the underlying goal of getting the data clustered correctly, propose clustering instances that satisfy the condition that all $(1+\alpha)$ approxi-

[1] If k is constant, then k-median in finite metrics can be trivially solved in polynomial time and there is a PTAS known for k-means (and k-median) in Euclidean space [16]. There is also a PTAS known for low-dimensional Euclidean spaces (dimension at most $\log \log n$) [1], [12].

mations to the given objective (e.g., k-median or k-means) are δ-close, in terms of how points are partitioned, to a target clustering (such as a correct clustering of proteins by function or a correct clustering of images by who is in them). This can be viewed as an assumption made *implicitly* when considering approximation algorithms for problems of this nature where the true goal is to get close to the target. Balcan et al. show that for any α and δ, given an instance satisfying this property for k-median or k-means objectives, one can in fact efficiently produce a clustering that is $O(\delta/\alpha)$-close to the target clustering (so, $O(\delta)$-close for any constant $\alpha > 0$), *even though obtaining a* $1+\alpha$ *approximation to the objective is* NP*-hard* for $\alpha < \frac{1}{e}$, and remains hard even under this assumption. Thus they show that one can approximate the target even though it is hard to approximate the objective. One interesting question that has remained is the approximability of the objectives when all target clusters are large compared to δn, since the hardness of approximation requires allowing small clusters.[2] Here, we show that for both k-median and k-means objectives, if all clusters contain more than δn points, then for any constant $\alpha > 0$ we can in fact get a PTAS. Thus, we (nearly) resolve the approximability of these objectives under this condition. Note that under this condition, this further implies finding a δ-close clustering (setting $\epsilon = \alpha$). Thus, we also extend the results of Balcan et al. [4] in the case of large clusters and constant α by getting exactly δ-close for both k-median and k-means objectives. (In [4] this exact closeness was achieved for the k-median objective but needed a somewhat larger $O(\delta n(1+1/\alpha))$ minimum cluster size requirement).

Our algorithmic results are achieved by examining implications of a property we call *weak deletion-stability* that is implied by both the separation condition of Ostrovsky et al. [18] as well as (when target clusters are large) the stability condition of Balcan et al. [4]. In particular, an instance of k-median/k-means clustering satisfies weak deletion-stability if in the optimal solution, deleting any of the centers c_i^* and assigning all points in cluster i instead to one of the remaining $k-1$ centers c_j^*, results in an increase in the k-median/k-means cost by an (arbitrarily small) constant factor.

We also show that weak deletion-stability still allows for NP-hard instances and that no FPTAS is possible as well (unless $\mathsf{P} = \mathsf{NP}$). Thus, our algorithm, whose running time is $(nk)^{\mathrm{poly}(1/\epsilon, 1/\beta)}$, is optimal in the sense that the super-polynomial dependence on $1/\epsilon$ and $1/\beta$ is unavoidable.

After presenting notation and preliminaries in Section 2, in Section 3 we introduce weak deletion-stability and relate it to the stability notions of [18] and [4]. We then define another property of a clustering being β*-distributed* which, while not so intuitive, we show is implied by weak deletion-stability and will be the actual condition that our algorithms will use. We then go on to prove that being β-distributed suffices to give a PTAS for k-median in Section 4. We extend the algorithm to k-means clustering in Section 5, where we also introduce a randomized version whose run-time is bounded by $n^3\left((\log(n)\cdot k)\right)^{\mathrm{poly}(1/\epsilon,1/\beta)}$. We conclude with discussion and open problems in Section 6.

2. Notation and Preliminaries

We are given a set S of n points. When discussing k-median, we assume the n points reside in a finite metric space, and when discussing k-means, we assume they all reside in a finite dimensional Euclidean space. We denote $d : S \times S \rightarrow \mathbb{R}_{\geq 0}$ as the distance function. A solution to the k-median objective partitions the n points into k disjoint subsets, $C_1, C_2, \ldots, C_k$ and assigns a center c_i for each subset. The k-median cost of this partition is then measured by $\sum_{i=1}^k \sum_{x \in C_i} d(x, c_i)$. A solution to the k-means objective again gives a k-partition of the n data points, but now we may assume uses the center of mass, $\mu_{C_i} = \frac{1}{|C_i|}\sum_{x\in C_i} x$, as the center of the C_i. We then measure the k-means cost of this clustering by $\sum_{i=1}^k \sum_{x\in C_i} d^2(x, \mu_{C_i}) = \sum_{i=1}^k \sum_{x\in C_i} \|x - \mu_{C_i}\|^2$.

The optimal clustering (w.r.t. to either the k-median or the k-means objective) is denoted as $\mathcal{C}^* = \{C_1^*, C_2^*, \ldots, C_k^*\}$, and its cost is denoted as OPT. The centers used in the optimal clustering are denoted as $\{c_1^*, c_2^*, \ldots, c_k^*\}$. Clearly, given the optimal clustering, we can find the optimal centers (either by brute-force checking all possible points for k-median, or by $c_i^* = \mu_{C_i^*}$ for k-means). Alternatively, given the optimal centers, we can assign each x to its nearest center, thus obtaining the optimal clustering. Thus, we use $\mathcal{C}^*$ to denote both the optimal k-partition, and the optimal list of k centers. We use OPT_i to denote the contribution of the cluster i to OPT, that is $\mathsf{OPT}_i = \sum_{x\in C_i^*} d(x, c_i^*)$ in the k-median case, or $\mathsf{OPT}_i = \sum_{x\in C_i^*} d^2(x, c_i^*)$ in the k-means case.

3. Stability Properties

As mentioned above, our results are achieved by exploiting implications of a stability condition we call weak deletion-stability, and in particular an implication we call being β-distributed. In this section we define weak deletion-stability and of being β-distributed, relate weak deletion-stability to conditions of Ostrovsky et al. [18] and Balcan et al. [4], and show that weak deletion-stability implies the clustering is β-distributed. In Sections 4 and 5 we use the property of being β-distributed to obtain a PTAS.[3]

Definition 3.1. *For* $\alpha > 0$*, a* k*-median/*k*-means instance satisfies* $(1+\alpha)$ weak deletion-stability, *if it has the following property. Let* $\{c_1^*, c_2^*, \ldots, c_k^*\}$ *denote the centers in the optimal* k*-median/*k*-means solution. Let* OPT *denote the optimal* k*-median/*k*-means cost and let* $\mathsf{OPT}^{(i\to j)}$ *denote the cost of the clustering obtained by removing* c_i^* *as a center*

[2] In fact, as shown in [19], the k-median algorithm in [4] for the case that clusters are sufficiently large compared to $\delta n(1+1/\alpha)$ achieves a better constant-factor approximation. Note that δ need not be a constant.

[3] Technically, we could skip the "middleman" of weak deletion-stability and just define the property of being β-distributed as our main stability notion, but weak deletion-stability is a more intuitive condition.

and assigning all its points instead to c_j^*. Then for any $i \neq j$, it holds that

$$\mathsf{OPT}^{(i \to j)} > (1+\alpha)\mathsf{OPT}$$

We use weak deletion-stability via the following implication we call being β-distributed.

Definition 3.2. *For $\beta > 0$, a k-median instance is β-*distributed *if for any center c_i^* of the optimal clustering and any data point $x \notin C_i^*$, it holds that*

$$d(x, c_i^*) \geq \beta \cdot \frac{\mathsf{OPT}}{|C_i^*|}.$$

A k-means instance is β-distributed if for any such c_i^* and $x \notin C_i^*$, it holds that

$$d^2(x, c_i^*) \geq \beta \cdot \frac{\mathsf{OPT}}{|C_i^*|}$$

We prove that $(1+\alpha)$ weak deletion-stability implies the clustering is $\alpha/2$-distributed for k-median ($\alpha/4$-distributed for k-means) in Theorem 3.5 below. First, however, we relate weak deletion-stability to the conditions considered in [18] and [4].

A. ORSS-Separability

Ostrovsky, Rabani, Schulman and Swamy [18] define a clustering instance to be ϵ-separated if the optimal k-means solution is cheaper than the optimal $(k-1)$-means solution by at least a factor ϵ^2. For a given objective (k-means or k-median) let us use $\mathsf{OPT}_{(k-1)}$ to denote the cost of the optimal $(k-1)$-clustering. Introducing a parameter $\alpha > 0$, say a clustering instance is $(1+\alpha)$-*ORSS separable* if

$$\frac{\mathsf{OPT}_{(k-1)}}{\mathsf{OPT}} > 1 + \alpha$$

If an instance satisfies $(1+\alpha)$-ORSS separability then all $(k-1)$ clusterings must have cost more than $(1+\alpha)\mathsf{OPT}$ and hence it is immediately evident that the instance will also satisfy $(1+\alpha)$-weak deletion-stability. Hence we have the following claim:

Claim 3.3. *Any $(1+\alpha)$-ORSS separable k-median/k-means instance is also $(1+\alpha)$-weakly deletion stable.*

B. BBG-Stability

Balcan, Blum, and Gupta [4] (see also Balcan and Braverman [5] and Balcan, Röglin, and Teng [6]) consider a notion of stability to approximations motivated by settings in which there exists some (unknown) target clustering $\mathcal{C}^{target}$ we would like to produce. Balcan et al. [4] define a clustering instance to be $(1+\alpha, \delta)$ approximation-stable with respect to some objective Φ (such as k-median or k-means), if any k-partition whose cost under Φ is at most $(1+\alpha)\mathsf{OPT}$ agrees with the target clustering on all but at most δn data points. That is, for any $(1+\alpha)$ approximation $\mathcal{C}$ to objective Φ, we have $\min_{\sigma \in S_k} \sum_i |C_i^{target} - C_{\sigma(i)}| \leq \delta n$ (here, σ is simply a matching of the indices in the target clustering to those in $\mathcal{C}$). In general, δn may be larger than the smallest target cluster size, and in that case approximation-stability need not imply weak deletion-stability (not surprisingly since [4] show that k-median and k-means remain hard to approximate). However, when all target clusters have size greater than δn (note that δ need not be a constant) then approximation-stability indeed also implies weak deletion-stability, allowing us to get a PTAS (and thereby δ-close to the target) when $\alpha > 0$ is a constant.

