Clustering under Approximation Stability

MARIA-FLORINA BALCAN
School of Computer Science, Georgia Institute of Technology
AVRIM BLUM
Computer Science Department, Carnegie Mellon University
ANUPAM GUPTA
Computer Science Department, Carnegie Mellon University

A common approach to clustering data is to view data objects as points in a metric space, and then to optimize a natural distance-based objective such as the k-median, k-means, or min-sum score. For applications such as clustering proteins by function or clustering images by subject, the implicit hope in taking this approach is that the optimal solution to the chosen objective will closely match the desired "target" clustering (e.g., a correct clustering of proteins by function or of images by who is in them). However, most distance-based objectives, including those above, are NP-hard to optimize. So, this assumption by itself is not sufficient, assuming P $\neq$ NP, to achieve clusterings of low-error via polynomial time algorithms.

In this paper, we show that we can bypass this barrier if we slightly extend this assumption to ask that for some small constant c, not only the optimal solution, but also all c-approximations to the optimal solution, differ from the target on at most some ϵ fraction of points—we call this (c, ϵ)-*approximation-stability*. We show that under this condition, it is possible to efficiently obtain low-error clusterings even if the property holds only for values c for which the objective is known to be NP-hard to approximate. Specifically, for any constant $c > 1$, (c, ϵ)-approximation-stability of k-median or k-means objectives can be used to efficiently produce a clustering of error $O(\epsilon)$, as can stability of the min-sum objective if the target clusters are sufficiently large. Thus, we can perform nearly as well in terms of agreement with the target clustering *as if* we could approximate these objectives to this NP-hard value.

Categories and Subject Descriptors: F.2.0 [**Analysis of Algorithms and Problem Complexity**]: General; I.5.3 [**Pattern Recognition**]: Clustering

General Terms: Algorithms, Theory.

Additional Key Words and Phrases: Clustering, Approximation Algorithms, k-Median, k-Means, Min-Sum, Clustering Accuracy.

A preliminary version of these results appears under the title "Approximate Clustering without the Approximation" in the ACM-SIAM Symposium on Discrete Algorithms, 2009.

Authors' addresses: M.-F. Balcan, School of Computer Science, Georgia Institute of Technology, Atlanta, GA 30332, email: ninamf@cc.gatech.edu; A. Blum, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213-3891, email: avrim@cs.cmu.edu; A. Gupta, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213-3891, email: anupamg@cs.cmu.edu.

1. INTRODUCTION

Overview. Problems of clustering data are ubiquitous throughout science. They arise in many different fields, from computational biology where such problems include clustering protein sequences by function, to computer vision where one may want to cluster images by subject, to information retrieval, including problems of clustering documents or search results by topic, just to name a few.

A commonly used approach for data clustering is *objective-based clustering* where we first represent data as points in a metric space and then choose a particular distance-based objective function (e.g., k-median or k-means) to optimize. However, in many of the clustering applications mentioned above there is an unknown desired or *target* clustering, and while the distance information among data points is merely heuristically defined, the real goal in these applications is to minimize the clustering error with respect to the target (e.g., the true clustering of protein sequences by function). From a modeling point of view, the implicit hope (or inductive bias) in using objective-based clustering to solve such problems is that the optimal solution to the chosen objective is close to or has small error with respect to the target clustering. Unfortunately, however, most distance based objectives are NP-hard to optimize, so this assumption by itself is not sufficient (assuming P $\neq$ NP) to achieve clusterings of low-error via polynomial time algorithms. In this paper we argue that a better inductive bias is to assume that not only the optimal solution, but also all approximately optimal solutions to the distance based objective in fact have low error to the target – we call this *approximation-stability.* We analyze the implications of the approximation-stability assumption in the context of three well studied distance-based objectives and provide algorithms for finding low-error clusterings. Surprisingly, we show it is possible to obtain low-error clusterings even if approximation-stability holds only for approximation factors that are known to be NP-hard to achieve!

Problem setup and results. We assume that data is represented as a set S of points in some metric space and consider three commonly studied objective functions: k-median, k-means, and min-sum. In the k*-median* problem, the goal is to partition S into k clusters C_i, assigning each a center c_i, to minimize the sum of the distances between each datapoint and the center of its cluster. In the k-means problem, the goal is to partition S into k clusters C_i, assigning each a center c_i, to minimize the sum of *squares* of distances between each datapoint and the center of its cluster. In min-sum clustering, the goal is to partition S into k clusters C_i that minimize the sum of all intra-cluster pairwise distances. These objectives are all NP-hard to optimize exactly especially when k is large and not a constant [Sahni and Gonzalez 1976; Megiddo and Supowit 1984; Mahajan et al. 2009]. As a result, from a theoretical point of view, substantial research has focused on the design of *approximation algorithms*: algorithms guaranteed to produce a solution at most some factor c larger than the optimum. For the k-median and k-means problems the best approximation guarantees known are constant factors of 3 or larger [Arya et al. 2004; Kanungo et al. 2004] and for min-sum the best known result for general metric spaces is an $O(\log^{1+\delta} n)$-approximation [Bartal et al. 2001]; all these approximation guarantees do not match the known hardness results, and much effort is spent on obtaining tighter approximation ratios (see Related Work).

In this work we particularly focus on clustering instances whose solutions are stable to small approximation factors. As argued earlier, this is especially relevant for problems where there is some unknown correct "target" clustering (e.g., a correct clustering of proteins by their function or a correct clustering of images by who is in them) and the underlying goal is not to minimize some function of the distances, but rather to match this target as closely as possible (see Related Work for further discussion). Formally, the key property we introduce and study is (c, ϵ)*-approximation-stability*; a clustering instance (i.e., a data set) satisfies (c, ϵ)-approximation-stability with respect to a given objective Φ (such as k-median, k-means, or min-sum) and an unknown target clustering $\mathcal{C}_T$, if it has the property that every c-approximation to Φ is ϵ-close to $\mathcal{C}_T$ in terms of the fraction of misclustered points. That is, for any c-approximation to Φ, at most an ϵ fraction of points would have to be reassigned in that clustering to make it perfectly match $\mathcal{C}_T$.

