Carnegie Mellon University
Research Showcase @ CMU

Computer Science Department | School of Computer Science

2-2010

Clustering Under Natural Stability Assumptions

Pranjal Awasthi
Carnegie Mellon University

Avrim Blum
Carnegie Mellon University

Or Sheffet
Carnegie Mellon University

Follow this and additional works at: http://repository.cmu.edu/compsci

This Working Paper is brought to you for free and open access by the School of Computer Science at Research Showcase @ CMU. It has been accepted for inclusion in Computer Science Department by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

Clustering Under Natural Stability Assumptions

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

Optimal clustering is a notoriously hard task. Recently, several papers have suggested a new approach to clustering, motivated by examining natural assumptions that arise in practice, or that are made *implicitly* by many standard algorithmic approaches. These assumptions concern various measures of stability of our given clustering instance. The work of Bilu and Linial [BL10] refers to stability with respect to perturbations in the metric space, and gives positive results for inputs stable to perturbations of size $O(n^{1/3})$ for max-cut based clustering. The work of Balcan et al [BBG09] refers to stability with respect to approximations of the objective, and gives positive results for inputs such that all $(1+\alpha)$-approximations to the k-median (or k-means) optimal solution are close, as partitions of the data, to the actual desired clustering. Related assumptions are considered by Ostrovsky et al. [ORSS06].

In this paper we extend these directions in several important ways. For the Bilu-Linial assumption, we show that for center-based clustering objectives (such as k-median, k-means, and k-center) we can efficiently find the optimal clustering assuming only stability to *constant*-magnitude perturbations of the underlying metric. For the approximation assumption of Balcan et al., we relax their condition to require good behavior only for "natural" $(1+\alpha)$-approximations of the objective: specifically, to make assumptions only about approximations in which each point is assigned to its nearest center. This relaxed assumption now allows for clusters in the optimum solution to be quite close together. Nonetheless, we show that for the k-median objective, for any $\alpha > 0$, under the assumption that all such approximations yield the same partitioning it is still possible to obtain an optimal solution in polynomial time. In addition, we show extensions of this result to infinite metric spaces for any $\alpha \geq 1$.

1 Introduction

Problems of clustering data arise in a wide range of different areas – clustering proteins by function, clustering documents by topic, and clustering images by who or what is in them, just to name a few. Unfortunately, clustering can be a challenging problem, with many natural objective functions such as k-median, k-means, and k-center optimization criteria being NP-hard to optimize [GK98, JMS02]. As a result, there has been substantial work on approximation algorithms [ARR98, AGK$^+$01, BCR01, CGTS99, KSS04, dlVKKR03] with both upper and lower bounds on approximability of these and other objective functions.

In this paper we focus on center based clustering objectives, such as k-median and k-means. Under these objectives we not only partition the data into k subsets, but we also assign k special points, called the *centers*, one in each cluster. The quality of a solution is then measured as a function of the distances between the data points and their centers. In general, algorithms for these objectives do not explicitly assume any special structure of the input. However, in order for the results produced to be *meaningful* for the task at hand, one *implicitly* makes the assumption that the optimum clustering according to the given objective is a desirable partitioning of the data (correctly clusters proteins by their actual function or images by what is actually in them). Often, when clustering arises in practice, other subtle yet non-trivial assumptions are made. In this paper, we investigate implications of these assumptions.

Recently, Bilu and Linial [BL10] considered the assumption that not only is the optimal solution to a given objective Φ the desired clustering one is looking for, but that this is also maintained even under bounded multiplicative perturbations to the distance matrix (i.e., the optimum is stable to such perturbations).

This is motivated by the fact that in practice, distances between data points are typically just the result of some heuristic measure (e.g., edit-distance between strings or Euclidean distance in some feature space) rather than true "semantic distance" between objects. Thus, unless the optimal solution on the given distances is correct by pure luck, it likely is correct or nearly so on small perturbations of the given distances as well. This can also be viewed as a conceptual analog of a *large margin* assumption with respect to clustering objective Φ. Bilu and Linial [BL10] analyze this type of assumption in the context of max-cut, but require stability up to quite large perturbations, of multiplicative factors of roughly $O(n^{1/3})$. Here we show that for k-median and k-means objectives (in fact, any center-based objective function), we can use this stability assumption to find the desired clustering for much smaller perturbation levels, on the order of small constants.

In a related vein, in the same way that exact optimization algorithms implicitly assume that the optimum clustering according to the given objective is a desirable partitioning of the data (e.g., correctly clusters proteins by their actual function), approximation algorithms make the implicit assumption that *approximations* to the objective will be desirable as well (e.g., clustering most proteins by their actual function). Balcan et al. [BBG09] show that in fact these implicit assumptions can be used to bypass approximation hardness results for these problems and allow one to efficiently achieve high-accuracy solutions. In particular, for k-median and k-means objectives, they show that for any constant $\alpha > 0$, if data satisfies the property that all $(1+\alpha)$-approximations to the objective are ϵ-close to the desired clustering in terms of how points are partitioned, then one can efficiently get $O(\epsilon)$-close to the desired clustering. This is true *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). That is, one can perform nearly as well in terms of distance to the desired solution as if one could approximate the objective to the NP-hard value. Balcan and Braverman [BB09] extend these results to the min-sum objective as well.

One drawback of the results in [BBG09], however, is that they rely on the assumption that *all* approximations to the objective are close to the desired clustering, not only those that a natural algorithm might output. In particular, natural algorithms for these objectives will output only "Voronoi-based" clusterings: clusterings defined by k cluster centers (chosen in different ways depending on the algorithm) in which each data point is assigned to its nearest center; i.e., clusters must correspond to Voronoi cells.[1] What if we assume only that *Voronoi-based* approximations of good objective value are desirable clusterings? In this case, many of the useful properties implied by the Balcan et al. assumption and used in [BBG09] no longer hold, such as the property that only an ϵ fraction of points can be "roughly indifferent" between their nearest and second-nearest cluster centers (see Figure 3). Can we still use these weakened assumptions to cluster well? In this work we show that indeed we can, though via different methods. We consider two assumptions of this form, depending on whether Steiner points are allowed as cluster centers. When Steiner points are not allowed, we show that for any constant $\alpha > 0$, if all Voronoi-based $(1+\alpha)$-approximate clusterings produce the same partitioning of the data, then we can indeed algorithmically find this optimal partitioning (running time is $n^{O(1/\alpha)}$). When Steiner points *are* allowed and we assume the above for Voronoi-based 2-approximations, we can also find the optimal partitioning.

Note that this assumption is related to the Bilu-Linial perturbation assumption because the optimum clustering for a perturbed distance matrix will be an approximate optimum for the original distance matrix in which each data point is assigned to an "approximately nearest" center. However, as we show in Section 4.1, these assumptions are formally incomparable.

Other notions of stability for clustering have also been considered. The work of Ostrovsky et al. [ORSS06] considers k-means clustering in Euclidean spaces, and considers an instance to be stable if the cost of the optimal k-means solution is small relative to the cost of the $k-1$-means optimal. They show that over such instances, an appropriately-initialized Lloyd's method will achieve a constant factor approximation of the optimal k-means cost. The works of Ben-David et al. [BDvLP06, BDPS07] consider a notion of stability where the n data points come from a distribution. In their work, stability refers to the clustering *algorithm*, which is called stable if it outputs similar clusters for any set of m input points (drawn from the distribution). For k-means, the work of Meila [Mei06] discusses the opposite direction – classifying instances where an approximated solution for k-means is close to the target clustering.

1.1 Our Results

As mentioned above, we consider two notions of stability. The first is the property of resilience to α-perturbations. Here we assume that perturbing any pairwise distances in the metric space up or down by a multiplicative factor of α, does not change the optimal clustering according to the given objective Φ. We show that for $\alpha = \sqrt{3}$, under this assumption we can indeed retrieve the optimal clustering in time $O(n^3)$ (Theorem 5). This result holds not only for the k-median objective, but also for k-means, k-center or any

[1] In particular, any non-Voronoi-based clustering can be easily converted to a Voronoi-based one with at least as good an objective value.

center-based objective function in which points must be assigned to their nearest center in the optimum solution.

We next consider the property of stability with respect to *Voronoi-based approximations*, where the solution is given as a list of k center points, and any other point is assigned to its nearest center. Here we focus in particular on the k-median objective and assume that not only is the k-median optimal equal to the desired target clustering but so is any Voronoi-based $(1+\alpha)$-approximation to the k-median objective. Under this assumption we show how to retrieve the target clustering in polynomial time (Theorem 14). We present results for both finite metrics in Section 4, and for infinite metric spaces (in Section 5). Formally, for finite metrics, under the assumption that there exists $\alpha > 0$ such that all Voronoi-based $(1+\alpha)$-approximations of the k-median objective are the same as the target clustering, we give an $n^{O(1/\alpha)}$ time algorithm that finds the target clustering. For infinite metrics, a case where k-median is NP-hard even for $k = 2$, we show the following. Under the assumption that all Voronoi-based 2-approximations of the k-median objective are the same as the target clustering, we give a $n^{O(1)}$ algorithm that finds the target clustering. Note that there exists a PTAS for the k-median objective in Euclidean space [ARR98] but its running time is exponential in the dimension of the space. In contrast, our result is polynomial time, regardless of the dimension.