Claim 3.4. *A k-median/k-means clustering instance that satisfies $(1+\alpha, \delta)$ approximation-stability, and in which all clusters in the target clustering have size greater than δn, also satisfies $(1+\alpha)$ weak deletion-stability.*

Proof: Consider an instance of k-median/k-means clustering which satisfies $(1+\alpha, \delta)$ approximation-stability. As before, let $\{c_1^*, c_2^*, \ldots, c_k^*\}$ be the centers in the optimal solution and consider the clustering $C^{(i \to j)}$ obtained by no longer using c_i^* as a center and instead assigning each point from cluster i to c_j^*, making the ith cluster empty. The distance of this clustering from the target is defined as $\frac{1}{n} \min_{\sigma \in S_k} \sum_{i'} |C_{i'}^{target} - C_{\sigma(i')}^{(i \to j)}|$. Since $C^{(i \to j)}$ has only $(k-1)$ nonempty clusters, one of the target clusters must map to an empty cluster under any permutation σ. Since by assumption, this target cluster has more than δn points, the distance between C^{target} and $C^{(i \to j)}$ will be greater than δ and hence by the BBG stability condition, the k-median/k-means cost of $C^{(i \to j)}$ must be greater than $(1+\alpha)\mathsf{OPT}$. ∎

C. Weak Deletion-Stability implies β-distributed

We show now that weak deletion-stability implies the instance is β-distributed.

Theorem 3.5. *Any $(1+\alpha)$-weakly deletion-stable k-median instance is $\frac{\alpha}{2}$-distributed. Any $(1+\alpha)$-weakly deletion-stable k-means instance is $\frac{\alpha}{4}$-distributed.*

Proof: Fix any center in the optimal k-clustering, c_i^*, and fix any point p that does not belong to the C_i^* cluster. Denote by C_j^* the cluster that p is assigned to in the optimal k-clustering. Therefore it must hold that $d(p, c_j^*) \leq d(p, c_i^*)$. Consider the clustering obtained by deleting c_i^* from the list of centers, and assigning each point in C_i^* to C_j^*. Since the instance is $(1+\alpha)$-weakly deletion-stable, this should increase the cost by at least $\alpha\mathsf{OPT}$.

Suppose we are dealing with a k-median instance. Each point $x \in C_i^*$ originally pays $d(x, c_i^*)$, and now, assigned to c_j^*, it pays $d(x, c_j^*) \leq d(x, c_i^*) + d(c_i^*, c_j^*)$. Thus, the new cost of the points in C_i^* is upper bounded by $\sum_{x \in C_i^*} d(x, c_j^*) \leq \mathsf{OPT}_i + |C_i^*| d(c_i^*, c_j^*)$. As the increase in cost is lower bounded by $\alpha\mathsf{OPT}$ and upper bounded by $|C_i^*| d(c_i^*, c_j^*)$, we deduce that $d(c_i^*, c_j^*) > \alpha \frac{\mathsf{OPT}}{|C_i^*|}$. Observe that triangle inequality gives that $d(c_i^*, c_j^*) \leq d(c_i^*, p) + d(p, c_j^*) \leq 2d(c_i^*, p)$, so we have that $d(c_i^*, p) > (\alpha/2) \frac{\mathsf{OPT}}{|C_i^*|}$.

Suppose we are dealing with a Euclidean k-means instance. Again, we have created a new clustering by assigning all points in C_i^* to the center c_j^*. Thus, the cost of transitioning from the optimal k-clustering to this new $(k-1)$-clustering, which is at least $\alpha\mathsf{OPT}$, is upper bounded by $\sum_{x\in C_i^*}\|x-c_j^*\|^2-\|x-c_i^*\|^2$. As $c_i^*=\mu_{C_i^*}$, it follows that this bound is *exactly* $\sum_{x\in C_i^*}\|c_j^*-c_i^*\|^2=|C_i^*|d^2(c_i^*,c_j^*)$, see [13] (§2, Theorem 2). It follows that $d^2(c_i^*,c_j^*)>\alpha\frac{\mathsf{OPT}}{|C_i^*|}$. As before, $d^2(c_i^*,c_j^*)\le\left(d(c_i^*,p)+d(p,c_j^*)\right)^2\le 4d^2(c_i^*,p)$, so $d^2(c_i^*,p)>\frac{\alpha}{4}\frac{\mathsf{OPT}}{|C_i^*|}$. ■

D. NP-*hardness under weak deletion-stability*

Finally, we would like to point out that NP-hardness of the k-median problem in maintained even if we restrict ourselves only to weakly deletion-stable instances. Also the reduction sketched below uses only integer poly-size distances, and hence rules out the existence of a FPTAS for the problem, unless $\mathsf{P}=\mathsf{NP}$. In addition, the reduction can be modified to show that NP-hardness is maintained under the conditions studied in [18] and [4].

Theorem 3.6. *For any constant $\alpha>0$, finding the optimal k-median clustering of $(1+\alpha)$-weakly deletion-stable instances is* NP-*hard.*

Proof Sketch. Fix any constant $\alpha>0$. We give a 1-1 poly-time reduction from Set-Cover to $(1+\alpha)$-weakly deletion-stable k-median instances. Under standard notation, we assume our input consists of n subsets of a given universe of size m, for which we seek a k-cover. We reduce such an instance to a k-median instance over $m+k(n+4\beta km)$ points. We start with the usual reduction of Set-Cover to an instance with m points representing the items of the universe and n points representing all possible sets. Fix integer $D\gg 1$ to be chosen later. If j belongs to the ith set, fix the distance $d(i,j)=D$, otherwise we fix the distance $d(i,j)=D+1$, and between any two set-points we fix the distance to be 1. (The distance between any two item points is shortest-path distance.) However, we augment the n set-points with additional $2mD$ points, setting the distance between all of the $(n+2mD)$ points as 1. Furthermore, we replicate k copies of the augmented set-point, all connected only via the m-item points.

Observe that each of the k copies of our augmented set-points components contains many points, and all points outside this copy are of distance $\ge D$ from it. Therefore, in the optimal k-median solution, each center resides in one unique copy of the augmented set-points. Now, if our Set-Cover instance has a k-cover, then we can pick the respective centers and have an optimal solution with cost exactly $k(n+2mD-1)+mD$. Otherwise, no k sets cover all m items, so for any k centers, some item-point must have distance $D+1$ from its center, and so the cost of any k-partition is $\ge k(n+2mD-1)+mD+1$. Furthermore, the resulting instance is $(1+\alpha)$ weakly deletion-stable as using one center from each augmented set-point results in a k-median solution of cost $\le m(D+1)+k(n+2mD-1)$. Hence, OPT is atmost this quantity. However, in any $k-1$ clustering, one of the set-points must pay a high cost and hence $\mathsf{OPT}_{(k-1)}\ge(m-1)D+(k-1)(n+2mD-1)+(n+2mD)D$. One can choose D large enough so that this cost is at least $(1+\alpha)\mathsf{OPT}$. □

4. A PTAS for any β-distributed k-Median Instance

We now present the algorithm for finding a $(1+\epsilon)$-approximation of the k-median optimum for β-distributed instances. First, we comment that using a standard doubling technique, we can assume we approximately know the value of OPT.[4] Our algorithm works if instead of OPT we use a value v s.t. $\mathsf{OPT}\le v\le(1+\epsilon/2)\mathsf{OPT}$, but for ease of exposition, we assume that the exact value of OPT is known.

Below, we informally describe the algorithm for a special case of β-distributed instances in which no cluster dominates the overall cost of the optimal clustering. Specifically, we say a cluster C_i^* in the optimal k-median clustering $\mathcal{C}^*$ (hereafter also referred to as the target clustering) is *cheap* if $\mathsf{OPT}_i\le\frac{\beta\epsilon\mathsf{OPT}}{32}$, otherwise, we say C_i^* is *expensive*. Note that in any event, there can be at most a constant $(\frac{32}{\beta\epsilon})$ number of expensive clusters.

Algorithm Intuition: The intuition for our algorithm and for introducing the notion of cheap clusters is the following. Pick some cluster C_i^* in the optimal k-median clustering. Since the instance is β-distributed, any $x\notin C_i^*$ is far from c_i^*, namely, $d(x,c_i^*)>\beta\frac{\mathsf{OPT}}{|C_i^*|}$. In contrast, the average distance of $x\in C_i^*$ from c_i^* is $\frac{\mathsf{OPT}_i}{|C_i^*|}$. Thus, if we focus on a cluster whose contribution, OPT_i, is no more than, say, $\frac{\beta}{100}\mathsf{OPT}$, we have that c_i^* is 100 times closer, on *average*, to the points of C_i^* than to the points outside C_i^*. Furthermore, using the triangle inequality we have that any two "average" points of C_i^* are of distance at most $\frac{2\beta}{100}\frac{\mathsf{OPT}}{|C_i^*|}$, while the distance between any such "average" point and any point outside of C_i^* is at least $\frac{99\beta}{100}\frac{\mathsf{OPT}}{|C_i^*|}$. So, if we manage to correctly guess the size s of a cheap cluster, we can set a radius $r=\Theta\left(\beta\frac{\mathsf{OPT}}{s}\right)$ and collect data-points according to the size and intersection of the r-balls around them. We note that this use of balls with an inverse relation between size and radius is similar to that in the min-sum clustering algorithm of [5].