Clearly, if the (c, ϵ)-approximation-stability condition is true for a value of c such that a c-approximation is known for these problems, then we could simply use the associated approximation algorithm. As mentioned above, however, existing results for approximating the three objectives we consider would require fairly large constant values of c, or even $c = \omega(\log n)$ in the case of the min-sum objective. What we show in this paper is that we can do much better. In particular, we show that we can efficiently produce a clustering that is $O(\epsilon)$-close to the target *even if stability holds only for values c for which obtaining a c-approximation is provably NP-hard*. Specifically, we achieve this guarantee for *any constant* $c > 1$ for the k-median and k-means objectives, as well as for any constant $c > 1$ for the min-sum objective when the target clusters are sufficiently large compared to $\frac{\epsilon n}{c-1}$. Moreover, if the target clusters are sufficiently large compared to $\frac{\epsilon n}{c-1}$, for k-median we can actually get ϵ-close (rather than $O(\epsilon)$-close) to the target.[1] Furthermore, we achieve these guarantees without necessarily approximating the associated objective; in fact, we show that achieving a c-approximation for instances satisfying (c, ϵ)-approximation-stability is as hard as achieving a c-approximation in general. Note that one should view k and ϵ here as parameters and not as constants: our algorithms will run in time polynomial in the number of points n and the number of clusters k. Indeed, in many of the types of scenarios motivating this work such as clustering protein sequences by function or clustering images by who is in them, k can be quite large.

Our algorithms operate by carefully exploiting the structure of the (c, ϵ)-approximation-stability condition. We begin by showing that approximation-stability, even for a constant c such as 1.01, implies that most data points are well-behaved with respect to the optimal solution for the given objective, which itself (by assumption) is close to the target. Specifically, all but an $O(\epsilon)$ fraction of data points will be a constant factor closer to their own center of the optimal solution than to their second-closest center (for min-sum, the condition is a bit different). We then use this property to design algorithms that are able to correctly cluster these "good" points without being misled by the (unknown) subset of "bad" points. This in turn requires algorithms that are especially robust to outliers. Since a $1 - O(\epsilon)$

[1] Results in this paper for both k-means and min-sum objectives are strengthened over those in the conference version of this work [Balcan et al. 2009].

fraction of points are good, this then leads to our $O(\epsilon)$ error guarantees. Finally, in the case that the target clusters are large, for the k-median objective we are able to recover nearly all the bad points through a second re-clustering step that can be viewed as an outlier-resistant form of a 2-stage Lloyd's k-means algorithm. This allows us to produce a clustering of error at most ϵ: exactly as low as if we had a generic c-approximation algorithm.

Overall, our results bring together approximation (of distance-based objectives) and accuracy (of clusterings), and show that one can achieve polynomial-time accuracy guarantees under substantially weaker assumptions than would seem to be required given the worst-case approximability of these objectives. Our results also show that there is an interesting computational difference between assuming that the *optimal* solution to, say, the k-median objective is ϵ-close to the target, and assuming that any *approximately optimal* solution is ϵ-close to the target, even for approximation factor $c = 1.01$ (say). In the former case, the problem of finding a solution that is $O(\epsilon)$-close to the target remains computationally hard (see Section 2.4 and Theorem A.3), and yet for the latter case we give efficient algorithms.

From a broader theoretical perspective, our results can be viewed in the context of work such as Ostrovsky et al. [2006] (see also Sections 1.1 and 7.4) showing that one can bypass worst-case hardness barriers if one makes certain natural stability assumptions on the data. In our case, the stability assumptions we consider are especially motivated by the relation between objective functions and accuracy: in particular, they allow us to conclude that we can perform nearly as well in terms of accuracy *as if* we had a generic PTAS for the associated objective.

Subsequent work has also demonstrated the practicality of our approach for real world clustering problems. For example, Voevodski et al. [2010; 2012] show that a variant of the algorithm we propose for the k-median problem provides state of the art results for clustering biological datasets. Our work has inspired a number of other subsequent developments and we discuss these further in Section 7.

1.1 Related Work

Work on recovering a target clustering. Accuracy in matching a target clustering is commonly used to compare different clustering algorithms experimentally, e.g., [Manning et al. 2008; Yan et al. 2009; Brohée and van Helden 2006].

In theoretical work, much of the research on analyzing clustering accuracy has been in the context of clustering or learning with mixture models [Achlioptas and McSherry 2005; Arora and Kannan 2005; Duda et al. 2001; Devroye et al. 1996; Kannan et al. 2005; Vempala and Wang 2004; Dasgupta 1999; Belkin and Sinha 2010; Moitra and Valiant 2010]. That work, like ours, has an explicit notion of a correct ground-truth clustering; however, it makes strong probabilistic assumptions about how data points are generated.

Balcan et al. [2008] investigate the goal of approximating a desired target clustering without probabilistic assumptions. They analyze what properties of a pairwise similarity function allow one to produce a tree such that the target is close to some pruning of the tree, or a small list of clusterings such that the target is close to at least one clustering in the list. Regarding assumptions related to approximate optimization, they show that for k-median, the assumption that any 2-approximation is ϵ-close to the target can be used to construct a hierarchical clustering such that the target clustering is close to some pruning of the hierarchy. Inspired by their approach, in this paper we initiate a systematic investigation of the consequences of such assumptions in the context of commonly-used distance-based objective functions as well as their connections to approximation algorithms. Moreover, the goals in this paper are stronger — we want to output a *single* approximately correct clustering (as opposed to a list of clusterings or a hierarchy), and we want to succeed for *any* $c > 1$.

Work on approximation algorithms. The study of approximation algorithms for distance-based clustering objectives such as k-median, k-means, and min-sum is a very active area, with a large number of algorithmic results.

For *k-median*, $O(1)$-approximations were first given by Charikar et al. [1999], Jain and Vazirani [2001], and Charikar and Guha [1999], and the best approximation guarantee known is $(3 + \epsilon)$ due to Arya et al. [2004]. A straightforward reduction from max-k-coverage shows $(1+1/\mathrm{e})$-hardness of approximation [Guha and Khuller 1999; Jain et al. 2002]. The k-median problem on constant-dimensional Euclidean spaces admits a PTAS [Arora et al. 1999].