At first glance, it is not clear whether this restriction of the assumptions in [BBG09] is significantly weaker or whether it is no more than a minor technicality. In this paper, we show that indeed this assumption holds for a substantially more general set of clustering instances than those considered there. As a result our techniques differ significantly from those of Balcan et al. We refer the reader to Section 4.1.2 for examples and further discussion. We also note that the two notions of stability discussed in this paper are somewhat complementary in the following sense. The first property fixes the nature of the solution to be the exact optimum, yet allows the metric to vary. The latter fixes the metric, yet allows the quality of the solution to vary. However, these two notions are incomparable in a formal sense, and we elaborate on this point in Section 4.1.1. We conclude the paper with discussion and open problems in Section 6.

2 Notation and Preliminaries

We are given a set S of n points in a finite metric space, and we denote $d : S \times S \to \mathbb{R}_{\geq 0}$ as the distance function. Φ denotes the target function we want to optimize over the metric. Unless indicated otherwise, we assume that Φ is the k-median objective, where to minimize Φ we partition the n points into k disjoint subsets and assign a center c_i for each subset. Φ is then measured by $\sum_{i=1}^{k} \sum_{x \in C_i} d(x, c_i)$. The optimal clustering w.r.t. Φ is denoted as $\mathcal{C}^* = \{C_1^*, C_2^*, \ldots, C_k^*\}$, and its cost is denoted as OPT. Clearly, in an optimal solution, we can output a list of k points as centers, $\{c_1^*, c_2^*, \ldots, c_k^*\}$, and assign each x to its nearest center. Alternatively, for a finite metric, given a k-partition $\{C_1^*, C_2^*, \ldots, C_k^*\}$, we can find the best point c_i^* to serve as the least-costly center for every cluster[2]. We use $\mathcal{C}^*$ to denote both the optimal k-partition, and the optimal list of k centers. Given $\mathcal{C}^*$, we denote OPT_i as the contribution of the cluster i to OPT, that is $\mathsf{OPT}_i = \sum_{x \in C_i^*} d(x, c_i^*)$.

3 The α-Perturbation Property

As mentioned above, often when using clustering techniques in practice, one does not know exactly how to best measure distance between data points. Any given method is only an approximation to true "semantic distance" between data objects. Thus, unless the k-median optimal on the given distances is correct by pure luck, it likely is correct (or nearly so) on small perturbations of the given distances as well. Bilu and Linial [BL10] analyze this type of assumption in the context of max-cut, but require stability up to quite large perturbations, roughly $O(n^{1/3})$. Here (for center based clustering objectives) we are able to find the desired clustering for much smaller perturbation levels, namely $\sqrt{3}$.

Formally, given $\alpha > 1$ we consider the following property. If the clustering instance consists of a set of points S and a metric d, then for any function $d' : S \times S \to \mathbb{R}_{\geq 0}$ s.t. $\forall x, y \in S,\ d(x,y)/\alpha \leq d'(x,y) \leq \alpha \cdot d(x,y)$, the optimal clustering $\mathcal{C}^*$ for Φ under d should be equal to the optimal clustering $\mathcal{C}'$ for Φ under d'. We call instances that have this property α-*perturbation resilient* to Φ. Note that in this definition we allow d' to be any arbitrary function and not just a metric.

Our assumption in this section is that our input is α-perturbation resilient to some given center-based clustering objective Φ. In particular, all we require about Φ is that the optimal solution $\mathcal{C}^*$ is defined by some set of centers $c_1^*, \ldots, c_k^*$ and assigns each point $p \in S$ to its nearest center c_i^*. For example, k-median and k-means are both center-based clustering objectives.

Note that perturbation resilience has a natural connection to smoothed analysis. *If* a problem has polynomial smoothed complexity, *then* there is an immediate algorithm to solve instances that are perturbation-resilient: specifically, make random perturbations to the input, solve, and then use the fact that the solution

[2] This is also true when Φ is the k-means objective and data is represented as points in $\mathbb{R}^n$.

to the perturbed instance is by assumption an optimal solution to the original.[3] However, unfortunately the hardness-of-approximation reductions for problems such as k-median immediately imply that these problems remain hard in the smoothed analysis model, so this is not a valid approach here.

A key ingredient in our clustering algorithm is the *tree-clustering* formulation of Balcan et. al [BBV08]. In particular, we prove that if an instance is α-perturbation resilient for $\alpha > \sqrt{3}$ then it also satisfies the "min stability property" (defined below). This property, as shown in [BBV08], is a (necessary and) sufficient condition for the Single-Linkage algorithm to produce a tree such that the target clustering $\mathcal{C}^*$ is some pruning of this tree. We can then use a second dynamic programming step to explore the tree and identify $\mathcal{C}^*$. This is shown in Theorem 5.

Fact 1 *For every point p and its center c_i^* under the optimal clustering, it holds that $d(p, c_j^*) > \alpha^2 d(p, c_i^*)$ for any $j \neq i$.*

Proof: Assume we blow up the distances within cluster C_i^* by a factor of α, and that we also reduce all other distances by a factor of α. As this is a legitimate perturbation of the metric, it still holds that the optimal clustering for Φ under this perturbation is the same as the original optimum. Hence, p is still assigned to the same cluster. Furthermore, since the distances within C_i^* were all changed by the same constant factor, the optimal clustering might as well keep using c_i^* as the center of cluster i. The same holds for any other cluster C_j^*. It follows that even in this perturbed metric, p prefers c_i^* to c_j^*. Hence $\alpha d(p, c_i^*) = d'(p, c_i^*) < d'(p, c_j^*) = d(p, c_j^*)/\alpha$. ■

Corollary 2 *For every point p and its center c_i^*, and for every point p' from a different cluster, it follows that $d(p, p') > (\alpha^2 - 1)d(p, c_i^*)$.*

Proof: Denote by c_j^* the center of the cluster that p' belongs to. Now, if $d(p', c_j^*) \geq d(p, c_i^*)$, we use the fact that $d(p, p') \geq d(p', c_i^*) - d(p, c_i^*) > \alpha^2 d(p', c_j^*) - d(p, c_i^*) \geq (\alpha^2 - 1)d(p, c_i^*)$. Otherwise, $d(p', c_j^*) < d(p, c_i^*)$, and now we use the fact that $d(p, p') \geq d(p, c_j^*) - d(p', c_j^*) > \alpha^2 d(p, c_i^*) - d(p', c_j^*) > (\alpha^2 - 1)d(p, c_i^*)$. ■

We now define the "min-stability" property, and prove that a clustering instance resilient to α-perturbations satisfies this property. For any two subsets A and B of S, we use the notation $d_{\min}(A, B)$ to denote $\min_{a \in A, b \in B}\{d(a, b)\}$.

Definition 3 *A clustering instance satisfies the* min-stability property *if for any two clusters C and C' in the target clustering, and any two subsets $A \subsetneq C$, $A' \subseteq C'$, it holds that $d_{\min}(A, C \setminus A) \leq d_{\min}(A, A')$.*

In words, the min-stability property means that for any set A that is a strict subset of some target cluster C, the closest point to A is a point from $C \setminus A$, and not from some other cluster. The next lemma lies at the heart of our algorithm.

Lemma 4 *Assume $\alpha \geq \sqrt{3}$. Then a clustering instance that is α-perturbation resilient satisfies the min-stability property.*

Proof: Let C_i^*, C_j^* be any two clusters in the target clustering. Let A and A' be any two subsets s.t. $A \subsetneq C_i^*$ and $A' \subseteq C_j^*$. Let $p \in A$ and $p' \in A'$ be the two points which obtain the minimum distance $d_{\min}(A, A')$. Let $q \in C_i^* \setminus A$ be the nearest point to p. Also, denote by c_i^* and c_j^* the centers of clusters C_i^* and C_j^* respectively.

For the sake of contradiction, assume that $d_{\min}(A, C_i^* \setminus A) \geq d_{\min}(A, A')$. Suppose $c_i^* \notin A$. This means that $d(p, p') = d_{\min}(A, A') \leq d_{\min}(A, C_i^* \setminus A) \leq d(p, c_i^*)$. As $\alpha \geq \sqrt{3} > \sqrt{2}$, this contradicts Corollary 2. We deduce that $c_i^* \in A$.