Note that in the general case we might have up to $\frac{32}{\beta\epsilon}$ expensive clusters. We handle them by brute force guessing their centers. In Subsection 4-A, we present the algorithm for clustering β-distributed instances of k-median under the assumption that for all the expensive clusters we have made the correct guess for their cluster centers. The algorithm populates a list $\mathcal{Q}$, where each element in this list is a subset of points. Ideally, each subset is contained in some target cluster, yet we might have a few subsets with points from two or more target clusters. The first stage of the algorithm is to add components into $\mathcal{Q}$, and the second stage is to find k good components in $\mathcal{Q}$, and use these k components to retrieve a clustering with low cost.

[4]Instead of doubling from 1, we can alternatively run an off-the-shelf 5-approximation of OPT, which will return a value $v\le 5\mathsf{OPT}$.

Since we do not have many expensive clusters, we can run the algorithm for all possible guesses for the centers of the expensive clusters and choose the solution which has the minimum cost. The analysis below shows that one such guess will lead to a solution of cost at most $(1+\epsilon)\mathsf{OPT}$. Later, in Section 5, when we deal with k-means in Euclidean space, we use sampling techniques, similar to those of Kumar et al. [16] and Ostrovsky et al. [18], to get good substitutes for the centers of the expensive clusters. Note however an important difference between the approach of [16], [18] and ours. While they sample points from all k clusters, we sample points only for the $O(1)$ expensive clusters. As a result, the runtime of the PTAS of [16], [18] has exponential dependence in k, while ours has only a polynomial dependence in k.

A. Clustering β-distributed Instances

1) **Initialization Stage:** Set $\mathcal{Q} \leftarrow \mathcal{Q}_{init}$.
2) **Population Stage:** For $s = n, n-1, n-2, \ldots, 1$ do:
   a) Set $r = \frac{\beta\mathsf{OPT}}{4s}$.
   b) Remove any point x such that $d(x, \mathcal{Q}) < 2r$.
   (Here, $d(x, \mathcal{Q}) = \min_{T\in\mathcal{Q}; y\in T} d(x, y)$.)
   c) For any remaining data point x, denote the set of data points whose distance from x is at most r, by $B(x, r)$. Connect any two remaining points a and b if:
   (i) $d(a, b) \le r$, (ii) $|B(a, r)| > \frac{s}{2}$ and (iii) $|B(b, r)| > \frac{s}{2}$.
   d) Let T be a connected component of size $> \frac{s}{2}$. Then:
      i) Add T to $\mathcal{Q}$. (That is, $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{T\}$.)
      ii) Define the set $B(T) = \{x : d(x, y) \le 2r \text{ for some } y \in T\}$. Remove the points of $B(T)$ from the instance.
3) **Centers-Retrieving Stage:** For any choice of k components $T_1, T_2, \ldots, T_k$ out of $\mathcal{Q}$ (we later show that $|\mathcal{Q}| < k + O(1/\beta)$)
   a) Find the best center c_i for $T_i \cup B(T_i)$. That is $c_i = \arg\min_{p\in T_i\cup B(T_i)} \sum_{x\in T_i\cup B(T_i)} d(x, p)$.[a]
   b) Partition all n points according to the nearest point among the k centers of the current k components.
   c) If a clustering of cost at most $(1+\epsilon)\mathsf{OPT}$ is found – output these k centers and halt.

[a] This can be done before fixing the choice of k components out of $\mathcal{Q}$.

Figure 1. The algorithm to obtain a PTAS for β-distributed instances of k-median.

The algorithm is presented in Figure 1. In this section we assume that at the beginning, the list $\mathcal{Q}$ is initialized with $\mathcal{Q}_{init}$ which contains the centers of all the expensive clusters. In general, the algorithm will be run several times with $\mathcal{Q}_{init}$ containing different guesses for the centers of the expensive clusters. Before going into the proof of correctness of the algorithm, we introduce another definition. We define the *inner ring* of C_i^* as the set $\left\{x;\ d(x, c_i^*) \le \frac{\beta\mathsf{OPT}}{8|C_i^*|}\right\}$. Note the following fact:

Fact 4.1. *If C_i^* is a cheap cluster, then no more than an $\epsilon/4$ fraction of its points reside outside the inner ring. In particular, at least half of a cheap cluster is contained within the inner ring.*

Proof: This follows from Markov's inequality. If more than $(\epsilon/4)|C_i^*|$ points are outside of the inner ring, then $\mathsf{OPT}_i > \frac{\epsilon|C_i^*|}{4} \cdot \frac{\beta\mathsf{OPT}}{8|C_i^*|} = \beta\epsilon\mathsf{OPT}/32$. This contradicts the fact that C_i^* is cheap. ∎

Our high level goal is to show that for any cheap cluster C_i^* in the target clustering, we insert a component T_i that is contained within C_i^*, and furthermore, contains only points that are close to c_i^*. It will follow from the next claims that the component T_i is the one that contains points from the inner ring of C_i^*. We start with the following Lemma which we will utilize a few times.

Lemma 4.2. *Let T be any component added to $\mathcal{Q}$. Let s be the stage in which we add T to $\mathcal{Q}$. Let C_i^* be any cheap cluster s.t. $s \ge |C_i^*|$. Then (a) T does not contain any point z s.t. the distance $d(c_i^*, z)$ lies within the range $\left[\frac{\beta}{2}\frac{\mathsf{OPT}}{|C_i^*|}, \frac{3\beta}{4}\frac{\mathsf{OPT}}{|C_i^*|}\right]$, and (b) T cannot contain both a point p_1 s.t. $d(c_i^*, p_1) < \frac{\beta}{2}\frac{\mathsf{OPT}}{|C_i^*|}$ and a point p_2 s.t. $d(c_i^*, p_2) > \frac{3\beta}{4}\frac{\mathsf{OPT}}{|C_i^*|}$.*

Proof: We prove (a) by contradiction. Assume T contains a point z s.t. $\frac{\beta}{2}\frac{\mathsf{OPT}}{|C_i^*|} \le d(c_i^*, z) \le \frac{3\beta}{4}\frac{\mathsf{OPT}}{|C_i^*|}$. Set $r = \frac{\beta\mathsf{OPT}}{4s} \le \frac{\beta\mathsf{OPT}}{4|C_i^*|}$, just as in the stage when T was added to $\mathcal{Q}$, and let p be any point in the ball $B(z, r)$. Then by the triangle inequality we have that $d(c_i^*, p) \ge d(c_i^*, z) - d(z, p) \ge \frac{\beta}{4}\frac{\mathsf{OPT}}{|C_i^*|}$, and similarly $d(c_i^*, p) \le d(c_i^*, z) + d(z, p) \le \frac{\beta\mathsf{OPT}}{|C_i^*|}$. Since our instance is β-distributed it holds that p belongs to C_i^*, and from the definition of the inner ring of C_i^*, it holds that p falls *outside* the inner ring. However, z is added to T because the ball $B(z, r)$ contains more than $s/2 \ge |C_i^*|/2$ many points. So more than half of the points in C_i^* fall outside the inner ring of C_i^*, which contradicts Fact 4.1.

Assume now (b) does not hold. Recall that T is a connected component, so exists some path $p_1 \to p_2$. Each two consecutive points along this path were connected because their distance is at most $\frac{\beta\mathsf{OPT}}{4s} \le \frac{\beta\mathsf{OPT}}{4|C_i^*|}$. As $d(c_i^*, p_1) < \frac{\beta}{2}\frac{\mathsf{OPT}}{|C_i^*|}$ and $d(c_i^*, p_2) > \frac{3\beta}{4}\frac{\mathsf{OPT}}{|C_i^*|}$, there must exist a point z along the path whose distance from c_i^* falls in the range $\left[\frac{\beta}{2}\frac{\mathsf{OPT}}{|C_i^*|}, \frac{3\beta}{4}\frac{\mathsf{OPT}}{|C_i^*|}\right]$, contradicting (a). ∎

Claim 4.3. *Let C_i^* be any cheap cluster in the target clustering. By stage $s = |C_i^*|$, the algorithm adds to $\mathcal{Q}$ a component T that contains a point from the inner ring of*

C_i^*.

Proof: Suppose that up to the stage $s = |C_i^*|$ the algorithm has not inserted such a component into $\mathcal{Q}$. Now, it is possible that by stage s, the algorithm has inserted some component T' to $\mathcal{Q}$, s.t. some x in the inner ring of C_i^* is too close to some $y \in T'$ (namely, $d(x, y) \leq 2r$), thus causing x to be removed from the instance. Assume for now this is not the case. This means that the inner ring of cluster C_i^* still contains more than $|C_i^*|/2$ points. Also observe that all inner ring points are of distance at most $\frac{\beta \mathsf{OPT}}{8|C_i^*|}$ from the center, so every pair of inner ring points has a distance of at most $\frac{\beta \mathsf{OPT}}{4|C_i^*|}$. Hence, when we reach stage $s = |C_i^*|$, any ball of radius $r = \frac{\beta \mathsf{OPT}}{4s} = \frac{\beta \mathsf{OPT}}{4|C_i^*|}$ centered at any inner-ring point, must contain all other inner-ring points. This means that at stage $s = |C_i^*|$ all inner ring points are connected among themselves, so they form a component (in fact, a clique) of size $> s/2$. Therefore, the algorithm inserts a new component, containing all inner ring points.