For *k-means* in general metric spaces a constant-factor approximation is known [Kanungo et al. 2004], and an approximation-hardness of $1 + 3/e$ follows from the ideas of [Guha and Khuller 1999; Jain et al. 2002]. This problem is very often studied in Euclidean space, where a near-linear time $(1 + \epsilon)$-approximation algorithm is known for the case of *constant* k and ϵ [Kumar et al. 2004]. Arthur and Vassilvitskii [2007] show a fast randomized seeding approach gives an $O(\log k)$ approximation for general values of k, which they then use as a starting point for Lloyd's local search algorithm [Lloyd 1982]. An interesting extension of the k-means objective to clusters lying in different subspaces is given in [Agarwal and Mustafa 2004].

Min-sum k-clustering in general metric spaces admits a PTAS for the case of constant k [de la Vega et al. 2003] (see also Indyk [1999]). For the case of arbitrary k there is an $O(\delta^{-1} \log^{1+\delta} n)$-approximation algorithm running in time $n^{O(1/\delta)}$ due to Bartal et al. [2001]. The problem has also been studied in geometric spaces for constant k by Schulman [2000] who gave an algorithm for (R^d, ℓ_2^2) that either outputs a $(1+\epsilon)$-approximation, or a solution that agrees with the *optimum* clustering on a $(1-\epsilon)$-fraction of the points (but could have much larger cost than optimum); the runtime is $O(n^{\log \log n})$ in the worst case and linear for sublogarithmic dimension d. More recently, Czumaj and Sohler have developed a $(4 + \epsilon)$-approximation algorithm for the case when k is small compared to $\log n / \log \log n$ [Czumaj and Sohler 2007].

Clustering under natural stability conditions. Motivated by the fact that heurstics such as Lloyd's k-means local search algorithm [Lloyd 1982] are often used in practice despite poor worst-case performance, Ostrovsky et al. [2006] analyze clustering under a natural stability condition they call *ϵ-separation.* They show that under this condition, an appropriate seeding of Lloyd's algorithm will result in solutions with provable approximation guarantees. Their ϵ-separation condition has an interesting relation to approximation-stability, which we discuss more fully in Section 6. Essentially, it is a stronger assumption than ours; however, their goal is different—they want to approximate the objective whereas we want to approximate the target clustering.

More recently, Bilu and Linial [2010], Kumar and Kannan [2010], Awasthi et al. [2010], Awasthi et al. [2012], Balcan and Liang [2012], and Awasthi and Sheffet [2012] consider other stability conditions; we discuss these further in Section 7.4.

Other theoretical directions in clustering. There is a large body of work on other theoretical topics in clustering such as defining measures of clusterability of data sets, on formulating definitions of good clusterings [Gollapudi et al. 2006], and on axiomatizing clustering (in the sense of postulating what natural axioms a "good clustering algorithm" should satisfy), both with possibility and impossibility results [Kleinberg 2002]. There has also been significant work on approaches to comparing clusterings [Meila 2003; 2005], and on efficiently testing if a given data set has a clustering satisfying certain properties [Alon et al. 2000]. The main difference between this type of work and our work is that we have an explicit notion of a correct ground-truth clustering of the data points, and indeed the results we are trying to prove are quite different. The work of Meila [2006] is complementary to ours: it shows sufficient conditions under which k-means instances satisfy the property that near-optimal solutions are ϵ-close to the *optimal* k-means solution.

2. DEFINITIONS, PRELIMINARIES & FORMAL STATEMENT OF MAIN RESULTS

The clustering problems in this paper fall into the following general framework: we are given a metric space $\mathcal{M} = (X, d)$ with point set X and a distance function $d : \binom{X}{2} \to R_{\geq 0}$ satisfying the triangle inequality—this is the ambient space. We are also given the actual point set $S \subseteq X$ we want to cluster; we use n to denote the cardinality of S. A k-clustering $\mathcal{C}$ is a partition of S into k sets $C_1, C_2, \ldots, C_k$. In this paper, we always assume that there is an (unknown) *true* or *target* k-clustering $\mathcal{C}_T$ for the point set S.

2.1 The Objective Functions

Commonly used clustering objectives provide a distance-based cost to any given clustering that algorithms then seek to minimize. In all the objectives we consider, the cost of a clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ is a sum of costs on the individual clusters C_i. The *k-median* clustering objective defines the cost of a cluster C_i to be the total distance of all points in C_i to the best "median" point $c_i \in X$ for that cluster; that is,

$$\Phi_1(\mathcal{C}) = \sum_{i=1}^{k} \min_{c_i \in X} \sum_{x \in C_i} d(x, c_i).$$

The *k-means* objective defines the cost of a cluster C_i to be the sum of squared distances to the best center $c_i \in X$ for that cluster:

$$\Phi_2(\mathcal{C}) = \sum_{i=1}^{k} \min_{c_i \in X} \sum_{x \in C_i} d(x, c_i)^2.$$

Finally, the *min-sum* objective defines the cost of a cluster to be the sum of all pairwise intra-cluster distances:

$$\Phi_\Sigma(\mathcal{C}) = \sum_{i=1}^{k} \sum_{x \in C_i} \sum_{y \in C_i} d(x, y).$$

Given a function Φ and instance $(\mathcal{M}, S, k)$, let $\mathrm{OPT}_\Phi = \min_{\mathcal{C}} \Phi(\mathcal{C})$, where the minimum is over all k-clusterings of S. We will typically use $\mathcal{C}^*$ to denote the optimal clustering for the given objective, and will simply write an instance as $(\mathcal{M}, S)$ when k is clear from context.