Now, since $c_i^* \in A$, it follows that $d(q, c_i^*) \geq d(p, p') > (3 - 1)d(p, c_i^*) = 2d(p, c_i^*)$, so $d(p, c_i^*) < d(q, c_i^*)/2$. We therefore have that $d(p', c_i^*) \leq d(p, p') + d(p, c_i^*) \leq 3d(q, c_i^*)/2$. This implies that $d(p', c_j^*) < d(p', c_i^*)/\alpha^2 < d(q, c_i^*)/2$, and thus $d(q, c_j^*) \leq d(q, c_i^*) + d(c_i^*, p) + d(p, p') + d(p', c_j^*) < 3d(q, c_i^*) \leq \alpha^2 d(q, c_i^*)$. This contradicts Fact 1. ■

[3] Technically, perturbation resilience is defined in terms of multiplicative perturbations, whereas smoothed analysis is typically defined in terms of additive perturbations, so one would want to use a multiplicative version of smoothed complexity.

Theorem 5 *For $\sqrt{3}$-perturbation resilient instances, there exists a polynomial time algorithm that finds the optimal k-median (or k-means) clustering.*

Proof: We run the Single-Linkage (i.e. Kruskal's) algorithm, until we reach a single cluster. Running the algorithm produces a hierarchical clustering of the data (a tree on subsets) in which each node represents a cluster produced at some intermediate stage of the algorithm. By Lemma 4, the data satisfies the "min-stability" property of [BBV08], which, as shown there, is sufficient to guarantee that some pruning of this hierarchy is the target clustering.

We then find the k-median optimal clustering using dynamic programming by examining the entire hierarchy produced by single-linkage. The optimal k-clustering of a tree-node is either the entire subtree as one cluster (if $k = 1$), or the minimum over all choices of k_1-clusters over its left subtree and k_2-clusters over its right subtree (if $k > 1$). Here k_1, k_2 are positive integers, such that $k_1 + k_2 = k$. Therefore, we just traverse the tree bottom-up, recursively solving the k-median (or k-means) problem for each tree-node. A naive implementation of this algorithm takes $O(n(n^2 + k^2))$ time, but a slightly more careful one takes only $O(n^2 \log n + nk^2)$ time. ■

Interestingly, using single-linkage in the usual way (namely, stopping when there are k clusters remaining) is *not* sufficient to produce a good clustering. Figure 1 displays an instance for which the usual single-linkage fails, yet satisfies $\sqrt{3}$-perturbation resilience.

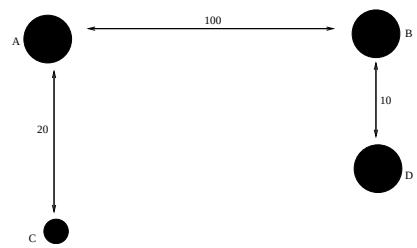

Figure 1: An example showing that the usual version of single-linkage fails. OPT consists of 3 clusters: $\{A, C\}, \{B\}, \{D\}$ where C is much smaller in size than the rest. Since C contains significantly less points than A,B, or D, this instance is stable – even if we perturb distances by a factor of $\sqrt{3}$, the cost of any alternative clustering is higher than OPT. However, because $d(A, C) > d(B, D)$ it follows that the usual version of single-linkage will unite B and D, and only then A and C.

We should also note that the $\sqrt{3}$ constant in Lemma 4 is a tight bound. To see this, consider a placement of all points on a line. In particular, set the following 5 points in a line, from left to right (See Figure 2): q, c, p, p', c'. In order to assure that c and c' will be the cluster centers, add a few additional points, all of distance ϵ from c or c'. Now, set the following distances: $d(q, c) = 2$, $d(c, p) = 1$, $d(p, p') = 2 - \epsilon$ (for some infinitesimally small $\epsilon > 0$) and $d(p', c') = 1$. The single linkage algorithm first connects p and c, then p' and c', then it can arbitrarily choose between connecting q with $\{c, p\}$, or connected $\{c, p\}$ with $\{c', p'\}$, as both the min-distances are 2. It remains to check that the distance of each point to its center is no more than $1/3$ its distance to the other center.

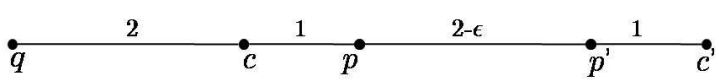

Figure 2: An example showing that single-linkage fails for $\alpha < \sqrt{3}$.

4 $(1 + \alpha)$-Stability to Voronoi-Based Approximations for k-Median

Let us start by formally defining the stability property of approximations.

Definition 6 *We call a k-partition* Voronoi-based, *if it is induced from a list of k-centers, where each point is assigned to its nearest center. We say a clustering instance has the property of* (1+α)-stability to Voronoi-based approximations *for a center-based objective function Φ, if any Voronoi-based $(1 + \alpha)$-approximation to Φ is in fact equal to $\mathcal{C}^*$ (up to renaming of the clusters).*

Denote the optimal solution of Φ as a partition $\{C_1^*, C_2^*, \ldots, C_k^*\}$. Suppose we apply an $(1+\alpha)$-approximation algorithm and get a list of k points, $p_1, p_2, \ldots, p_k$, as an output. By assigning each element in the clustering instance to its closest p_i, we get a Voronoi-based solution, denoted $\{D_1, D_2, \ldots, D_k\}$. Then $(1+\alpha)$-stability to Voronoi-based approximations implies that there exists a permutation $\pi \in S_k$ such that $D_{\pi(i)} = C_i^*$.

What we prove is that the optimal solution of a clustering instance that satisfies $(1+\alpha)$-stability to Voronoi-based approximations for the k-median objective, can be found in polynomial time. Therefore, the problem, which is NP-hard even to approximate beyond a factor of $1+1/e$ (see [GK98]), becomes polynomial time solvable for such instances. For ease of exposition, we present the algorithm and prove its correctness in stages. First we present an algorithm for 2-stability to Voronoi-based approximations, restricted to a particular type of instances. Then we extend it to all instances satisfying 2-stability to Voronoi-based approximations, and finally we handle all instances satisfying the property for $(1+\alpha)$-approximations (for any $\alpha > 0$).

4.1 Comparison of Stability Notions

Before presenting the algorithm, we wish to discuss the similarities and differences of Definition 6 to the property of resilience to α-perturbations and the definitions of Balcan et al. [BBG09] and Ostrovsky et al. [ORSS06].

4.1.1 Comparison with Resilience to Perturbations

We show here that resilience to α-perturbations is formally incomparable to the α-stability property for Voronoi-based approximations. For example, even if an instance has the 2-stability property for k-median, it still does not assure us that by perturbing its distances by a factor of 2, we get the same optimum. In Section 4.1.2 we present an instance (shown in Figure 3) which is 2-stable under Voronoi-based approximations, yet is not resilient even to very small perturbations.

To see that the other direction also does not hold, consider the following instance. We have k disjoint sets, $S_1, S_2, \ldots, S_k$, each contains t points. The distance between any two points that belong to the same S_i is 1. The distance between any two points that belong to two different sets $S_i \neq S_j$ is 5. Clearly, even if we perturb distances by a factor of 2, each point is still more attracted to its own cluster than to any other point in any other cluster. In contrast, the OPT solution has a cost of $k(t-1)$, but by placing two centers in one cluster we obtain a solution whose cost is $(k-2)(t-1)+t-2+5t \leq (k+4)(t-1) \leq 2\mathsf{OPT}$. Therefore, this instance does not have 2-stability to Voronoi-based approximations.

As an additional comment, we would like to add that if we are guaranteed that *all* k-partitions whose cost is at most $\alpha^2\mathsf{OPT}$ are exactly the same as the optimal solution (as in the $(\alpha^2, 0)$ notion of [BBG09]), this assures that the instance is resilient to α-perturbations. In particular, an optimal solution to an α-perturbation must be an α^2 approximation of OPT under the original metric (though it may not be Voronoi based).

4.1.2 Comparison with Balcan et al. and Ostrovsky et al. Assumptions

The instances considered in [BBG09] have the property that *any* k-partition which is a $(1+\alpha)$-approximation of OPT, yields a clustering which is ϵ-close to the target clustering. In comparison, here we only constrain a subset of all of these partitions, namely – all partitions that result from assigning each point to its closest center. We show that by focusing on instances with restrictions only over Voronoi-based partitions, we allow for a much broader class of clustering instances. To demonstrate, we now present an instance that does *not* satisfy the assumption of [BBG09], yet satisfies our version of the 2-stability property of Voronoi-based approximations.

Suppose n consists of $k+1$ different subsets. k subsets are of equal size s and are completely separated - within each subset the distances are 0, and between any points from two different subset the distance is 1. We call them the k corners. The additional $k+1$ subset is smaller, it's size is only $s/2$, and it is placed "in the middle". That is, all of its points are about in distance $1/2$ from the points in the k clusters, only with some variations. Partition this middle set into k "slices" of equal size, and assign the distances such that the distance of between any two points from slice i and corner i is $1/2-\epsilon$, whereas the distances between slice i and corner j is exactly $1/2$.