So, by stage $s = |C_i^*|$, one of two things can happen. Either the algorithm inserts a component that contains some inner ring point to Q, or the algorithm removes an inner ring point due to some component $T' \in \mathcal{Q}$. If the former happens, we are done. So let us prove by contradiction that we cannot have only the latter.

Let $s \geq |C_i^*|$ be the stage in which we throw away the first inner ring point of the cluster C_i^*. At stage s the algorithm removes this inner ring point x because there exists a point y in some component $T' \in Q$, s.t. $d(x, y) \leq 2r = \frac{\beta \mathsf{OPT}}{2s}$, and so $d(c_i^*, y) \leq d(c_i^*, x) + d(x, y) \leq \frac{\beta \mathsf{OPT}}{8|C_i^*|} + \frac{\beta \mathsf{OPT}}{2s} \leq \frac{5}{8}\frac{\beta \mathsf{OPT}}{|C_i^*|}$. This immediately implies that T' cannot be the center of an expensive cluster since any such point will be at a distance at least $\frac{\beta \mathsf{OPT}}{|C^*|}$ from c_i^*. Let $s' \geq s \geq |C_i^*|$ be the previous stage in which we added the component T' to Q. As Lemma 4.2 applies to T', we deduce that $d(c_i^*, y) < \frac{\beta}{2}\frac{\mathsf{OPT}}{|C_i^*|}$. Recall that T' contains $> s'/2 \geq |C_i^*|/2$ many points, yet, by assumption, contains none of the $|C_i^*|/2$ points that reside in the inner ring of C_i^*. It follows from Fact 4.1 that some point $w \in T'$ must belong to a different cluster C_j^*. Since the instance is β-distributed, we have that $d(c_i^*, w) > \frac{\beta \mathsf{OPT}}{|C_i^*|}$. The existence of both y and w in T' contradicts part (b) of Lemma 4.2. ■

We call a component $T \in \mathcal{Q}$ *good* if it contains an inner ring point of some cheap cluster C_i^*. A component is called *bad* if it is not good and is not one of the initial centers present in $\mathcal{Q}_{init}$. We now discuss the properties of good components.

Claim 4.4. *Let T be a good component added to $\mathcal{Q}$, containing an inner ring point from a cheap cluster C_i^*. (By Claim 4.3 we know at least one such T exists.) Then: (a) all points in T are of distance at most $\frac{\beta \mathsf{OPT}}{2|C_i^*|}$ from c_i^*, (b) $T \cup B(T)$ is fully contained in C_i^*, and (c) the entire inner ring of C_i^* is contained in $T \cup B(T)$, and (d) no other component $T' \in \mathcal{Q}$, $T' \neq T$, contains an inner ring point from C_i^*.*

Proof: As we do not know (d) in advance, it might be the case that $\mathcal{Q}$ contains many good components, all containing an inner-ring point from the same cluster, C_i^*. Out of these (potentially many) components, let T denote the first one inserted to $\mathcal{Q}$. Denote the stage in which T was inserted to $\mathcal{Q}$ as s. Due to the previous claim, we know $s \geq |C_i^*|$, and so Lemma 4.2 applies to T. We show (a), (b), (c) and (d) hold for T, and deduce that T is the only good component to contain an inner ring point from C_i^*.

Part (a) follows immediately from Lemma 4.2. We know T contains some inner ring point x from C_i^*, so $d(c_i^*, x) \leq \frac{\beta}{8}\frac{\mathsf{OPT}}{|C_i^*|} < \frac{\beta}{2}\frac{\mathsf{OPT}}{|C_i^*|}$, so we know that any $y \in T$ must satisfy that $d(c_i^*, y) < \frac{\beta}{2}\frac{\mathsf{OPT}}{|C_i^*|}$. Since we now know (a) holds and the instance is β-distributed, we have that $T \subset C_i^*$, so we only need to show $B(T) \subset C_i^*$. Fix any $y \in B(T)$. The point y is assigned to $B(T)$ (thus removed from the instance) because there exists some point $x \in T$ s.t. $d(x, y) \leq 2r$. So again, we have that $d(c_i^*, y) \leq d(c_i^*, x) + d(x, y) \leq \frac{\beta \mathsf{OPT}}{|C_i^*|}$, which gives us that $y \in C_i^*$ (since the instance is β-distributed).

We now prove (c). Because of (b), we deduce that the number of points in T is at most $|C_i^*|$. However, in order for T to be added to $\mathcal{Q}$, it must also hold that $|T| > s/2$. It follows that $s < 2|C_i^*|$. Let x be an inner ring point of C_i^* that belongs to T. Then the distance of any other inner ring point of C_i^* and x is at most $\frac{\beta \mathsf{OPT}}{4|C_i^*|} < \frac{\beta \mathsf{OPT}}{2s} = 2r$. It follows that any inner ring point of C_i^* which isn't added to T is assigned to $B(T)$. Thus $T \cup B(T)$ contains all inner-ring points. Finally, observe that (d) follows immediately from the definition of a good component and from (c). ■

We now show that in addition to having all k good components, we cannot have too many bad components.

Claim 4.5. *We have less than $16/(3\beta)$ bad components.*

Proof: Let T be a bad component, and let s be the stage in which T was inserted to $\mathcal{Q}$. Let y be any point in T, and let C^* be the cluster to which y belongs in the optimal clustering with center c^*. We show $d(c^*, y) > \frac{3\beta}{8}\frac{\mathsf{OPT}}{s}$. We divide into cases.

Case 1: C^* is an expensive cluster. Note that we are working under the assumption that $\mathcal{Q}_{init}$ contains the correct centers of the expensive clusters. In particular, $\mathcal{Q}_{init}$ contains c^*. Also, the fact that point y was not thrown out in stage s implies that $d(c^*, y) > 2r = \frac{\beta \mathsf{OPT}}{2s} > \frac{3\beta \mathsf{OPT}}{8s}$.

Case 2: C^* is a cheap cluster and $s \geq |C^*|$. We apply Lemma 4.2, and deduce that either $d(c^*, y) < \frac{\beta}{2}\frac{\mathsf{OPT}}{|C^*|}$ or that $d(c^*, y) > \frac{3\beta}{4}\frac{\mathsf{OPT}}{|C^*|} \geq \frac{3\beta}{4}\frac{\mathsf{OPT}}{s}$. As the inner ring of C^* contains $> |C^*|/2$ and T contains $> s/2 \geq |C^*|/2$ many points, none of which is an inner ring point, some point $w \in T$ does not belong to C^* and hence $d(c^*, w) > \frac{\beta \mathsf{OPT}}{|C^*|} > \frac{3\beta}{4}\frac{\mathsf{OPT}}{|C^*|}$. Part (b) of Lemma 4.2 assures us that all points in T are also far from c^*.

Case 3: C^* is a cheap cluster and $s < |C^*|$. Using Claim 4.3 we have that some good component containing a point x from the inner ring of C^* was already added to $\mathcal{Q}$. So it must hold that $d(x, y) > 2r$, for otherwise we removed

y from the instance and it cannot be added to any T. We deduce that $d(c^*, y) \geq d(x, y) - d(c^*, x) \geq \frac{\beta\mathsf{OPT}}{2s} - \frac{\beta\mathsf{OPT}}{8|C^*|} > \frac{3\beta}{8}\frac{\mathsf{OPT}}{s}$.

All points in T have distance $> \frac{3\beta\mathsf{OPT}}{8s}$ from their respective centers in the optimal clustering, and recall that T is added to $\mathcal{Q}$ because T contains at least $s/2$ many points. Therefore, the contribution of all elements in T to OPT is at least $\frac{3\beta\mathsf{OPT}}{16}$. It follows that we can have no more than $16/3\beta$ such bad components. ∎

We can now prove the correctness of our algorithm.

Theorem 4.6. *The algorithm outputs a k-clustering whose cost is no more than $(1+\epsilon)\mathsf{OPT}$.*

Proof: Using Claim 4.4, it follows that there exists some choice of k components, $T_1, \ldots, T_k$, such that we have the center of every expensive cluster and the good component corresponding to every cheap cluster $\mathcal{C}^*$. Fix that choice. We show that for the optimal clustering, replacing the true centers $\{c_1^*, c_2^*, ..., c_k^*\}$ with the centers $\{c_1, c_2, ..., c_k\}$ that the algorithm outputs, increases the cost by at most a $(1+\epsilon)$ factor. This implies that using the $\{c_1, c_2, ..., c_k\}$ as centers must result in a clustering with cost at most $(1+\epsilon)\mathsf{OPT}$.

Fix any C_i^* in the optimal clustering. Let OPT_i be the cost of this cluster. If C_i^* is an expensive cluster then we know that its center c_i^* is present in the list of centers chosen. Hence, the cost paid by points in C_i^* will be at most OPT_i. If C_i^* is a cheap cluster then denote by T the good component corresponding to it. We break the cost of C_i^* into two parts: $\mathsf{OPT}_i = \sum_{x \in C_i^*} d(x, c_i^*) = \sum_{x \in T \cup B(T)} d(x, c_i^*) + \sum_{x \in C_i^*, \text{ yet } x \notin T \cup B(T)} d(x, c_i^*)$ and compare it to the cost C_i^* using c_i, the point picked by the algorithm to serve as center: $\sum_{x \in C_i^*} d(x, c_i) = \sum_{x \in T \cup B(T)} d(x, c_i) + \sum_{x \in C_i^*, \text{ yet } x \notin T \cup B(T)} d(x, c_i)$. Now, the first term is exactly the function that is minimized by c_i, as $c_i = \arg\min_p \sum_{x \in T \cup B(T)} d(x, p)$. We also know c_i^*, the actual center of C_i^*, resides in the inner ring, and therefore, by Claim 4.4 must belong to $T \cup B(T)$. It follows that $\sum_{x \in T \cup B(T)} d(x, c_i) \leq \sum_{x \in T \cup B(T)} d(x, c_i^*)$. We now upper bound the 2nd term, and show that $\sum_{x \in C_i^*, \text{ yet } x \notin T \cup B(T)} d(x, c_i) \leq (1+\epsilon) \sum_{x \in C_i^*, \text{ yet } x \notin T \cup B(T)} d(x, c_i^*)$

Any point $x \in C_i^*$, s.t. $x \notin T \cup B(T)$, must reside outside the inner ring of C_i^*. Therefore, $d(x, c_i^*) > \frac{\beta\mathsf{OPT}}{8|C_i^*|}$. We show that $d(c_i, c_i^*) \leq \epsilon\frac{\beta\mathsf{OPT}}{8|C_i^*|}$, and thus we have that $d(x, c_i) \leq d(x, c_i^*) + d(c_i^*, c_i) \leq (1+\epsilon)d(x, c_i^*)$, which gives the required result.