2.2 Distance between Clusterings

In order to define the notion of approximation-stability, we need to specify what it means for a clustering to be *close* to the target $\mathcal{C}_T$. Formally, we define the distance $dist(\mathcal{C}, \mathcal{C}')$ between two k-clusterings $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ and $\mathcal{C}' = \{C'_1, C'_2, \ldots, C'_k\}$ of a point set S as the fraction of points in S on which they disagree under the optimal matching of clusters in $\mathcal{C}$ to clusters in $\mathcal{C}'$; i.e.,

$$dist(\mathcal{C}, \mathcal{C}') = \min_{\sigma \in \mathfrak{S}_k} \frac{1}{n} \sum_{i=1}^{k} |C_i - C'_{\sigma(i)}|,$$

where $\mathfrak{S}_k$ is the set of bijections $\sigma : [k] \to [k]$. Equivalently, $dist(\mathcal{C}, \mathcal{C}')$ is the number of mistakes, or 0/1-loss, of $\mathcal{C}$ with respect to $\mathcal{C}'$ if we view each as a k-way classifier, under the best matching between their k class labels.[2]

We say that two clusterings $\mathcal{C}$ and $\mathcal{C}'$ are *ϵ-close* if $dist(\mathcal{C}, \mathcal{C}') < \epsilon$. Note that if $\mathcal{C}$ and $\mathcal{C}'$ are ϵ-close and all clusters C_i have size at least $2\epsilon n$, then the bijection σ minimizing $\frac{1}{n}\sum_{i=1}^{k} |C_i - C'_{\sigma(i)}|$ has the property that for all i, $|C_i \cap C'_{\sigma(i)}| \geq |C_i| - (\epsilon n - 1) > \frac{1}{2}|C_i|$. This implies for instance that such σ is unique, in which case we call this the *optimal bijection* and we say that $\mathcal{C}$ and $\mathcal{C}'$ *agree* on x if $x \in C_i \cap C'_{\sigma(i)}$ for some i, and $\mathcal{C}$ and $\mathcal{C}'$ *disagree* on x otherwise.

2.3 (c, ϵ)-approximation-stability

We now present our main definition whose implications we study throughout this paper:

DEFINITION 1 ((c, ϵ)-APPROXIMATION-STABILITY). *Given an objective function Φ (such as k-median, k-means, or min-sum), we say that instance $(\mathcal{M}, S)$* satisfies

[2] There are other reasonable notions of distance between clusterings that one can also consider. For example, one could remove the restriction that σ be a permutation (see Lemma B.1 in Appendix B for analysis of this notion). Alternatively, one could count the fraction of *pairs* x, y such that the clusterings disagree on whether or not x and y belong to the same cluster. Note, however, that if k is large and all clusters are about the same size, then any two clusterings will be fairly close (distance $\leq 2/k$) under the pair-based measure, since most pairs x, y belong to different clusters, so it is not very discriminative. In contrast, under the point-based misclassification measure, two random k-clusterings would have distance approximately $1 - 1/k$. See [Meila 2012] for further discussion of various notions of distance between clusterings and how they relate. We also wish to emphasize that $dist(.,.)$ is a distance between *clusterings*, whereas $d(.,.)$ is a distance between *points*.

(c, ϵ)-approximation-stability for Φ *with respect to (unknown) target clustering* $\mathcal{C}_T$ *if all clusterings* $\mathcal{C}$ *with* $\Phi(\mathcal{C}) \leq c \cdot \mathrm{OPT}_\Phi$ *are* ϵ*-close to* $\mathcal{C}_T$.

As mentioned above, if we have an instance satisfying (c, ϵ)-approximation-stability for c sufficiently large that we have a polynomial-time c-approximation algorithm for Φ, then we could simply use that algorithm to achieve a clustering ϵ-close to $\mathcal{C}_T$. However, our key interest will be in values of c that are significantly smaller. In particular, since we will be thinking of c as being only slightly larger than 1 (e.g., assuming that all 1.1-approximations to the k-median objective are ϵ-close to $\mathcal{C}_T$), we will often write c as $1+\alpha$ and look at the implications in terms of the parameters α and ϵ. Additionally, we will typically drop the phrase "with respect to the (unknown) target clustering $\mathcal{C}_T$" when this is clear from context.

It is important to note that $1/\epsilon$, $1/\alpha$, and k need not be constants. For example, we might have that $\mathcal{C}_T$ consists of $n^{0.1}$ clusters of size $n^{0.9}$, $\epsilon = 1/n^{0.2}$ and $\alpha = 1/n^{0.09}$ (this would correspond to the "large clusters case" of Theorem 3.6).

Note that for any $c > 1$, (c, ϵ)-approximation-stability does not require that the target clustering $\mathcal{C}_T$ exactly coincide with the optimal clustering $\mathcal{C}^*$ under objective Φ. However, it does imply the following simple facts (where part (b) below follows from the fact that the distance between k-clusterings itself is a metric):

Fact 2.1. *If* $(\mathcal{M}, S)$ *satisfies* (c, ϵ)*-approximation-stability for* Φ *with respect to target clustering* $\mathcal{C}_T$*, then:*

(a) The target clustering $\mathcal{C}_T$*, and the optimal clustering* $\mathcal{C}^*$ *for* Φ *are* ϵ*-close.*

(b) $(\mathcal{M}, S)$ *satisfies* $(c, \epsilon + \epsilon^*)$*-approximation-stability for* Φ *with respect to the optimal clustering* $\mathcal{C}^*$*, where* $\epsilon^* = dist(\mathcal{C}^*, \mathcal{C}_T)$.

Thus, we can act as if the optimal clustering is indeed the target up to a constant factor loss in the error rate.

Finally, we will often want to take some clustering $\mathcal{C}$, reassign some $\tilde{\epsilon}n$ points to different clusters to produce a new clustering $\mathcal{C}'$, and then argue that $dist(\mathcal{C}, \mathcal{C}') = \tilde{\epsilon}$. As mentioned above, if all clusters of $\mathcal{C}$ have size at least $2\tilde{\epsilon}n$, then it is clear that no matter how $\tilde{\epsilon}n$ points are reassigned, the optimal bijection σ between the original clusters and the new clusters is the identity mapping, and therefore $dist(\mathcal{C}, \mathcal{C}') = \tilde{\epsilon}$. However, this need not be so when small clusters are present: for instance, if we reassign all points in C_i to C_j and all points in C_j to C_i then $dist(\mathcal{C}, \mathcal{C}') = 0$. Instead, in this case we will use the following convenient lemma.