Clearly, the optimal k-median solution assigns one center in each corner. This way, we only split the middle subset, and pay $1/2-\epsilon$ for each point in this middle subset, so OPT is roughly $s/4$. Furthermore, to find any other Voronoi-based solution that yields a different clustering, we must push (at least) one of the centers to either the middle, or to a corner. In each case, we move a whole corner by distance at least $1/2$, so we pay $s/2$ just for that transition. Therefore, this instance indeed satisfies the 2-stability property. See Figure 3 for the case $k=3$.

In contrast, notice that it does not satisfy the property that any *arbitrary* partition into k subsets which approximates OPT also approximates the target clustering. To see that, consider a partition that assigns each slice in the middle subset to its second nearest corner. Every point in the middle subset now increased its cost by at most ϵ, yet the assignment of points to cluster changed by $s/2$ points, which is a $\Omega(1/k)$ fraction! This example also demonstrates the difference between our stability assumption and the one of Ostrovksy

et al [ORSS06]. The cost of the optimal $(k-1)$-median partition is roughly s;[4] the cost of the optimal k-median partition is roughly $s/4$; and the cost of the $(k+1)$-median optimal is $o(s)$. Therefore, this instance is considered stable in the Ostrovsky et al. sense only for $k+1$ partitions.

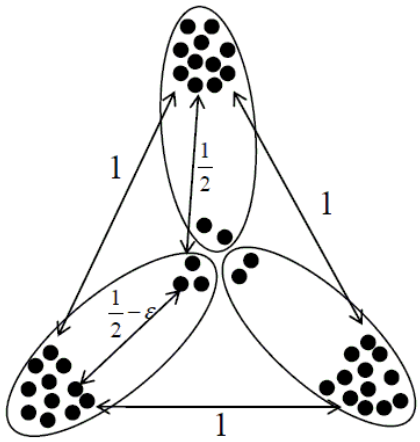

Figure 3: An example satisfying 2-stability to Voronoi-based approximations for k-median but not the $(2,0)$ property in [BBG09] or resilience even to $\alpha = 1+\epsilon$ perturbations. Here $k=3$. This example points out that algorithms for clustering under 2-stability to Voronoi-based approximations must handle such "adversarially designed" distance functions correctly.

4.2 An Algorithm for 2-Stability to Voronoi-Based Approximations

Before presenting the theorem regarding the $(1+\alpha)$-stability property, we first discuss the case where $\alpha = 1$, i.e. the 2-stability property. This case provides the the outline for the general theorem.

We say a target cluster C_i^* is *small* if $\mathsf{OPT}_i \leq \frac{\mathsf{OPT}}{16}$, otherwise, we say C_i^* is *large*. For now, let us assume that all clusters are small, i.e., no cluster costs a $\Omega(1)$ fraction of the optimal solution. In this section we show an algorithm for solving this particular case (i.e. the target clustering consists only of small clusters and satisfies the 2-stability property). In later sections, we handle large clusters as well, then modify the general algorithm to deal with instances satisfying the $(1+\alpha)$-stability property.

Our algorithm populates a list 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 algorithm works in stages, where in stage s it tries to capture a cluster of size (roughly) s. It does so by setting r, a certain radius, and accumulating elements according to the number of points that fall into their r-ball.

Initially, set $Q = \emptyset$.
for $s = n$ to 1 **do**
  Set $r = \frac{\mathsf{OPT}}{4s}$.
  Remove any point x such that $d(x,Q) < 2r$. Here, $d(x,Q) = min_{T \in Q; y \in T} d(x,y)$.
  For every remaining element x, count the number of points that are of distance $\leq r$ from x, that is – the size $|B(x,r)|$.
  Connect any two remaining point a and b if $d(a,b) \leq r$ and both $|B(a,r)| > \frac{s}{2}$ and $|B(b,r)| > \frac{s}{2}$.
  Add every connected component of size $> \frac{s}{2}$ to Q, where every component corresponds to one element in Q.
end for
for any choice of k components out of Q (we later show that $|Q| < k+6$) **do**
  Arbitrarily choose a point in each component.
  Partition all n points according to the nearest point among these k centers.
  Re-evaluate the k-median cost of a partition by choosing the best center for each subset.
  If these k centers give a solution of cost OPT – output these k centers and halt.
end for

[4] place one center in the "middle" set, instead of two in two corners

4.2.1 Proof of Correctness

Before going into the proof of correctness of the algorithm, note the following two facts.

Fact 7 *Given a target cluster C_i^* with center c_i^* and point $p \notin C_i^*$, it holds that $d(p, c_i^*) > \frac{\mathsf{OPT}}{|C_i^*|}$.*

Proof: Assume that p is not the center of any other target cluster. Let's assume that instead of having $\{c_1^*, c_2^*, \ldots, c_k^*\}$ as the centers of the clustering, we move the center c_i^* to the point p, while maintaining all other centers c_j^* as they were. Using these new centers we must obtain a different clustering, because p is not assigned to its original cluster center. Therefore, the cost of this new clustering has to be $> 2\mathsf{OPT}$, so this change must increase the cost by at least OPT. However, this increase in cost only emanates from points in C_i^*, as the cost of any other point can only decrease. Furthermore, the change in cost for cluster C_i^* is upper bounded by $\sum_{x \in C_i^*} d(x,p) - \sum_{x \in C_i^*} d(x, c_i^*) \leq \sum_{x \in C_i^*} (d(x, c_i^*) + d(c_i^*, p)) - \sum_{x \in C_i^*} d(x, c_i^*) = |C_i^*| d(c_i^*, p)$. The required follows. Note that alternatively, we could assign all points in C_i^* to some c_j^*, thus increasing cost by at most $|C_i^*| d(c_i^*, c_j^*)$, so the claim also holds when $p = c_j^*$ for some j. ■

Fact 8 *We define the* inner ring *of C_i^* as the set $\left\{x;\ d(x, c_i^*) \leq \frac{\mathsf{OPT}}{8|C_i^*|}\right\}$. Then if C_i^* is a small cluster, it must hold that more than half of its points are contained within the inner ring.*

Proof: This follows from Markov's inequality. If at least $|C_i^*|/2$ points are outside of the inner ring, then $\mathsf{OPT}_i > \frac{|C_i^*|}{2} \cdot \frac{\mathsf{OPT}}{8|C_i^*|} = \mathsf{OPT}/16$. This contradicts the fact that C_i^* is small. ■

Our high level goal is to show that for any cluster C_i^* in the target clustering, we insert a component T_i that will be contained within C_i^* and will only contain 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 9 *Let T be any component added to Q. Let s be the stage in which we add T to Q. Let C_i^* be any small cluster s.t. $s \geq |C_i^*|$. Then (a) T does not contain any point z s.t. the distance $d(c_i^*, z)$ lies within the range $\left[\frac{1}{2}\frac{\mathsf{OPT}}{|C_i^*|}, \frac{3}{4}\frac{\mathsf{OPT}}{|C_i^*|}\right]$, and (b) T cannot contain both a point p_1 s.t. $d(c_i^*, p_1) < \frac{1}{2}\frac{\mathsf{OPT}}{|C_i^*|}$ and a point p_2 s.t. $d(c_i^*, p_2) > \frac{3}{4}\frac{\mathsf{OPT}}{|C_i^*|}$.*

Proof: We prove (a) by contradiction. Assume T contains a point z s.t. $\frac{1}{2}\frac{\mathsf{OPT}}{|C_i^*|} \leq d(c_i^*, z) \leq \frac{3}{4}\frac{\mathsf{OPT}}{|C_i^*|}$. Let p be any point in the ball $B(z, \frac{\mathsf{OPT}}{4s})$. Then by the triangle inequality we have that $d(c_i^*, p) \geq d(c_i^*, z) - d(z, p) \geq \frac{1}{4}\frac{\mathsf{OPT}}{|C_i^*|}$, and similarly $d(c_i^*, p) \leq d(c_i^*, z) + d(z, p) \leq \frac{\mathsf{OPT}}{|C_i^*|}$. Due to Fact 7 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 $B(z, \frac{\mathsf{OPT}}{4s})$ contains more than $s/2 \geq |C_i^*|/2$ many points. So more than half of the points in C_i^* fall outside the inner ring of C_i^*, which contradicts Fact 8.