Note that thus far, we have only used the fact that the cost of any cheap cluster is proportional to $\beta\mathsf{OPT}/|C_i^*|$. Here is the first (and the only) time we use the fact that the cost is actually at most $(\epsilon/32) \cdot \beta\mathsf{OPT}/|C_i^*|$. Using the Markov inequality, we have that the set of points satisfying $\{x;\ \ d(x, c_i^*) \leq \epsilon \cdot \beta\mathsf{OPT}/(16|C_i^*|)\}$ contains at least half of the points in C_i^*, and they all reside in the inner ring, thus belong to $T \cup B(T)$. Assume for the sake of contradiction that $d(c_i, c_i^*) \geq \epsilon\frac{\beta\mathsf{OPT}}{8|C_i^*|}$. Then at least half of the points in C_i^* contribute more than $\epsilon\frac{\beta\mathsf{OPT}}{16|C_i^*|}$ to the sum $\sum_{x \in T \cup B(T)} d(x, c_i)$. It follows that this sum is more than $\epsilon\frac{\beta\mathsf{OPT}}{32|C_i^*|} \geq \mathsf{OPT}_i$. However, c_i is the point that minimizes the sum $\sum_{x \in T \cup B(T)} d(x, p)$, and by using $p = c_i^*$ we have $\sum_{x \in T \cup B(T)} d(x, p) \leq \mathsf{OPT}_i$. Contradiction. ∎

B. Runtime analysis

A naive implementation of the 2nd step of algorithm in Section 4-A takes $O(n^3)$ time (for every s and every point x, find how many of the remaining points fall within the ball of radius r around it). Finding c_i for all components takes $O(n^2)$ time, and measuring the cost of the solution using a particular set of k data points as centers takes $O(nk)$ time. Guessing the right k components takes $k^{O(1/\beta)}$ time. Overall, the running time of the algorithm in Figure 1 is $O(n^3 k^{O(1/\beta)})$. The general algorithm that brute-force guesses the centers of all expensive clusters, makes $n^{O(1/\beta\epsilon)}$ iterations of the given algorithm, so its overall running time is $n^{O(1/\beta\epsilon)} k^{O(1/\beta)}$.

5. A PTAS FOR ANY β-DISTRIBUTED EUCLIDEAN k-MEANS INSTANCE

Analogous to the k-median algorithm, we present an essentially identical algorithm for k-means in Euclidean space. Indeed, the fact that k-means considers distances squared, makes upper (or lower) bounding distances a bit more complicated, and requires that we fiddle with the parameters of the algorithm. In addition, the centers c_i^* may not be data points. However, the overall approach remains the same. Roughly speaking, converting the k-median algorithm to the k-means case, we use the same constants, only squared.[5] As before we handle expensive clusters by guessing good substitutes for their centers and obtain *good* components for cheap clusters.

Often, when considering the Euclidean space k-means problem, the dimension of the space plays an important factor. In contrast, here we make no assumptions about the dimension, and our results hold for any $\mathrm{poly}(n)$ dimension. In fact, for ease of exposition, we assume all distances between any two points were computed in advance and are given to our algorithm. Clearly, this only adds $O(n^2 \cdot \mathsf{dim})$ to our runtime. In addition to the change in parameters, we utilize the following facts that hold for the center of mass in Euclidean space.

Fact 5.1. *Let U be a (finite) set of points in an Euclidean space, and let μ_U denote their center of mass ($\mu = \frac{1}{|U|}\sum_{x \in U} x$). Let A be a random subset of U, and denote by μ_A the center of mass of A. Then for any $\delta < 1/2$, we have both*

$$\Pr\left[\|\mu_U - \mu_A\|^2 > \frac{1}{\delta|A|} \cdot \frac{1}{|U|}\sum_{x \in U}\|x - \mu_U\|^2\right] < \delta \quad (1)$$

[5] We stress that we made no attempt to optimize the constants.

$$\Pr\left[\sum_{x\in U}\|x-\mu_A\|^2 > (1+\frac{1}{\delta|A|})\cdot\sum_{x\in U}\|x-\mu_U\|^2\right] < \delta \tag{2}$$

Fact 5.2. *Let U be a (finite) set of points in an Euclidean space, and let $A \neq \emptyset$ and B be a partition of U. Denote by μ_U and μ_A the center of mass of U and A resp. Then $\|\mu_U - \mu_A\|^2 \leq \frac{1}{|U|}\sum_{x\in U}\|x-\mu_U\|^2 \cdot \frac{|B|}{|A|}$.*

Fact 5.2, proven in [18] (Lemma 2.2), allows us to upper bound the distance between the real center of a cluster and the empirical center we get by averaging all points in $T \cup B(T)$ for a good component T. Fact 5.1 allows us to handle expensive clusters. Since we cannot brute force guess a center (as the center of the clusters aren't necessarily data points), we guess a sample of $O(\beta^{-1}+\epsilon^{-1})$ points from every expensive cluster, and use their average as a center. Both properties of Fact 5.1, proven in [13] (§3, Lemma 1 and 2), assure us that the center is an adequate substitute for the real center and is also close to it. This motivates the approach behind our first algorithm, in which we brute-force traverse all choices of $O(\epsilon^{-1}+\beta^{-1})$ points for any of the expensive clusters.

The second algorithm, whose runtime is $(k\log n)^{\mathrm{poly}(1/\epsilon,1/\beta)}O(n^3)$, replaces brute-force guessing with random sampling. Indeed, if a cluster contains $\mathrm{poly}(1/k)$ fraction of the points, then by randomly sampling $O(\epsilon^{-1}+\beta^{-1})$ points, the probability that all points belong to the same expensive cluster, and furthermore, their average can serve as a good empirical center, is at least $1/k^{\mathrm{poly}(1/\epsilon,1/\beta)}$. In contrast, if we have expensive clusters that contain few points (e.g. an expensive cluster of size $\sqrt{n}$, while $k = \mathrm{poly}(\log(n))$), then random sampling is unlikely to find good empirical centers for them. However, recall that our algorithm collects points and deletes them from our instance. So, it is possible that in the middle of the run, we are left with so few points, so that expensive clusters whose size is small in comparison to the original number of points, contain a $\mathrm{poly}(1/k)$ fraction of the *remaining* points.

Indeed, this is the motivation behind our second algorithm. We run the algorithm while interleaving the Population Stage of the algorithm with random sampling. Instead of running s from n to 1, we use $\left\{n, \frac{n}{k^2}, \frac{n}{k^4}, \frac{n}{k^6}, \ldots, 1\right\}$ as break points. Correspondingly, we define l_i to be the number of expensive clusters whose size is in the range $\left[n\cdot k^{-2i-2}, n\cdot k^{-2i}\right)$. Whenever s reaches such a $n\cdot k^{-2i}$ break point, we randomly sample points in order to guess the l_{i+3} centers of the clusters that lie 3 intervals "ahead" (and so, initially, we guess all centers in the first 3 intervals). We prove that in every interval we are likely to sample good empirical centers. This is a simple corollary of Fact 5.2 along with the following two claims. First, we claim that at the end of each interval, the number of points remaining is at most $n\cdot k^{-2i+1}$. Secondly, we also claim that in each interval we do not remove even a single point from a cluster whose size is smaller than $n\cdot k^{-2i-6}$. We refer the reader to Appendix A for the algorithms and their analysis.

6. Discussion and Open Problems

The algorithm we present here for k-median has runtime of $\mathrm{poly}(n^{1/\beta}, n^{1/\epsilon}, k)$, and the algorithm for k-means has runtime $\mathrm{poly}(n, (k\log n)^{1/\epsilon}, (k\log n)^{1/\beta})$.[6] We comment that it is unlikely that we can obtain an algorithm of runtime $\mathrm{poly}(n^{1/\epsilon}, 1/\beta, k)$. Observe that for any clustering instance and any $k > 1$ we have that $\frac{\mathsf{OPT}_{(k-1)}}{\mathsf{OPT}} > 1+\frac{1}{n}$, simply by considering the k-clustering that results from taking the optimal $(k-1)$-clustering, and setting the point which is the furthest from its center in a cluster of its own (as a new center). Hence, any k-median/k-means instance is β-distributed for $\beta = \Omega(\frac{1}{n})$. Recall from Section 3-D the k-median problem restricted only to weakly-stable instances has no FPTAS. So the fact that our algorithm's runtime has super-polynomial dependence in *both* $1/\beta$ and $1/\epsilon$ is unavoidable. Nonetheless, one might still hope to do better. In particular, one major runtime expense of our algorithm comes from handling expensive clusters by brute-force guessing or sampling. Can one improve the runtime by doing something more clever for expensive clusters? It is worth noting that for the stability conditions of [4], Voevodski et al. [20] develop an especially efficient implementation with good performance (in terms of both accuracy and speed) on real-world protein sequence datasets.