Lemma 2.2. *Let* $\mathcal{C} = \{C_1, \ldots, C_k\}$ *be a* k*-clustering in which each cluster is nonempty, and let* $R = \{(x_1, j_1), (x_2, j_2), \ldots, (x_t, j_t)\}$ *be a set of* t *reassignments of points* x_i *to clusters* C_{j_i} *(assume that* $x_i \notin C_{j_i}$ *for all* i*). Then there must exist a set* $R' \subseteq R$ *of size at least* $t/3$ *such that the clustering* $\mathcal{C}'$ *produced by reassigning points in* R' *has distance exactly* $\frac{1}{n}|R'|$ *from* $\mathcal{C}$.

Proof. See Appendix A.1. □

2.4 Intuition and Challenges

Before proceeding to our results, we first present some challenges in using the (c, ϵ)-approximation-stability condition to achieve low-error clusterings, which also provide intuition into what this condition does and does not imply.

First, suppose that (c, ϵ)-approximation-stability for some objective Φ implied, say, $(2c, 2\epsilon)$-approximation-stability. Then it would be sufficient to simply apply an $O(c)$ approximation in order to have error $O(\epsilon)$ with respect to the target. However, it turns out that for any $c_1 < c_2$ and any $\epsilon > 0$, for each of the three objectives we consider (k-median, k-means, and min-sum), there exists a family of metric spaces and target clusterings that are (c_1, ϵ)-approximation-stable for that objective, and yet have a c_2-approximation with error 49% with respect to the target (see Appendix, Theorem A.1). Thus, a direct application of an arbitrary c_2-approximation would not achieve our goals.[3]

Second, one might hope that (c, ϵ)-approximation-stability would imply structure that allows one to more easily find a c-approximation. However, this is not the case either: for any $c > 1$ and $\epsilon > 0$, the problem of finding a c-approximation to any of the three objectives we consider under (c, ϵ)-approximation-stability is as hard as finding a c-approximation in general (Theorem A.2). Thus, we want to aim directly towards achieving low error rather than necessarily aiming to get a good approximation to the objective. Note that this reduction requires small clusters. Indeed, as pointed out by [Schalekamp et al. 2010], our k-median algorithm for the large-clusters case *is*, as a byproduct, a c-approximation.

It is also interesting to note that results of the form we are aiming for are *not possible given only* $(1, \epsilon)$*-approximation-stability.* Indeed, because the standard hardness-of-approximation proof for k-median produces a metric in which all pairwise distances lie in a bounded range, the proof also implies that it is NP-hard, given a data set with only the guarantee that the *optimal* solution is ϵ-close to the target, to find a clustering of error $O(\epsilon)$; see Theorem A.3.

2.5 Main results and organization of this paper

We present our analysis of the k-median objective in Section 3, the k-means objective in Section 4, and the min-sum objective in Section 5. Our main results for each of these objectives are as follows. (Theorems are numbered according to their location in the main body.)

THEOREM 3.6 (k-Median, Large Clusters Case) *There is an efficient algorithm that if the given instance* $(\mathcal{M}, S)$ *satisfies* $(1 + \alpha, \epsilon)$*-approximation-stability for the* k*-median objective, and each cluster in* $\mathcal{C}_T$ *has size at least* $(4 + 15/\alpha)\epsilon n + 2$*, will produce a clustering that is* ϵ*-close to* $\mathcal{C}_T$*.*

THEOREM 3.7 (k-Median: General Case) *There is an efficient algorithm that if the given instance* $(\mathcal{M}, S)$ *satisfies* $(1 + \alpha, \epsilon)$*-approximation-stability for the* k*-median objective, will produce a clustering that is* $O(\epsilon + \epsilon/\alpha)$*-close to* $\mathcal{C}_T$*.*

THEOREM 4.3 (k-Means: General Case) *There is an efficient algorithm that if*

[3] Balcan and Braverman [2009] show that interestingly a relationship of this form *does* hold for the *correlation-clustering* problem.

the given instance $(\mathcal{M}, S)$ satisfies $(1+\alpha, \epsilon)$-approximation-stability for the k-means objective, will produce a clustering that is $O(\epsilon + \epsilon/\alpha)$-close to $\mathcal{C}_T$.

THEOREM 5.4 (Min-sum: Large Clusters Case) *There is an efficient algorithm that if the given instance $(\mathcal{M}, S)$ satisfies $(1+\alpha, \epsilon)$-approximation-stability for the min-sum objective and each cluster in $\mathcal{C}_T$ has size greater than $(6+120/\alpha)\epsilon n$, will produce a clustering that is $O(\epsilon + \epsilon/\alpha)$-close to $\mathcal{C}_T$.*

We emphasize that our algorithms run in time polynomial in n and k with no dependence on α and ϵ; in particular, $1/\alpha$ and $1/\epsilon$ (and k) need not be constants.

For the "large-cluster" case of k-means, we also have a weaker version of Theorem 3.6, where we mark some $O(\epsilon n/\alpha)$ points as "don't know" and cluster the rest with error at most ϵ. That is, while the total error in this case may be more than ϵ, we can explicitly point out all but ϵn of the points we may err on (see Theorem 4.4). As noted earlier, we only give results for the large-cluster case of min-sum clustering, though [Balcan and Braverman 2009] have recently extended Theorem 5.4 to the case of general cluster sizes; in particular, they achieve the analog of Theorem 3.7 if a (good approximation to) the optimum objective value is provided to the algorithm, else a list of at most $O(\log\log n)$ clusterings such that at least one is $O(\epsilon + \epsilon/\alpha)$-close to $\mathcal{C}_T$ if such a value is not provided.

3. THE K-MEDIAN PROBLEM

We now study clustering instances satisfying (c, ϵ)-approximation-stability for the k-median objective. Our main results are that for any constant $c > 1$, (1) if all clusters are "large", then this property allows us to efficiently find a clustering that is ϵ-close to the target clustering, and (2) for *any* cluster sizes, we can efficiently find a clustering that is $O(\epsilon)$-close to the target. To prove these results, we first investigate the implications of (c, ϵ)-approximation-stability in Section 3.1. We then give our algorithm for the case that all clusters are large in Section 3.2, and our algorithm for arbitrary cluster sizes in Section 3.3.