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 $\frac{\mathsf{OPT}}{4s} \leq \frac{\mathsf{OPT}}{4|C_i^*|}$. As $d(c_i^*, p_1) < \frac{1}{2}\frac{\mathsf{OPT}}{|C_i^*|}$ and $d(c_i^*, p_2) > \frac{3}{4}\frac{\mathsf{OPT}}{|C_i^*|}$, there must exist a point z along the path whose distance from c_i^* falls in the range $\left[\frac{1}{2}\frac{\mathsf{OPT}}{|C_i^*|}, \frac{3}{4}\frac{\mathsf{OPT}}{|C_i^*|}\right]$, contradicting (a). ■

Claim 10 *Let C_i^* be any cluster in the target clustering. By stage $s = |C_i^*|$, the algorithm adds to 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 T. Also, assume for now that none of the inner ring points were removed in step (4). 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 $\frac{\mathsf{OPT}}{8|C_i^*|}$ from the center, so every pair of inner ring points has a distance of at most $\frac{\mathsf{OPT}}{4|C_i^*|}$. Hence, when we reach stage $s = |C_i^*|$, any ball of radius $r = \frac{\mathsf{OPT}}{4s} = \frac{\mathsf{OPT}}{4|C_i^*|}$ 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 in step (4). 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 some 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. $d(x,y) < \frac{\mathsf{OPT}}{2s}$, and so $d(c_i^*, y) \leq \frac{\mathsf{OPT}}{8|C_i^*|} + \frac{\mathsf{OPT}}{2s} \leq \frac{5}{8}\frac{\mathsf{OPT}}{|C_i^*|}$. Let $s' \geq s \geq |C_i^*|$ be the previous stage in which we added the component T' to Q. As Lemma 9 applies to T', we deduce that $d(c_i^*, y) < \frac{1}{2}\frac{\mathsf{OPT}}{|C_i^*|}$. 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 8 that some point $w \in T$ must belong to a different cluster C_j^*. By Fact 7, we have that $d(c_i^*, w) > \frac{\mathsf{OPT}}{|C_i^*|}$. The existence of both y and w in T contradicts Lemma 9. ■

We call a component $T \in Q$ *good* if it contains an inner ring point of some cluster C_i^*, and *bad* otherwise. We now discuss the properties of good components.

Claim 11 *Let T be a good connected component added to Q, containing an inner ring point from cluster C_i^*. Then: (a) all points in T are of distance at most $\frac{\mathsf{OPT}}{2|C_i^*|}$ from c_i^*, (b) T is fully contained in C_i^*, and (c) no other component $T' \neq T$ in Q contains an inner ring point from C_i^*.*

Proof: As we do not know (c) in advance, it might be the case that 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 Q. We show (a), (b) and (c) hold for T, and deduce that T is the only good component to contain an inner ring point from C_i^*. Denote the stage in which T was inserted to Q as s. Due to the previous claim, we know $s \geq |C_i^*|$, and so Lemma 9 applies to T.

To show (a), assume T contains a point z s.t. $d(c_i^*, z) > \frac{\mathsf{OPT}}{2|C_i^*|}$. Lemma 9 guarantees that $d(c_i^*, z) > \frac{3}{4}\frac{\mathsf{OPT}}{|C_i^*|}$. But we know T contains some inner ring point x from C_i^*, so $d(c_i^*, x) \leq \frac{1}{8}\frac{\mathsf{OPT}}{|C_i^*|} < \frac{1}{2}\frac{\mathsf{OPT}}{|C_i^*|}$. This contradicts part (b) of Lemma 9. Since we now know (a) to hold, we have that all points in T belong to C_i^* because of Fact 7, so (b) follows. 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 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. Any other inner ring point from C_i^* is of distance at most $\frac{\mathsf{OPT}}{4|C_i^*|} < \frac{\mathsf{OPT}}{2s}$ from x. So, in the next stage, when the algorithm applies step (4) and removes all points whose distance from x is $< \frac{\mathsf{OPT}}{2s}$, it must remove all inner-ring points of C_i^* that weren't added to T. Thus, other than T, no component in Q can contain an inner ring point of C_i^*. ■

Corollary 12 *For each small cluster C_i^*, there will be exactly one good component $T \in Q$.*

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

Claim 13 *We have less than 6 bad components.*

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

Case 1: $s \geq |C^*|$. We apply Lemma 9, and deduce that either $d(c^*, y) < \frac{1}{2}\frac{\mathsf{OPT}}{|C^*|}$ or that $d(c^*, y) > \frac{3}{4}\frac{\mathsf{OPT}}{|C^*|} \geq \frac{3}{4}\frac{\mathsf{OPT}}{s}$. 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^*. Fact 7 gives that $d(c^*, w) > \frac{\mathsf{OPT}}{|C^*|} > \frac{3}{4}\frac{\mathsf{OPT}}{|C^*|}$. Part (b) of Lemma 9 assures us that all points in T are also far from c^*.

Case 2: $s < |C^*|$. Using Claim 10 we have that some good component containing a point x from the inner ring of C^* was already added to Q. Since y survives step (4), we deduce that $d(c^*, y) \geq d(x, y) - d(c^*, x) \geq \frac{\mathsf{OPT}}{2s} - \frac{\mathsf{OPT}}{8|C^*|} > \frac{3}{8}\frac{\mathsf{OPT}}{s}$.

All points in T have distance $> \frac{3\mathsf{OPT}}{8s}$ from their respective centers in the target clustering, and recall that T is added to Q because T contains at least $s/2$ many points. Therefore, the contribution of all elements in T to OPT is at least $\frac{3\mathsf{OPT}}{16}$. It follows that we can have no more than $16/3 < 6$ such bad components. ■

We can now prove the correctness of the algorithm presented in Section 4.2.

Theorem 14 *The algorithm outputs the correct centers of the target clustering $\mathcal{C}^*$.*

Proof: Using Corollary 12, it follows that there exists some choice of k components, such that all components are good, and therefore, each component is contained within a single, unique, cluster of the target clustering $\mathcal{C}^*$. Fix that choice. Denote the arbitrary points we pick, one point from each good component, as $p_1, p_2, \ldots, p_k$. From claim 11 it follows that $d(c_i^*, p_i) \leq \frac{\mathsf{OPT}}{2|C_i^*|}$.

Assume that by using p_i's as centers we do not get the target clustering. So there exists some i, j and a point z such that the point z is assigned to the cluster of p_i, whereas in the target clustering it is assigned to C_j^*. I.e., $d(z, c_i^*) > d(z, c_j^*)$ whereas $d(z, p_i) < d(z, p_j)$. (Assume ties are broken in favor of lexicographic order.)

Imagine that we are taking the optimal k centers, $c_1^*, c_2^*, \ldots, c_k^*$, and we replace c_i^* by p_i, and c_j^* by p_j. We use these k points to produce a Voronoi based clustering. In this clustering it is evident that we do *not* assign z to cluster j, because z prefers p_i over p_j. Therefore, this solution differs from the optimal solution, and by the 2-stability property of Voronoi-based approximations, this solution has cost $> 2\mathsf{OPT}$. We now apply the same argument from the proof of Fact 7 to derive a contradiction. In this new clustering, only points in C_i^* and C_j^* can increase their cost, and using the triangle inequality we can upper bound the cost by $\mathsf{OPT} + \sum_{x \in C_i^*} d(c_i^*, p_i) + \sum_{x \in C_j^*} d(c_j^*, p_j) = \mathsf{OPT} + |C_i^*| \cdot \frac{\mathsf{OPT}}{2|C_i^*|} + |C_j^*| \cdot \frac{\mathsf{OPT}}{2|C_j^*|} \leq 2\mathsf{OPT}$.

To complete the proof, note that once we have the target partitioning $\{C_1^*, C_2^*, \ldots, C_k^*\}$, we can find the best center for each cluster. By re-calculating the k-median cost using the best centers, we verify that we indeed obtain the optimal clustering. ■

4.3 Handling Large Clusters

The algorithm presented in Section 4.2 only dealt with the case where all clusters were small. We now remove this restriction. Note that we only have less than 16 large clusters and we can handle them by brute force – simply guessing the number of large clusters and their centers. The general algorithm is as follows.

1: **for** $l = 0$ to 15, and any choice of l points $p_1, p_2, \ldots, p_l$ **do**
2: Run the algorithm from Section 4.2, with the exception that initially $Q = \{\ \{p_1\},\ \{p_2\},\ \ldots,\ \{p_l\}\ \}$.
3: If a clustering of cost OPT is found, output that clustering and halt.
4: **end for**

Suppose that there are $l \leq 15$ large clusters, and we guessed the right l centers of these clusters. We show that the analysis of algorithm presented in Section 4.2 extends to this case.

Fact 7 does not change, and neither does all the facts and claims about the inner rings of small clusters. As for bad clusters, the proof of Claim 13 still holds. Assuming we pick the right centers for the large clusters, even if $y \in T$ belongs to a large cluster, we still have that $d(c^*, y) > \frac{\mathsf{OPT}}{2s}$, and so it still holds that each bad component contributes $\frac{3\mathsf{OPT}}{16}$ to the optimal solution. Finally, observe that Theorem 14 still holds by the assumption that the first l components are the centers of the large clusters.

4.4 Runtime analysis

A naive implementation of the first part of algorithm in Section 4.2 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). The second part requires k^5 iterations, in which we do $O(n^2)$ work (of finding the best center for each cluster). Overall, the running time of the algorithm presented in Section 4.2 is $O(n^3 + k^5 n^2)$.

The algorithm presented in Section 4.3 requires up to n^{16} iterations of the former algorithm for every possible brute force guess of centers. This gives a total running time of $O(n^{19} k^5)$.