A different open problem lies in the relation to results of Ostrovsky et al. [18]. Their motivating question was to analyze the performance of Lloyd-type methods over stable instances. Is it possible that weak deletion-stability is sufficient for some version of the k-means heuristic to converge to the optimal clustering?

References

[1] Sanjeev Arora, Prabhakar Raghavan, and Satish Rao. Approximation schemes for Euclidean k-medians and related problems. In *STOC*, 1998.

[2] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristic for k-median and facility location problems. In *STOC*, 2001.

[3] Mihai Bādoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *STOC*, pages 250–257, 2002.

[4] Maria-Florina Balcan, Avrim Blum, and Anupam Gupta. Approximate clustering without the approximation. In *SODA*, 2009.

[5] Maria-Florina Balcan and Mark Braverman. Finding low error clusterings. In *COLT*, 2009.

[6] Maria-Florina Balcan, Heiko Röglin, and Shang-Hua Teng. Agnostic clustering. In *ALT*, pages 384–398, 2009.

[7] Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k-median problem. In *STOC*, 1999.

[6]When dealing with k-means in a Euclidean space of dimension dim, we need to explicitly compute the distances, so we add $n^2\mathsf{dim}$ to the runtime.

[8] Sanjoy Dasgupta. The hardness of k-means clustering. Technical report, University of California at San Diego, 2008.

[9] W. Fernandez de la Vega, Marek Karpinski, Claire Kenyon, and Yuval Rabani. Approximation schemes for clustering problems. In *STOC*, 2003.

[10] Michelle Effros and Leonard J. Schulman. Deterministic clustering with data nets. *ECCC*, (050), 2004.

[11] Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. In *Journal of Algorithms*, pages 649–657, 1998.

[12] Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *STOC*, pages 291–300, 2004.

[13] Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted voronoi diagrams and randomization to variance-based k-clustering: (extended abstract). In *Proc. 10th Symp. Comp. Geom.*, pages 332–339, 1994.

[14] Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems (extended abstract). In *STOC*, pages 731–740, 2002.

[15] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. In *Proc. 18th Symp. Comp. Geom.*, 2002.

[16] Amit Kumar, Yogish Sabharwal, and Sandeep Sen. A simple linear time (1+ ϵ)-approximation algorithm for k-means clustering in any dimensions. In *FOCS*, 2004.

[17] R. Ostrovsky and Y. Rabani. Polynomial time approximation schemes for geometric k-clustering. In *FOCS*, 2000.

[18] Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman, and Chaitanya Swamy. The effectiveness of Lloyd-type methods for the k-means problem. In *FOCS*, pages 165–176, 2006.

[19] F. Schalekamp, M. Yu, and A. van Zuylen. Clustering with or without the Approximation. In *COCOON*, 2010.

[20] Konstantin Voevodski, Maria Florina Balcan, Heiko Roglin, ShangHua Teng, and Yu Xia. Efficient clustering with limited distance information. In *Proc. 26th UAI*, 2010.

APPENDIX

We present the algorithm for $(1+\epsilon)$-approximation to the k-means optimum of a β-distributed instance. Much like in Section 4, we call a cluster in the optimal k-means solution cheap if $\mathsf{OPT}_i = \sum_{x \in C_i^*} d^2(x, c_i^*) \leq \frac{\beta\epsilon\mathsf{OPT}}{4^6}$.

A. Clustering β-distributed Instances of Euclidean k-means

The algorithm is presented in Figure 2. The correctness is proved in a similar fashion to the proof of correctness presented in Section 4. First, observe that by the Markov inequality, for any cheap cluster C_i^*, we have that the set $\left\{x;\ d^2(x, c_i^*) > t\frac{\beta\mathsf{OPT}}{|C_i^*|}\right\}$ cannot contain more than $\epsilon/(4^6 t)$ fraction of the points in $|C_i^*|$. It follows that the inner ring of C_i^*, the set $\left\{x;\ d^2(x, c_i^*) \leq \frac{\beta\mathsf{OPT}}{256|C_i^*|}\right\}$, contains at least half of the points of C_i^*. As mentioned Section 5 the algorithm populates the list $\mathcal{Q}$ with *good* components corresponding to cheap clusters. Also from Section 5, we know that for every expensive cluster, there exists a sample of $O(\frac{1}{\beta} + \frac{1}{\epsilon})$ data points whose center is a good substitute for the center of the cluster. In the analysis below, we assume that $\mathcal{Q}$ has been initialized correctly with $\mathcal{Q}_{init}$ containing these good substitutes. In general, the algorithm will be run multiple times for all possible guesses of samples from expensive clusters. We start with the following lemma which is similar to Lemma 4.2.

1) **Initialization Stage:** Set $\mathcal{Q} \leftarrow \mathcal{Q}_{init}$.
2) **Population Stage:** For $s = n, n-1, n-2, \ldots, 1$ do:
   a) Set $r = \frac{\beta\mathsf{OPT}}{64s}$.
   b) Remove any point x such that $d^2(x, \mathcal{Q}) < 4r$.
   (Here, $d(x, \mathcal{Q}) = \min_{T \in \mathcal{Q}; y \in T} d(x, y)$.)
   c) For any remaining data point x, denote the set of data points whose distance squared from x is at most r, by $B(x, r)$. Connect any two remaining points a and b if:
   (i) $d^2(a, b) \leq r$, (ii) $|B(a, r)| > \frac{s}{2}$ and (iii) $|B(b, r)| > \frac{s}{2}$.
   d) Let T be a connected component of size $> \frac{s}{2}$. Then:
      i) Add T to $\mathcal{Q}$. (That is, $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{T\}$.)
      ii) Define the set $B(T) = \{x : d^2(x, y) \leq 4r$ for some $y \in T\}$. Remove the points of $B(T)$ from the instance.
3) **Centers-Retrieving Stage:** For any choice of k components $T_1, T_2, \ldots, T_k$ out of $\mathcal{Q}$
   a) Find the best center c_i for $T_i \cup B(T_i)$. That is $c_i = \mu(T_i \cup B(T_i)) = \frac{1}{|T_i \cup B(T_i)|} \sum_{x \in T_i \cup B(T_i)} x$.
   b) Partition all n points according to the nearest point among the k centers of the current k components.
   c) If a clustering of cost at most $(1+\epsilon)\mathsf{OPT}$ is found – output these k centers and halt.

Figure 2. A PTAS for β-distributed instances of Euclidean k-means.

Lemma A.1. *Let $T \in \mathcal{Q}$ be any component and let s be the stage in which we insert T to $\mathcal{Q}$. Let C_i^* be any cheap cluster s.t. $s \geq |C_i^*|$. Then (a) T does not contain any point z s.t. the distance $d^2(c_i^*, z)$ lies within the range $\left[\frac{\beta}{16}\frac{\mathsf{OPT}}{|C_i^*|}, \frac{\beta}{4}\frac{\mathsf{OPT}}{|C_i^*|}\right]$, and (b) T cannot contain both a point p_1 s.t. $d^2(c_i^*, p_1) \leq \frac{\beta}{16}\frac{\mathsf{OPT}}{|C_i^*|}$ and a point p_2 s.t. $d^2(c_i^*, p_2) > \frac{\beta}{4}\frac{\mathsf{OPT}}{|C_i^*|}$.*

Proof: Assume (a) does not hold. Let z be such point, and let $B(z, r)$ be the set of all points p s.t. $d^2(z, p) \leq r = \frac{\beta\mathsf{OPT}}{64s} \leq \frac{\beta\mathsf{OPT}}{64|C_i^*|}$. As $d^2(z, c_i^*) \geq \frac{\beta\mathsf{OPT}}{16|C_i^*|}$, we have that $d(z,p) \leq \frac{1}{2}d(z,c_i^*)$. It follows that $d^2(c_i^*,p) \geq (d(c_i^*,z)-d(z,p))^2 \geq (d(c_i^*,z)/2)^2 = \frac{\beta\mathsf{OPT}}{64|C_i^*|}$. Similarly, $d^2(c_i^*,p) \leq (d(c_i^*,z)+d(z,p))^2 \leq (3d(c_i^*,z)/2)^2 \leq \frac{9\beta}{16}\frac{\mathsf{OPT}}{|C_i^*|}$. Thus $B(z,r)$ is contained in C_i^*, but falls outside the inner-ring of C_i^*, yet contains $s/2 \geq |C_i^*|/2$ many points. Contradiction.

Assume (b) does not hold. Let p_1 and p_2 the above mentioned points. As T is a connected components, it follows that along the path $p_1 \to p_2$, exists a pairs of neighboring nodes, x,y, s.t. $d^2(x,y) \leq r \leq \frac{\beta\mathsf{OPT}}{64|C_i^*|}$ yet $d^2(c_i^*,x) \leq \frac{\beta}{16}\frac{\mathsf{OPT}}{|C_i^*|}$ while $d^2(c_i^*,y) \geq \frac{\beta}{4}\frac{\mathsf{OPT}}{|C_i^*|}$. However, a simple computation gives that $d^2(c_i^*,y) \leq (3d(c_i^*,x)/2)^2 \leq \frac{9\beta}{64}\frac{\mathsf{OPT}}{|C_i^*|}$. Contradiction. ∎

Lemma A.1 allows us to give the analogous claims to Claims 4.3 and 4.4. As before, call a component T *good* if it is contained within some target cluster C_i^* *and* $T \cup B(T)$ contains all of the inner ring points of C_i^*. Otherwise, the component is called *bad* provided it is not one of the initial centers present in $\mathcal{Q}_{init}$. We now show that each cheap target cluster will have a single, unique, good component.