3.1 Implications of (c, ϵ)-approximation-stability

Given a clustering instance specified by a metric space $\mathcal{M} = (X, d)$ and a set of points $S \subseteq X$, fix an optimal k-median clustering $\mathcal{C}^* = \{C_1^*, \ldots, C_k^*\}$, and let c_i^* be the center point (a.k.a. "median") for C_i^*. For $x \in S$, define

$$w(x) = \min_i d(x, c_i^*)$$

to be the contribution of x to the k-median objective in $\mathcal{C}^*$ (i.e., x's "weight"), and let $w_2(x)$ be x's distance to the second-closest center point among $\{c_1^*, c_2^*, \ldots, c_k^*\}$. Also, define

$$w_{avg} = \frac{1}{n}\sum_{i=1}^{n} w(x) = \frac{\mathrm{OPT}}{n}$$

to be the average weight of the points. Finally, let $\epsilon^* = dist(\mathcal{C}_T, \mathcal{C}^*)$. As noted in Fact 2.1, approximation-stability implies $\epsilon^* < \epsilon$.

LEMMA 3.1. *If the instance $(\mathcal{M}, S)$ satisfies $(1+\alpha, \epsilon)$-approximation-stability for the k-median objective, then*

(a) If each cluster in $\mathcal{C}_T$ has size at least $2\epsilon n$, then less than $(\epsilon - \epsilon^*)n$ points $x \in S$ on which $\mathcal{C}_T$ and $\mathcal{C}^*$ agree have $w_2(x) - w(x) < \frac{\alpha w_{avg}}{\epsilon}$.

(a') For general cluster sizes in $\mathcal{C}_T$, less than $6\epsilon n$ points $x \in S$ have $w_2(x) - w(x) < \frac{\alpha w_{avg}}{2\epsilon}$.

Also, for any $t > 0$ we have:

(b) At most $t(\epsilon n/\alpha)$ points $x \in S$ have $w(x) \geq \frac{\alpha w_{avg}}{t\epsilon}$.

PROOF. To prove Property (a), assume to the contrary. Then one could take $\mathcal{C}^*$ and move $(\epsilon - \epsilon^*)n$ points x on which $\mathcal{C}_T$ and $\mathcal{C}^*$ agree to their second-closest clusters, increasing the objective by at most αOPT. Moreover, this new clustering $\mathcal{C}' = \{C'_1, \ldots, C'_k\}$ has distance at least ϵ from $\mathcal{C}_T$, because we begin at distance ϵ^* from $\mathcal{C}_T$ and each move increases this distance by $\frac{1}{n}$ (here we use the fact that because each cluster in $\mathcal{C}_T$ has size at least $2\epsilon n$, the optimal bijection between $\mathcal{C}_T$ and $\mathcal{C}'$ remains the same as the optimal bijection between $\mathcal{C}_T$ and $\mathcal{C}^*$). Hence we have a clustering that is not ϵ-close to $\mathcal{C}_T$ with cost only $(1+\alpha)$OPT, a contradiction.

For Property (a'), we use Lemma 2.2. Specifically, assuming for contradiction that $6\epsilon n$ points satisfy (a'), Lemma 2.2 states that we can find a subset of $2\epsilon n$ of them such that starting from $\mathcal{C}^*$, for each one that we move to its second-closest cluster, the distance from $\mathcal{C}^*$ increases by $\frac{1}{n}$. Therefore, we can create a clustering $\mathcal{C}'$ that is distance at least 2ϵ from $\mathcal{C}^*$ while increasing the objective by at most αOPT; by Fact 2.1(b) this clustering $\mathcal{C}'$ is not ϵ-close to $\mathcal{C}_T$, thus contradicting $(1+\alpha, \epsilon)$-approximation-stability. Property (b) simply follows from the definition of the average weight w_{avg}, and Markov's inequality. □

Notation. For the case that each cluster in $\mathcal{C}_T$ has size at least $2\epsilon n$, define the *critical distance* $d_{crit} = \frac{\alpha w_{avg}}{5\epsilon}$, else define $d_{crit} = \frac{\alpha w_{avg}}{10\epsilon}$; note that these quantities are 1/5 times the values in properties (a) and (a') respectively of Lemma 3.1.

DEFINITION 2. *Define point $x \in S$ to be* good *if both $w(x) < d_{crit}$ and $w_2(x) - w(x) \geq 5d_{crit}$, else x is called* bad. *Let $X_i \subseteq C^*_i$ be the* good *points in the optimal cluster C^*_i, and let $B = S \setminus (\cup X_i)$ be the bad points.*

PROPOSITION 3.2. *If the instance $(\mathcal{M}, S)$ satisfies $(1+\alpha, \epsilon)$-approximation-stability for the k-median objective, then*

(i) If each cluster in $\mathcal{C}_T$ has size at least $2\epsilon n$, then $|B| < (1 + 5/\alpha)\epsilon n$.

(ii) For the case of general cluster sizes in $\mathcal{C}_T$, $|B| < (6 + 10/\alpha)\epsilon n$.

PROOF. By Lemma 3.1(a), the number of points on which $\mathcal{C}^*$ and $\mathcal{C}_T$ agree where $w_2(x) - w(x) < 5d_{crit}$ is at most $(\epsilon - \epsilon^*)n$, and there can be at most $\epsilon^* n$ additional such points where $\mathcal{C}^*$ and $\mathcal{C}_T$ disagree. Setting $t = 5$ in Lemma 3.1(b) bounds the number of points that have $w(x) \geq d_{crit}$ by $(5\epsilon/\alpha)n$, proving (i). The proof of (ii) similarly follows from Lemma 3.1(a'), and applying Lemma 3.1(b) with $t = 10$. □

Definition 3 (Threshold Graph). *Define the* τ-threshold graph $G_\tau = (S, E_\tau)$ *to be the graph produced by connecting all pairs* $\{x, y\} \in \binom{S}{2}$ *with* $d(x, y) < \tau$.