4.5 Extension to $(1+\alpha)$-Stability to Voronoi-Based Approximations

The algorithm in Section 4.3 can be adapted to handle instances which have $(1+\alpha)$-stability to Voronoi-based approximations. Note that all claims and lemmas in the previous section depend only on the increase in cost we get by changing the clustering. Therefore, if our instance satisfies $(1+\alpha)$-stability to Voronoi-based approximations, simply replace every occurrence of OPT in the previous algorithm with $\alpha\mathsf{OPT}$, and the same proof goes through.

In more details, it now holds that $d(c^*, p) > \frac{\alpha\mathsf{OPT}}{|C^*|}$ for every target cluster C^*, and a point $p \notin C^*$. We now define a large cluster as a cluster whose cost is $\mathsf{OPT}_i > \frac{\alpha\mathsf{OPT}}{16}$. Hence, there are less than $16/\alpha$ large clusters, and we can brute-force guess their centers. For every small cluster C_i^*, we define its inner ring as all points whose distance from the center is at most $\frac{\alpha\mathsf{OPT}}{8|C_i^*|}$. Therefore, if we form balls of radii $r = \frac{\alpha\mathsf{OPT}}{4|C_i^*|}$ around points from the inner ring, they all contain the inner ring of C_i^*. The proof of correctness now follows from the same arguments as before. We present the formal algorithm and its proof in Appendix A.

5 2-Stability to Voronoi-Based Approximations for k-Median, for Infinite Metric Spaces

So far we've only considered finite metrics where the centers in the optimal solution can be assumed to be points in the data set. However, instances often reside in an infinite metric space (e.g., a Euclidean space), in which case the centers of the k-median optimal solution need not be data points, but rather any point in the metric space. Note that in this case, even solving k-median for $k = O(1)$ is non-trivial. It is easy to see that in infinite metric spaces, the center points lie in the convex hull of the given n data points. In this section, we consider metric spaces that allow "Steiner points". In particular, we allow setting cluster centers at any convex combination of two data-points. Namely, given two finite points x, y, we consider all points between them of the form $\tilde{x} = \lambda x + (1 - \lambda)y$. The distance of any point p to $\tilde{x}$ is at most $\min\{d(p, x) + \lambda d(x, y),\ d(p, y) + (1 - \lambda)d(x, y)\}$. We present a result that holds for any metric space that allows the use of Steiner points (such as the Euclidean space). We show that under the assumption that any Voronoi based 2 approximation to the k-median objective is equal to the target clustering, we can find the target clustering in polynomial time. We refer the reader to Appendix B for the algorithm and its analysis. We stress that for infinite metric spaces, we can only handle the 2-stability property and we do not know how to handle values below 2.

6 Conclusions and Open Problems

There are several natural open questions left by this work. Perhaps the most clear open question is whether one can reduce the $\alpha = \sqrt{3}$ factor given for efficient clustering under stability with respect to α-perturbations in the metric. Alternatively, perhaps there is a lower bound showing that NP-hardness results can be extended to instances satisfying this condition.

Another open question involves the relation to the $((1+\alpha), \epsilon)$-property of Balcan et al. [BBG09]. Balcan et al. assume that all $(1 + \alpha)$-approximations for the k-median are ϵ-close to the target clustering in terms of how points are clustered, and under this assumption they provide polynomial time algorithms for achieving clusterings that are $O(\epsilon)$-close to the target. Here, we assumed that all $(1+\alpha)$-Voronoi-based approximations for the k-median are *identical* to the target clustering and gave algorithms for finding the optimal solution exactly. Can we relax our assumption to approximations that are ϵ-close to the target and still have polynomial time algorithms that cluster well? Namely, if we assume that all $(1 + \alpha)$-Voronoi-based approximations for the k-median objective are ϵ-close to the optimal solution – is there an efficient algorithm that outputs a clustering which is $O(\epsilon)$-close to the target clustering? Another interesting direction is whether we can extend our results on stable Voronoi based approximations to other common objectives such as k-means and k-center.

On a more general note, this paper considers a natural setting in which we solve or approximate one objective while hoping to obtain the solution for a different one. For example, in this paper we optimize the k-median objective, hoping to obtain our target clustering. This setting often occurs in real-life problems and as with [BBG09, BL10] by making natural assumptions explicit, one might be able to bypass existing hardness results. In addition, this approach might be useful in the study of other NP-hard problems. We view this as a complementary approach to the more commonly analyzed framework of smoothed analysis [ST04].

References

[AGK+01] 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 '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, 2001.

[ARR98] Sanjeev Arora, Prabhakar Raghavan, and Satish Rao. Approximation schemes for euclidean k-medians and related problems. In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 1998.

[BB09] Maria-Florina Balcan and Mark Braverman. Finding low error clusterings. In *COLT*, 2009.

[BBG09] Maria-Florina Balcan, Avrim Blum, and Anupam Gupta. Approximate clustering without the approximation. In *SODA '09: Proceedings of the Nineteenth Annual ACM -SIAM Symposium on Discrete Algorithms*, 2009.

[BBV08] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. A discriminative framework for clustering via similarity functions. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, 2008.

[BCR01] Yair Bartal, Moses Charikar, and Danny Raz. Approximating min-sum k-clustering in metric spaces. In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, 2001.

[BDPS07] Shai Ben-David, Dávid Pál, and Hans-Ulrich Simon. Stability of -means clustering. In *COLT*, pages 20–34, 2007.

[BDvLP06] Shai Ben-David, Ulrike von Luxburg, and Dvid Pl. A sober look at clustering stability. In Gábor Lugosi and Hans-Ulrich Simon, editors, *COLT*, volume 4005 of *Lecture Notes in Computer Science*, pages 5–19. Springer, 2006.

[BL10] Yonatan Bilu and Nati Linial. Are stable instances easy? In *1st Symposium on Innovations in Computer Science (ICS)*, pages 332–341, 2010.

[CGTS99] Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k-median problem. In *STOC '99: Proceedings of the thirty-first annual ACM symposium on Theory of computing*, 1999.

[dlVKKR03] W. Fernandez de la Vega, Marek Karpinski, Claire Kenyon, and Yuval Rabani. Approximation schemes for clustering problems. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, 2003.

[GK98] Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. In *Journal of Algorithms*, pages 649–657, 1998.

[JMS02] Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems (extended abstract). In *In Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 731–740, 2002.

[KSS04] Amit Kumar, Yogish Sabharwal, and Sandeep Sen. A simple linear time (1+ ϵ) -approximation algorithm for k-means clustering in any dimensions. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, 2004.

[Mei06] Marina Meilă. The uniqueness of a good optimum for k-means. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 625–632, New York, NY, USA, 2006. ACM.

[ORSS06] Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman, and Chaitanya Swamy. The effectiveness of lloyd-type methods for the k-means problem. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 165–176, Washington, DC, USA, 2006. IEEE Computer Society.

[ST04] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3), 2004.

A An Algorithm $(1+\alpha)$-Stability to Voronoi-Based Approximations

We present the algorithm for finding the target clustering of any instance satisfying $(1+\alpha)$ to Voronoi-based approximations in this section.

for $l = 0$ to $(\frac{16}{\alpha} - 1)$, and any choice of l points $p_1, p_2, \ldots, p_l$ **do**
  Initialize $Q = \{\ \{p_1\},\ \{p_2\},\ \ldots,\ \{p_l\}\ \}$.
  for $s = n$ to 1 **do**
    Set $r = \frac{\alpha \mathsf{OPT}}{4s}$.
    Remove any point x such that $d(x, Q) < 2r$. Here, $d(x, Q) = min_{T \in Q; y \in T} d(x, y)$.
    For every remaining element x, count the number of points that are of distance $\leq r$ from x, that is – the size $|B(x, r)|$.
    Connect any two remaining point a and b if $d(a, b) \leq r$ and both $|B(a, r)| > \frac{s}{2}$ and $|B(b, r)| > \frac{s}{2}$.
    Add every connected component of size $> \frac{s}{2}$ to Q, where every component corresponds to one element in Q.
  end for
  for any choice of k components out of Q **do**
    Arbitrarily choose a point in each component.
    Partition all n points according to the nearest point among these k centers.
    Re-evaluate the partition by choosing the best center for each subset.
    If these k centers give a solution of cost OPT – output these k centers and halt.
  end for
end for

The running time of this algorithm is $O(n^{(\frac{16}{\alpha}+3)} k^{\frac{6}{\alpha}})$, using a straight-forward implementation. The proof of correctness for this algorithm is very similar to the proof presented in Section 4.2.1. First, note that for every target cluster C_i^* and a point $p \notin C_i^*$ it holds that $d(c_i^*, p) > \alpha \frac{\mathsf{OPT}}{|C_i^*|}$, as otherwise we can move the center of C_i^* to p and increase the cost by no more than $\alpha \mathsf{OPT}$. Define a *large* cluster as a cluster s.t. $\mathsf{OPT}_i > \frac{\alpha \mathsf{OPT}}{16}$, and otherwise call a cluster *small*. Clearly, there are less than $16/\alpha$ large clusters, and we can brute-force guess their centers. For every small cluster C_i^*, we define its inner ring as all points whose distance from the center is at most $\frac{\alpha \mathsf{OPT}}{8|C_i^*|}$. Therefore, if we form balls of radii $r = \frac{\alpha \mathsf{OPT}}{4|C_i^*|}$ around points from

the inner ring, they all contain the inner ring of C_i^*.