Claim A.2. *Let C_i^* be any cheap cluster in the target clustering. By stage $s = |C_i^*|$, the algorithm adds to $\mathcal{Q}$ a component T that contains a point from the inner ring of C_i^*.*

Claim A.3. *Let T be a good connected component added to $\mathcal{Q}$, containing an inner ring point from cluster C_i^*. Then: (a) all points in T are of distance squared at most $\frac{\beta\mathsf{OPT}}{16|C_i^*|}$ from c_i^*, (b) $T \cup B(T)$ is fully contained in C_i^*, and (c) the entire inner ring of C_i^* is contained in $T \cup B(T)$, and (d) no other component $T' \neq T$ in $\mathcal{Q}$ contains an inner ring point from C_i^*.*

As the proofs of Claims A.2 and A.3 are identical to the Claims 4.3 and 4.4, we omit them.

Lemma A.4. *We do not add to $\mathcal{Q}$ more than $1000/\beta$ bad components.*

Proof: Consider any bad component T that we add to $\mathcal{Q}$ and denote that stage in which we insert T to $\mathcal{Q}$ as s. So the size of this component is $> \frac{s}{2}$. Let y be an arbitrary point from T which belongs to cluster C^* in the optimal clustering. Let c^* be the center of C^*. We show that $d^2(c^*,y) > \frac{\beta\mathsf{OPT}}{500s}$.

We divide into cases.

Case 1: C^* is a cheap cluster and $s \geq |C^*|$. Recall that T must contain $s/2 \geq |C^*|/2$ points, so it follows that T contains some point x that does not belong to C^*. β-stability gives that this point has distance $d^2(c^*,x) > \beta\frac{\mathsf{OPT}}{|C^*|}$, and we apply Lemma A.1 to deduce that all points in T are of distance squared of at least $\frac{\beta}{4}\frac{\mathsf{OPT}}{|C^*|}$.

Case 2: C^* is a cheap cluster and $s < |C^*|$. In this case we have that the entire inner ring of C^* already belongs to some $T' \in Q$. Let $x \in T'$ be any inner ring point from C^*, and we have that $d(c^*,x)^2 \leq \frac{\beta\mathsf{OPT}}{256|C^*|} \leq \frac{\beta\mathsf{OPT}}{256s}$, while $d^2(x,y) > \frac{\beta\mathsf{OPT}}{16s}$. It follows that $d^2(c^*,y) \geq (3d(x,y)/4)^2 > \frac{\beta\mathsf{OPT}}{500s}$.

Case 3: C^* is an expensive cluster and $s > 2|C^*|$. We claim that $d^2(c^*,y) > \frac{\beta\mathsf{OPT}}{32|C^*|}$. If, by contradiction, we have that $d^2(c^*,y) \leq \frac{\beta\mathsf{OPT}}{32|C^*|}$, then we show that the ball $B(y,r)$ contains only points from C_i^*, yet it must contains $s/2 > |C_i^*|$ points. This is because each $p \in B(y,r)$ satisfies that $d^2(c^*,p) \leq (d(c^*,y)+d(y,p))^2 \leq \left(\sqrt{\frac{\beta\mathsf{OPT}}{32|C^*|}}+\sqrt{\frac{\beta\mathsf{OPT}}{16s}}\right)^2 < \frac{\beta\mathsf{OPT}}{|C^*|}$.

Case 4: C^* is an expensive cluster and $s \leq 2|C^*|$. In this case, from Fact 5.1 we know that $\mathcal{Q}_{init}$ contains a a good empirical center c for the expensive cluster C^*, in the sense that $\|c-c^*\|^2 \leq \frac{\beta\mathsf{OPT}}{512|C^*|} \leq \frac{\beta\mathsf{OPT}}{256s}$. Then, similarly case 2 above we have $d^2(y,c^*) \geq (d(y,c)-d(c,c^*))^2 > \frac{\beta\mathsf{OPT}}{500s}$. It follows that every point in T has a large distance from its center. Therefore, the $s/2$ points in this component contribute at least $\beta\mathsf{OPT}/1000$ to the k-means cost. Hence, we can have no more than $1000/\beta$ such bad components. ∎

We now prove the main theorem.

Theorem A.5. *The algorithm outputs a k-clustering whose cost is at most $(1+\epsilon)\mathsf{OPT}$.*

Proof: Using Claim A.3, it follows that there exists some choice of k components which has good components for all the cheap clusters and good substitutes for the centers of the expensive clusters. Fix that choice and consider a cluster C_i^* with center c_i^*. If C_i^* is an expensive cluster then from Section 5 we know that $\mathcal{Q}_{init}$ contains a point c_i such that $d^2(c_i,c_i^*) \leq \frac{\beta\epsilon}{\beta+\epsilon}\frac{\mathsf{OPT}_i}{|C_i^*|}$. Hence, the cost paid by the points in C_i^* will be atmost $(1+\epsilon)\mathsf{OPT}_i$. If C_i^* is a cheap cluster then denote by T the good component that resides within C_i^*. Denote $T \cup B(T)$ by A, and $C_i^* \setminus A$ by B. Let c_i be the center of A. We know that the entire inner-ring of C_i^* is contained in A, therefore, B cannot contain more than $\epsilon/16$ fraction of the points of C_i^*. Fact 5.2 dictates that in this case, $\|c_i^*-c_i\|^2 \leq \epsilon^2\frac{\beta\mathsf{OPT}}{4^6|C_i^*|}$. We know every $x \in B$ contributes at least $\frac{\beta\mathsf{OPT}}{256|C_i^*|}$ to the cost of C_i^*, so $\|c_i^*-c_i\|^2 \leq \frac{\epsilon}{16}\|x-c_i^*\|^2$. Thus, for every $x \in B$, we have that $\|x-c_i\|^2 \leq (1+\epsilon)\|x-c_i^*\|^2$. It follows that $\sum_{x\in B}\|x-c_i\|^2 \leq (1+\epsilon)\sum_{x\in B}\|x-c_i^*\|^2$, and obviously $\sum_{x\in A}\|x-c_i\|^2 \leq \sum_{x\in A}\|x-c_i^*\|^2$ as c_i is the center of mass of A. Therefore, when choosing the good k components out of $\mathcal{Q}$, we can assign them to the centers in such a way that costs no more than $(1+\epsilon)\mathsf{OPT}$. Obviously the assignment of each point to the nearest of the k-centers only yields a less costly clustering, and thus its cost is also at most $(1+\epsilon)\mathsf{OPT}$. ∎

B. A Randomized Algorithm for β-distributed k-Means Instances

We now present a randomized algorithm which achieves a $(1+\epsilon)$ approximation to the k-means optimum of a β-distributed instance and runs in time $(k\log_k n)^{\mathrm{poly}(1/\epsilon,1/\beta)}O(n^3)$. The algorithm is similar in nature to the one presented in the previous section, except

1) Guess $l \le \frac{4^6}{\beta\epsilon}$, the number of expensive clusters. Set $t = \frac{1}{2}(\log_k n)$. Guess non-negative integers $g_1, g_2, \ldots g_t$, such that $\sum_i g_i = l$.
2) Sample $g_1 + g_2 + g_3$ sets, by sampling independently and u.a.r $O(\frac{1}{\beta} + \frac{1}{\epsilon})$ points for each set. For each such set $\tilde{T}_j$, add the singleton $\{\mu(\tilde{T}_j)\}$ to $\mathcal{Q}$.
3) Modify the Population Stage from the previous algorithm, so that whenever $s = \frac{n}{k^{2i}}$ for some $i \ge 1$ (We call this the *interval* i)
   - Sample g_{i+3} sets, by sampling independently and u.a.r $O(\frac{1}{\beta} + \frac{1}{\epsilon})$ points for each set. For each such set $\tilde{T}_j$, add the singleton $\{\mu(\tilde{T}_j)\}$ to $\mathcal{Q}$.

that for expensive clusters we replace brute force guessing of samples with random sampling. Note that the straightforward approach of sampling the points right at the start of the algorithm might fail, if there exist expensive clusters which contain very few points. A better approach is to interleave the sampling step with the rest of the algorithm. In this way we sample points from an expensive cluster only when it contains a reasonable fraction of the total points remaining, hence our probability of success is noticeable (namely, $\mathrm{poly}(1/k)$).

The high-level approach of the algorithm is to partition the main loop of the Population Stage, in which we try all possible values of s (starting from n and ending at 1), into *intervals*. In interval i we run s on all values starting with $\frac{n}{k^{2i}}$ and ending with $\frac{n}{k^{2i+2}}$. So overall, we have no more than $t = \frac{1}{2}\log_k(n)$ intervals. Our algorithm begins by guessing l, the number of expensive clusters, then guessing $g_1, g_2, \ldots, g_t$ s.t. $\sum_i g_i = l$. Each g_i is a guess for the number of expensive clusters whose size lies in the range $\left[\frac{n}{k^{2i}}, \frac{n}{k^{2(i-1)}}\right)$. Note that $\sum_i g_i = \#$ expensive clusters $\le \frac{4^6}{\beta\epsilon}$. Hence, there are at most $(\log_k n)^{\frac{4^6}{\beta\epsilon}}$ number of possible assignments to g_i's and we run the algorithm for every such possible guess.