Lemma 3.3 (Threshold Graph Lemma). *For an instance satisfying* $(1+\alpha, \epsilon)$-*approximation-stability and* $\tau = 2d_{crit}$*, the threshold graph* G_τ *has the following properties:*

(i) For all x, y *in the same* X_i*, the edge* $\{x, y\} \in E(G_\tau)$.

(ii) For $x \in X_i$ *and* $y \in X_{j \neq i}$*,* $\{x, y\} \notin E(G_\tau)$*. Moreover, such points* x, y *do not share any neighbors in* G_τ.

Proof. For part (i), since x, y are both good, they are at distance less than d_{crit} to their common cluster center c_i^*, by definition. Hence, by the triangle inequality, the distance

$$d(x, y) \leq d(x, c_i^*) + d(c_i^*, y) < 2 \times d_{crit} = \tau.$$

For part (ii), note that the distance from any good point x to any other cluster center, and in particular to y's cluster center c_j^*, is at least $5d_{crit}$. Again by the triangle inequality,

$$d(x, y) \geq d(x, c_j^*) - d(y, c_j^*) \geq 5d_{crit} - d_{crit} = 2\tau.$$

Since each edge in G_τ is between points at distance less than τ, the points x, y cannot share any common neighbors. □

Hence, the graph G_τ for the above value of τ is fairly simple to describe: each X_i forms a clique, and its neighborhood $N_{G_\tau}(X_i) \setminus X_i$ lies entirely in the bad set B with no edges going between X_i and $X_{j \neq i}$, or between X_i and $N_{G_\tau}(X_{j \neq i})$. We now show how we can use this structure to find a clustering of error at most ϵ if the size of each X_i is large (Section 3.2) and how we can get error $O(\epsilon)$ for general cluster sizes (Section 3.3).

3.2 An algorithm for Large Clusters

We begin with the following lemma.

Lemma 3.4. *There is a deterministic polynomial-time algorithm that given a graph* $G = (S, E)$ *satisfying properties (i), (ii) of Lemma 3.3 and given* $b \geq |B|$ *such that each* $|X_i| \geq b + 2$*, outputs a* k*-clustering with each* X_i *contained in a distinct cluster.*

Proof. Construct a graph $H = (S, E')$ where we place an edge $\{x, y\} \in E'$ if x and y have at least b common neighbors in G. By property (i), each X_i is a clique of size $\geq b + 2$ in G, so each pair $x, y \in X_i$ has at least b common neighbors in G and hence $\{x, y\} \in E'$. Now consider $x \in X_i \cup N_G(X_i)$, and $y \notin X_i \cup N_G(X_i)$: we claim there is no edge between x, y in this new graph H. Indeed, by property (ii), x and y cannot share neighbors that lie in X_i (since $y \notin X_i \cup N_G(X_i)$), nor in some $X_{j \neq i}$ (since $x \notin X_j \cup N_G(X_j)$). Hence the common neighbors of x, y all lie in B, which has size at most b. Moreover, at least one of x and y must itself belong to

B for them to have any common neighbors at all (again by property (ii))—hence, the number of distinct common neighbors is at most $b-1$, which implies that $\{x,y\} \notin E'$.

Thus each X_i is contained within a distinct component of the graph H. Note that the component containing some X_i may also contain some vertices from B; moreover, there may also be components in H that only contain vertices from B. But since the X_i's are larger than B, we can obtain the claimed clustering by taking the largest k components in H, and adding the vertices of all other smaller components in H to any of these, using this as the k-clustering. □

We now show how we can use Lemma 3.4 to find a clustering that is ϵ-close to $\mathcal{C}_T$ when all clusters are large. For simplicity, we begin by assuming that we are given the value of $w_{avg} = \frac{\mathrm{OPT}}{n}$, and then we show how this assumption can be removed.

Theorem 3.5 (Large clusters, known w_{avg}). *There is an efficient algorithm such that if the given instance $(\mathcal{M}, S)$ satisfies $(1+\alpha,\epsilon)$-approximation-stability for the k-median objective and each cluster in $\mathcal{C}_T$ has size at least $(3+10/\alpha)\epsilon n + 2$, then given w_{avg} it will find a clustering that is ϵ-close to $\mathcal{C}_T$.*

Proof. Let us define $b := (1+5/\alpha)\epsilon n$. By assumption, each cluster in the target clustering has at least $(3+10/\alpha)\epsilon n + 2 = 2b + \epsilon n + 2$ points. Since the *optimal k-median clustering* $\mathcal{C}^*$ differs from the target clustering by at most $\epsilon^* n \le \epsilon n$ points, each cluster C_i^* in $\mathcal{C}^*$ must have at least $2b+2$ points. Moreover, by Proposition 3.2(i), the bad points B have $|B| \le b$, and hence for each i,

$$|X_i| = |C_i^* \setminus B| \ge b+2.$$

Now, given w_{avg}, we can construct the graph G_τ with $\tau = 2d_{crit}$ (which we can compute from the given value of w_{avg}), and apply Lemma 3.4 to find a k-clustering $\mathcal{C}'$ where each X_i is contained within a distinct cluster. Note that this clustering $\mathcal{C}'$ differs from the optimal clustering $\mathcal{C}^*$ only in the bad points, and hence, $dist(\mathcal{C}', \mathcal{C}_T) \le \epsilon^* + |B|/n \le O(\epsilon + \epsilon/\alpha)$. However, our goal is to get ϵ-close to the target, which we do as follows.