As in Section 4.2.1, Lemma 9 holds in this case with OPT being replaced by $\alpha\mathsf{OPT}$. Next, as in Claim 10, it is easy to show that for each small cluster C_i^*, by stage $s = |C_i^*|$, a *good* component will be added to Q. All the properties of a *good* component which were proved in Claim 11 remain true with the change that points in a good component corresponding to C_i^* will be at a distance of $\frac{\alpha\mathsf{OPT}}{2|C_i^*|}$ from the center of the cluster.

Now, we claim that we have no more than $16/3\alpha$ bad components in Q. Suppose we insert a bad component into Q in stage s. That means this bad component contains $s/2$ points. Any point p in this component has to have distance $\geq \frac{3\alpha\mathsf{OPT}}{8s}$ from its target cluster's center, from the same considerations mentioned in Claim 13. Therefore, all points in the bad component contribute $3\alpha\mathsf{OPT}/16$ to the total cost of OPT, and hence we have less than $16/3\alpha$ such bad components.

Assume we have l large target clusters, and k' small target clusters. Also assume we picked the l centers of the large clusters, and the k' good components out of Q. Therefore we pick k' points, each from a different cluster, each within distance $\alpha\frac{\mathsf{OPT}}{2|C_i^*|}$ from its center, and additional l points, each is the center of a large cluster. We claim that by assigning points as these $k = k' + l$ points as cluster centers, we get the target clustering. The proof is essentially the same as in Theorem 14. If this wasn't the case, then there exists a change of no more than two cluster centers changing the clustering of at least one point. Hence, just by changing (at most) two cluster centers we increase the cost by at least $\alpha\mathsf{OPT}$. However, our new center points are of distance at most $\alpha\frac{\mathsf{OPT}}{2|C_i^*|}$ from their original center points, so we can increase our cost by no more than $\alpha\mathsf{OPT}$.

B An Algorithm for 2-Stability to Voronoi-Based Approximations, for Infinite Metric Spaces

The algorithm presented here is a variation of the algorithm presented in 4.3. The major difference between the two is that now we collect components by sheer size, without considering the number of points contained in its vicinity. The correctness proof of this algorithm follows the outline of the proof presented in Section 4, yet it is simpler on some aspects.

We call a cluster C_i^* *small* if $\mathsf{OPT}_i \leq \frac{\mathsf{OPT}}{64}$, otherwise, we call C_i^* *large*. The algorithm is given below.

Guess l = # of large clusters. Note that l is at most 63.
Guess l points $p_1, p_2, \ldots, p_l$. Here p_i is a guess for the point closest to the center of a large cluster C_i^*.
Initialize $Q = \{\{p_1\},\ \{p_2\},\ \ldots,\ \{p_l\}\}$.
for $s = n$ to 1 **do**
    Remove any point x such that $d(x, Q) < \frac{OPT}{2s}$. (Again, $d(x, Q) = min_{T \in Q; y \in T} d(x, y)$.)
    Among the remaining points, form a graph as follows: connect (a, b) if $d(a, b) \leq \frac{OPT}{2s}$.
    Add to Q all connected components in this graph of size $> \frac{s}{2}$.
end for
for every choice of k components in Q **do**
    For each component C choose as center the point p which minimizes $\sum_{x \in C} d(x, p)$.
    Compute the k-median cost of the resulting clustering.
    Over all guesses of N points, and all choices of k components in Q, output the clustering that minimized the k-median cost.
end for

B.1 Proof of Correctness

Fact 15 *For any two target clusters C_i^* and C_j^*, and any two points $x \in C_i^*$ and $y \in C_j^*$, it holds that* $d(x, y) > \frac{\mathsf{OPT}}{2\min(|C_i^*|, |C_j^*|)}$.

Proof: For convenience of notation, let x be a point in C_1^*, and y a point in C_2^*. Wlog, assume $|C_1^*| \leq |C_2^*|$. For now, assume y isn't c_2^*. We now shift c_1^* along the edge $\langle c_1^*, y\rangle$ to a point c_1', where $d(c_1', y) = d(c_2^*, y)$.[5] Now, we claim that the Voronoi-based clustering we get by using c_1' instead of c_1^* differs from the optimal clustering. Indeed, in this clustering, y is indifferent between c_1' and c_2^*. Therefore, push c_1' an infinitesimally small distance towards y, so that y will prefer c_1' over c_2^*.

First, observe that among all points in the metric space, c_2^* is the point p that minimizes the quantity $\sum_{z \in C_2^*} d(z, p)$. In particular, $\sum_{z \in C_2^*} d(z, c_2^*) \leq \sum_{z \in C_2^*} d(z, c_1')$. Since y prefers c_1' to c_2^*, it follows that for some point $z_2 \in C_2^*$, it must hold that $d(z_2, c_2^*) < d(z_2, c_1')$. Therefore, z_2 prefer c_2^*, yet y prefers c_1', and this clustering must be different from the optimal clustering. By the 2-stability property, we have that this

[5] Observe, even if c_1^* isn't a data point, we can still shift c_1^* along some path towards y until it is sufficiently close to y.

clustering's cost is at least $2\mathsf{OPT}$. Let $\lambda = d(c_1^*, c_1')$. Clearly, $\lambda = d(c_1^*, y) - d(c_2^*, y) \leq d(c_1^*, x) + d(x, y) - d(c_2^*, y)$. As x prefers c_1^* over c_2^* we have that $\lambda \leq d(x, y) + d(c_1^*, x) - d(c_2^*, y) \leq d(x, y) + d(c_2^*, x) - d(c_2^*, y) \leq 2d(x, y)$. This allows us to upper bound the cost of the Voronoi-based clustering produced by $\{c_1', c_2^*, \ldots, c_k^*\}$ with $\mathsf{OPT} + \sum_{z \in C_1^*} d(c_1^*, c_1') \leq \mathsf{OPT} + 2|C_1^*|d(x, y)$. We infer that $d(x, y) > \frac{\mathsf{OPT}}{2|C_1^*|}$.

If $y = c_2^*$, then consider the Voronoi-based clustering formed by replacing c_1^* with an arbitrary point $z \in C_2^*$.[6] Here, we assign y and z to two different clusters, so again, our cost increases by more than OPT. On the other hand, we can upper bound the cost of this solution by $\mathsf{OPT} + |C_1^*|d(c_1^*, c_2^*)$ so we deduce $d(c_1^*, c_2^*) > \frac{\mathsf{OPT}}{|C_i^*|}$. Now, observe that $d(c_1^*, c_2^*) \leq d(c_1^*, x) + d(x, c_2^*) \leq 2d(x, c_2^*)$ as x prefers c_1^* to c_2^*. We deduce that even when $y = c_2^*$ it holds that $d(x, y) = d(x, c_2^*) > \frac{\mathsf{OPT}}{2|C_1^*|}$. ■

As before, for every small cluster C_i^*, we call the set $\{x : d(x, c_i^*) \leq \frac{\mathsf{OPT}}{8|C_i^*|}\}$ as the *inner ring* of C_i^*. As before, the inner ring of small clusters contain strictly more than half of the cluster. In fact, the inner ring contains more than $\frac{7}{8}|C_i^*|$, but for the majority of our argument, the crude bound of $1/2$ will do. Call a component *good* if it is contained within some target cluster C_i^* *and* it contains all of the inner ring points of C_i^*. We now show that each small target cluster will have a single, unique, good component.

Lemma 16 *For each small cluster C_i^*, there is exactly one* good *component in Q.*

Proof: Let T be any component in Q, and let s be the stage in which T was inserted. We claim that for any smaller cluster C_i^*, s.t. $|C_i^*| \leq s$, we either have $T \subset C_i^*$ or $T \cap C_i^* = \emptyset$. To see this, suppose that T contains a point $x \in C_i^*$ and a point $y \notin c_i^*$. From Fact 15 it follows that $d(x, y) > \frac{\mathsf{OPT}}{2|C_i^*|} \geq \frac{\mathsf{OPT}}{2s}$. Therefore, we cannot directly connect x and y. However, T is a connected component, so exists a path $x \to y$. Along this path there must exist a pair x', y' of two consecutive points s.t. $x' \in C_i^*$ and $y' \notin C_i^*$. The existence of these two points contradicts Fact 15.