Fixing $g_1, g_2, \ldots, g_t$, we run the Population Stage of the previous algorithm. However, whenever s reaches a new interval, we apply random sampling to obtain good empirical centers for the expensive clusters whose size lies *three intervals "ahead"*. That is, in the beginning of interval i, the algorithm tries to collect centers for the clusters whose size $\ge \frac{n}{k^{6+2i}} = \frac{s}{k^6}$, yet $\le \frac{n}{k^{4+2i}} = \frac{s}{k^4}$. We assume for this algorithm that k is significantly greater than $\frac{1}{\beta}$. Obviously, if k is a constant, then we can use the existing algorithm of Kumar et al. [16].

In order to prove the correctness of the new algorithm, we need to show that the sampling step in the initialization stage succeeds with noticeable probability. Let l_i be the actual number of expensive clusters whose size belongs to the range $\left[\frac{n}{k^{2i}}, \frac{n}{k^{2(i-1)}}\right)$. In the proof which follows, we assume that the correct guess for l_i's has been made, i.e. $g_i = l_i$, for every i. We say that the algorithm *succeeds at the end of interval* i if the following conditions hold:

1) In the beginning of the interval, our guess for all clusters that belong to interval $(i+3)$ produces good empirical centers. That is, for every expensive cluster C^* of size in the range $\left[\frac{n}{k^{6+2i}}, \frac{n}{k^{4+2i}}\right)$, the algorithm picks a sample $\tilde{T}$ such that the mean $\mu(\tilde{T})$ satisfies:
   (a) $d^2(\mu(\tilde{T}), c^*) \le \frac{\beta \mathsf{OPT}}{256|C^*|}$.
   (b) $\sum_{x \in C^*} d^2(x, \mu(\tilde{T})) \le (1+\epsilon)\sum_{x \in C^*} d^2(x, c^*)$.
2) During the interval, we do not delete any point p that belongs to some target cluster C^* of size $\le \frac{n}{k^{4+2(i+1)}}$ points.
3) At the end of the interval, the total number of remaining points (points that were not added to some $T \in \mathcal{Q}$ or deleted from the instance because they are too close to some $T' \in \mathcal{Q}$) is at most $\frac{n}{k^{2i-1}}$.

Lemma A.6. *For every $i \ge 1$, let S_i denote the event that the algorithm succeeds at the end of interval i. Then* $\Pr[S_i | S_1, S_2, \ldots, S_{i-1}] \ge k^{-l_{(i+3)} \cdot O(\frac{1}{\beta} + \frac{1}{\epsilon})}$

Before going into the proof we show that Lemma A.6 implies that with noticeable probability, our algorithm returns a $(1+\epsilon)$-approximation of the k-means optimal clustering. First, observe the technical fact that for the first three intervals l_1, l_2, l_3, we need to guess the centers of clusters of size $\ge \frac{n}{k^6}$ before we start our Population Stage. However, as these clusters contain k^{-6} fraction of the points, then using Fact 5.1, our sampling finds good empirical centers for all of these $l_1 + l_2 + l_3$ expensive clusters w.p. $\ge k^{-(l_1+l_2+l_3)O(\frac{1}{\beta}+\frac{1}{\epsilon})}$. Applying Lemma A.6 we get that the probability our algorithm succeeds after all intervals is $\ge 1/k^{O(\frac{\beta+\epsilon}{\beta^2\epsilon^2})}$. Now, a similar analysis as in the previous section gives us that for the correct guess of the good components in $\mathcal{Q}$, we find a clustering of cost at most $(1+\epsilon)\mathsf{OPT}$.

Proof of Lemma A.6: Recall that β is a constant, whereas k is not. Specifically, we assume throughout the proof that $k^2 > \frac{500}{\beta}$, and so we allow ourselves to use asymptotic notation.

We first prove that condition 2 holds during interval i. Assume for the sake of contradiction that for some cluster C^* whose size is less than $\frac{n}{k^{6+2i}}$, there exists some point $y \in C^*$, which was added to some component T during interval i, at some stage $s \in \left[\frac{n}{k^{2i+2}}, \frac{n}{k^{2i}}\right)$. This means that by setting the radius $r = \frac{\beta \mathsf{OPT}}{64 s}$, the ball $B(y, r)$ contains $> s/2 \ge \frac{n}{2k^{2i+2}}$ points. Since C^* contains at most $\frac{n}{k^{6+2i}}$ many point, we have $|C^*| \ll s/2$, so at least $s/4$ points in $B(y,r)$ belong to other clusters. Our goal is to show that these $s/4$ points contribute more than OPT to the target clustering, thereby achieving a contradiction.

Let x be such point, and denote the cluster that x is assigned to in the target clustering by $C_j^* \ne C^*$. Since the instance is β-distributed we have that $d^2(c^*, x) > \frac{\beta \mathsf{OPT}}{|C^*|} \ge \beta \mathsf{OPT} \frac{k^{6+2i}}{n}$. On the other hand, $d^2(x,y) \le r =$

$\beta\mathsf{OPT}\frac{1}{64s} \leq \beta\mathsf{OPT}\frac{k^{2+2i}}{64n}$. Therefore, $d^2(c^*, x) = \Omega(k^4) \cdot r$, so $d^2(y, c^*) = (d(c^*, x) - d(x, y))^2 = \Omega(k^4) \cdot r$. Recall that in the target clustering each point is assigned to its nearest center, so $d^2(c^*_j, y) \geq d^2(c^*, y) = \Omega(k^4) \cdot r$. So we have that $d^2(c^*_j, x) \geq \left(d(c^*_j, y) - d(x, y)\right)^2 = \Omega(k^4) \cdot r = \Omega(k^4) \cdot \frac{\beta\mathsf{OPT}}{64}\frac{k^{2i}}{n}$.

So, at least $s/4 = \Omega(\frac{n}{k^{2i+2}})$ points contribute $\Omega(k^4)\frac{\beta\mathsf{OPT}}{64}\frac{k^{2i}}{n}$ to the cost of the optimal clustering. Their total contribution is therefore $\Omega(k^2) \cdot \frac{\beta}{64}\mathsf{OPT} > \mathsf{OPT}$. Contradiction.

A similar proof gives that no point $y \in C^*$ is deleted from the instance because for some $x \in T$, where T is some component in $\mathcal{Q}$, we have that $d^2(y, x) < 4r$. Again, assume for the sake of contradiction that such y,x and T exist. Denote by $s \in \left[\frac{n}{k^{2i+2}}, \frac{n}{k^{2i}}\right)$ the stage in which we remove y, and denote by $s' \geq s$ the stage in which we insert T into $\mathcal{Q}$. By setting the radius $r' = \frac{\beta\mathsf{OPT}}{64s'} \leq r$, we have that the ball $B(x, r')$ contains at least $s'/2 \geq s/2$ points, and therefore, the ball $B(y, 5r)$ contains at least $s/2$ points. We now continue as in the previous case.

We now prove condition 1. We assume the algorithm succeeded in all previous intervals. Therefore, at the beginning of interval i, all points that belong to clusters of size $\leq \frac{n}{k^{2i+4}}$ remain in the instance, and in particular, the clusters we wish to sample from at interval i remain intact. Furthermore, by the assumption that the algorithm succeeded up to interval $(i-1)$, we have that each expensive cluster that should be sampled at the beginning of interval i, contains a $1/k^7$ fraction of the remaining points. We deduce that the probability that we pick a random sample of $O(\frac{1}{\beta} + \frac{1}{\epsilon})$ points from such expensive cluster is at least $k^{-O(\frac{1}{\beta}+\frac{1}{\epsilon})}$. Using Fact 5.1 we have that with probability $\geq k^{-O(\frac{1}{\beta}+\frac{1}{\epsilon})}$ this sample yields a good empirical center.

We now prove condition 3, under the assumption that 1 is satisfied. We need to bound the number of points left in the instance at the end of interval i. There are two types of remaining points: points that in the target clustering belong to clusters of size $> \frac{n}{2k^{2i}}$, and points that belong to clusters of size $\leq \frac{n}{2k^{2i}}$. To bound the number of points of the second type is simple – we have k clusters, so the overall number of points of the second type is at most $\frac{n}{2k^{2i-1}}$. We now bound the number of remaining points of the first type.

At the end of the interval $s = \frac{n}{k^{2i+2}}$, so we remove from the instance any point p whose distance (squared) from some point in $\mathcal{Q}$ is at most $4r = \frac{\mathsf{OPT}}{16}\frac{k^{2i+2}}{n}$. We already know that by the end of interval i, either by successfully sampling an empirical center or by adding an inner-ring point to a component in $\mathcal{Q}$, for every cluster C^* of size $> \frac{n}{2k^{2i}}$, exists some $T \in \mathcal{Q}$ with a point $c' \in T$, s.t. $d^2(c^*, c') \leq \frac{\beta\mathsf{OPT}}{256|C^*|} \leq \frac{\beta\mathsf{OPT}}{128}\frac{k^{2i}}{n}$. Thus, if $x \in C^*$ is a point that wasn't removed from the instance by the end of interval i, it must hold that $d^2(c^*, x) \geq (d(c', x) - d(c^*, c'))^2 = \Omega(k^{2i+2})\frac{\mathsf{OPT}}{n}$. Clearly, at most $n \cdot O(k^{-2i-2})$ points can contribute that much to the cost of the optimal k-means clustering, and so the number of points of the first type is at most $\frac{n}{2k^{2i-1}}$ as well. ■

As we need to traverse all guesses g_is, the runtime of this algorithm takes $O(n^3(\log_k n)^{O(\frac{1}{\beta\epsilon})})$. Repeating this algorithm $k^{O(l(\frac{1}{\beta}+\frac{1}{\epsilon}))}$ many times, we increase the probability of success to be $\geq 1/2$, and incur runtime of $O(n^3(\log_k n)^{O(\frac{1}{\beta\epsilon})}k^{O(\frac{\beta+\epsilon}{\beta^2\epsilon^2})})$.