We now follow the same reasoning from Claim 10. If by stage $s = |C_i^*|$ we haven't removed any point from the inner ring, then at this stage we form a connected component containing the entire inner ring. So, by stage $s = |C_i^*|$ we either insert a component containing an inner ring from C_i^* to Q or we remove some inner ring point of C_i^*. By contradiction, assume only the latter happens. That means we have a component $T' \in Q$, inserted at stage $s' \geq |C_i^*|$, and a pair of points, $y \in T$, x in the inner ring of C_i^*, s.t. $d(x, y) \leq \frac{\mathsf{OPT}}{2s} \leq \frac{\mathsf{OPT}}{2|C_i^*|}$. Since $x \in C_i^*$, we deduce from Fact 15, that $y \in C_i^*$. Therefore, $T \subset C_i^*$. However, $|T| > s/2 \geq |C_i^*|/2$, yet contains no inner ring point from C_i^*. Contradiction.

So denote T as the first component in Q to contain an inner ring point from C_i^*, and denote that inner ring point as p. As T is inserted to Q before stage $s = |C_i^*|$ we have that T is contained in C_i^*. We deduce that $s < 2|C_i^*|$, as $|C_i^*| \geq |T| > s/2$. Observe that the distance between any two inner ring points is at most $2\frac{\mathsf{OPT}}{8|C_i^*|} < \frac{\mathsf{OPT}}{2s}$, so when we insert T, we have that all inner ring points are connected to p, so the entire inner ring is added to T. We deduce that T is a good component and that it is the only good component containing inner ring points from C_i^*. ■

Lemma 17 *We do not add to Q more than 8 bad (non-good) components.*

Proof: Consider any bad component T that we add to Q and denote that stage in which we insert T to Q as s. So the size of this component is $> \frac{s}{2}$. Let y be an arbitrary point from T, and let c^* be its center in the target clustering. We show that $d(c^*, y) > \frac{\mathsf{OPT}}{4s}$.

Suppose C^* is a large cluster. Assume that our initial guess is correct, and we start with Q containing p^* already, where $p^* = \arg\min_{p \in C^*} d(p, c^*)$. Recall that we remove any point that is too close to the points in Q, so $\frac{\mathsf{OPT}}{2s} < d(y, p^*)$. Therefore, $\frac{\mathsf{OPT}}{2s} < d(y, c^*) + d(c^*, p^*) \leq 2d(y, c^*)$ and the required follows.

Suppose C^* is a small cluster. We divide into cases. If $s \geq |C^*|$, we apply the same observation from Lemma 16, and deduce that T is contained within C^*. However, T only contains points from outside the inner ring of C^* so $T < |C_i^*|/2 \leq s/2$. Contradiction. If $s < |C^*|$, we have that the entire inner ring of C_i^* already belongs to some $T' \in Q$. Let $x \in T'$ be any inner ring point from C^*, and we have that $d(c^*, y) \geq d(x, y) - d(x, c^*) > \frac{\mathsf{OPT}}{2s} - \frac{\mathsf{OPT}}{8|C_i^*|} > \frac{\mathsf{OPT}}{4s}$.

It follows that in either case, the cost of this component, that contains $s/2$ points, is at least $\mathsf{OPT}/8$. Hence, we can have no more than 8 such bad components. ■

[6] In the extreme case where $|C_2^*| = 1$, just replace $c_2^* = y$ with w, the nearest point to y. Clearly y and w now belong to the same cluster, so the increase in cost is $> \mathsf{OPT}$, yet it is at most $d(y, w)$.

Theorem 18 *If our given instance satisfies the 2-property of Voronoi-based approximations, then the algorithm presented in Section B outputs the optimal clustering.*

Proof: Suppose our initial guess is correct, and for the l large clusters ($l \leq 63$), we picked the point $p_i^* \in C_i^*$ which is the closest point to c_i^*. Also assume we picked the right k good components in Q. We show that in this case, we must output a clustering whose cost is $\leq 2\mathsf{OPT}$. This, combined with the 2-stability property, gives that we output the target clustering.

Denote the k-points we use as centers as $p_1^*, p_2^*, \ldots, p_l^*, p_{l+1}, p_{l+2}, \ldots, p_k$, where the first l points belong to the large clusters, and the latter $k-l$ belong to the small clusters. For every point x, let $p(x)$ be the center x is assigned to. Note, in the target clustering, all points from C_i^* are assigned to c_i^*. Here, we do not know that all $x \in C_i^*$ were in fact assigned to the same p_i. However, we show that $\sum_{x \in C_i^*} d(x, p(x)) \leq 2\mathsf{OPT}_i$, thus proving our claim.

First consider the case where C_i^* is a large cluster. For every $x \in C_i^*$ it holds that $d(x, p(x)) \leq d(x, p_i^*) \leq d(x, c_i^*) + d(c_i^*, p_i^*)$. As p_i^* is the closest point to the center, it follows that $d(x, p_i^*) \leq 2d(x, c_i^*)$, so $\sum_{x \in C_i^*} d(x, p(x)) \leq 2d(x, c_i^*) = 2\mathsf{OPT}_i$.

Now consider a small cluster, C_i^*. Recall, we pick p_i to be a prospective center because p_i minimized the quantity $\sum_{x \in T} d(x, p_i)$. Let p_i^* be the point in C_i^* which is the closest to c_i^*. Recall that C_i^* is small, so p_i^* must belong to the inner ring of C_i^* and therefore must belong to T. Imagine a perfect scenario, where $p_i = p_i^*$. Just as shown above, this would imply that $\sum_{x \in C_i^*} d(x, p(x)) \leq \sum_{x \in C_i^*} d(x, p_i^*) \leq 2 \sum_{x \in C_i^*} d(x, c_i^*) = 2\mathsf{OPT}_i$, and we would be done. However, we cannot guarantee that p_i is indeed the closest point to c_i^*. Instead, we break the summation $\sum_{x \in C_i^*} d(x, p_i)$ into two parts: $\sum_{x \in T} d(x, p_i) + \sum_{x \in C_i^* \setminus T} d(x, p_i)$.

The first term, $\sum_{x \in T} d(x, p_i)$, is the very quantity we minimized using p_i. We already know p_i^* belongs to T, so $\sum_{x \in T} d(x, p_i) \leq \sum_{x \in T} d(x, p_i^*) \leq 2 \sum_{x \in T} d(x, c_i^*)$. So we now focus on the 2nd term. Since T contains all the inner ring points, it follows that $C_i^* \setminus T$ contains only points x s.t. $d(c_i^*, x) > \frac{\mathsf{OPT}}{8|C_i^*|}$. We claim that p_i belongs to the inner ring of C_i^*. Given this claim, we are done, since we have that

$$\sum_{x \in C_i^* \setminus T} d(x, p(x)) \leq \sum_{x \in C_i^* \setminus T} d(x, p_i) \leq \sum_{x \in C_i^* \setminus T} d(x, c_i^*) + d(c_i^*, p_i) = \sum_{x \in C_i^* \setminus T} d(x, c_i^*) + \frac{\mathsf{OPT}}{8|C_i^*|} \leq 2 \sum_{x \in C_i^* \setminus T} d(x, c_i^*)$$

To see that indeed p_i is a point from the inner ring of C_i^*, we utilize the fact that a small cluster's cost is $\leq \frac{\mathsf{OPT}}{64}$. By Markov's inequality, we have that the set $S = \left\{ x \in C_i^*;\ d(c_i^*, x) \leq \frac{\mathsf{OPT}}{32|C_i^*|} \right\}$ contains at least $|C_i^*|/2$ many points. Suppose by contradiction that p_i does not belong to the inner ring, and we have that $d(c_i^*, p_i) > \frac{\mathsf{OPT}}{8|C_i^*|}$. Thus, for every $x \in S$ we have that $d(x, p_i) \geq \frac{\mathsf{OPT}}{8|C_i^*|} - \frac{\mathsf{OPT}}{32|C_i^*|} > \frac{\mathsf{OPT}}{16|C_i^*|}$. In such a case, S alone will contribute to the sum $\sum_{x \in T} d(x, p_i)$ more than $\frac{|C_i^*|}{2} \cdot \frac{\mathsf{OPT}}{16|C_i^*|} = \frac{\mathsf{OPT}}{32} > 2\mathsf{OPT}_i$. Recall, if we set $p_i = p_i^*$, we have that this sum is at most $2 \sum_{x \in T} d(x, c_i^*) \leq 2\mathsf{OPT}_i$. This contradicts the fact that p_i minimizes the above sum. ■

B.2 Analysis

For any given guess of l and $p_1, p_2, \ldots, p_l$, the algorithm runs in time $O(n^3 + n^2 k^8)$. Hence, the total running time is $O(n^{67} + n^{66} k^8)$.

We would like to note, that in contrast to the finite metric, here we have no trivial extension of the problem to a $(1+\alpha)$-stability property of Voronoi-based approximations. Furthermore, getting the 1.999-property, or any approximation ratio below 2, seems inherently difficult. The reason why we need the 2 factor approximation is that by shifting the center of a cluster to its closest data-point, we might double the cost of the cluster.