Call a point x "red" if it satisfies condition (a) in Lemma 3.1 (i.e., $w_2(x) - w(x) < 5d_{crit}$), "yellow" if it is not red but satisfies condition (b) in Lemma 3.1 with $t=5$ (i.e., $w(x) \ge d_{crit}$), and "green" otherwise. So, the green points are those in the sets X_i, and we have partitioned the bad set B into red points and yellow points. Let $\mathcal{C}' = \{C_1', \ldots, C_k'\}$ and recall that $\mathcal{C}'$ agrees with $\mathcal{C}^*$ on the green points, so without loss of generality we may assume $X_i \subseteq C_i'$. We now construct a new clustering $\mathcal{C}''$ that agrees with $\mathcal{C}^*$ on both the green and yellow points. Specifically, for each point x and each cluster C_j', compute the median distance $d_{median}(x, C_j')$ between x and all points in C_j'; then insert x into the cluster C_i'' for $i = \mathrm{argmin}_j d_{median}(x, C_j')$. Since each non-red point x satisfies $w_2(x) - w(x) \ge 5d_{crit}$, and all green points g satisfy $w(g) < d_{crit}$, this means that any non-red point x must satisfy the following two conditions: (1) for a green point g_1 in the *same* cluster as x in $\mathcal{C}^*$ we have

$$d(x, g_1) \le w(x) + d_{crit},$$

and (2) for a green point g_2 in a *different* cluster than x in $\mathcal{C}^*$ we have

$$d(x, g_2) \ge w_2(x) - d_{crit} \ge w(x) + 4d_{crit}.$$

Therefore, $d(x, g_1) < d(x, g_2)$. Since each cluster in $\mathcal{C}'$ has a strict majority of green points (even with point x removed) all of which are clustered as in $\mathcal{C}^*$, this means that for a non-red point x, the median distance to points in its correct cluster with respect to $\mathcal{C}^*$ is less than the median distance to points in any incorrect cluster. Thus, $\mathcal{C}''$ agrees with $\mathcal{C}^*$ on all non-red points. Finally, since there are at most $(\epsilon - \epsilon^*)n$ red points on which $\mathcal{C}_T$ and $\mathcal{C}^*$ agree by Lemma 3.1—and $\mathcal{C}''$ and $\mathcal{C}_T$ might disagree on all these points—this implies

$$dist(\mathcal{C}'', \mathcal{C}_T) \leq (\epsilon - \epsilon^*) + \epsilon^* = \epsilon$$

as desired. For convenience, the above procedure is given as Algorithm 1 below. □

```
Algorithm 1 $k$-median Algorithm: Large Clusters (given a guess $w$ of $w_{avg}$)
Input: $w$, $\epsilon \leq 1$, $\alpha > 0$, $k$.
Step 1: Construct the $\tau$-threshold graph $G_\tau$ with $\tau = 2d_{crit} = \frac{1}{5}\frac{\alpha w}{\epsilon}$.
Step 2: Apply the algorithm of Lemma 3.4 to find an initial clustering $\mathcal{C}'$. Specifically,
    construct graph $H$ by connecting $x, y$ if they share at least $b = (1+5/\alpha)\epsilon n$
    neighbors in $G_\tau$ and let $C'_1, \ldots, C'_k$ be the $k$ largest components of $H$.
Step 3: Produce clustering $\mathcal{C}''$ by reclustering according to smallest median distance
    in $\mathcal{C}'$. That is, $\mathcal{C}''_i = \{x : i = \text{argmin}_j d_{median}(x, C'_j)\}$.
Step 4: Output the $k$ clusters $\mathcal{C}''_1, \ldots, \mathcal{C}''_k$.
```

We now extend the above argument to the case where we are not given the value of w_{avg}.

Theorem 3.6 (Large Clusters, unknown w_{avg}). *There is an efficient algorithm that if the given instance $(\mathcal{M}, S)$ satisfies $(1+\alpha, \epsilon)$-approximation-stability for the k-median objective, and each cluster in $\mathcal{C}_T$ has size at least $(4+15/\alpha)\epsilon n+2$, will produce a clustering that is ϵ-close to $\mathcal{C}_T$.*

Proof. The algorithm for the case that we are not given the value w_{avg} is the following: we run Steps 1 and 2 of Algorithm 1 repeatedly for different guesses w of w_{avg}, starting with $w = 0$ (so the graph G_τ is empty) and at each step increasing w to the next value such that G_τ contains at least one new edge (so we have at most n^2 different guesses to try). If the current value of w causes the k largest components of H to miss more than $b := (1 + 5/\alpha)\epsilon n$ points, or if any of these components has size $\leq b$, then we discard this guess w, and try again with the next larger guess for w. Otherwise, we run Algorithm 1 to completion and let $\mathcal{C}''$ be the clustering produced.

Note that we still might have $w < w_{avg}$, but this just implies that the resulting graphs G_τ and H can only have fewer edges than the corresponding graphs for the correct w_{avg}. Hence, some of the X_i's might not have fully formed into connected components in H. However, if the k largest components together miss at most b points, then this implies we must have at least one component for each X_i, and therefore exactly one component for each X_i. So, we never misclassify the good points lying in these largest components. We might misclassify all the bad points (at most b of these), and might fail to cluster at most b of the points in the actual X_i's (i.e., those not lying in the largest k components), but this nonetheless guarantees that each cluster $\mathcal{C}'_i$ contains at least $|X_i| - b \geq b + 2$ correctly clustered green points (with respect to $\mathcal{C}^*$) and at most b misclassified points. Therefore, as shown in the proof of Theorem 3.5, the resulting clustering $\mathcal{C}''$ will correctly cluster all non-red points as in $\mathcal{C}^*$ and so is at distance at most $(\epsilon - \epsilon^*) + \epsilon^* = \epsilon$ from $\mathcal{C}_T$. For convenience, this procedure is given as Algorithm 2 below. □

```
Algorithm 2 $k$-median Algorithm: Large Clusters (unknown $w_{avg}$)
Input: $\epsilon \leq 1$, $\alpha > 0$, $k$.
For $j = 1, 2, 3 \ldots$ do:
  Step 1: Let $\tau$ be the $j$th smallest pairwise distance in $S$. Construct $\tau$-threshold graph $G_\tau$.
  Step 2: Run Step 2 of Algorithm 1 to construct graph $H$ and clusters $C'_1, \ldots, C'_k$.
  Step 3: If $\min(|C'_1|, \ldots, |C'_k|) > b$ and $|C'_1 \cup \ldots \cup C'_k| \geq n(1 - \epsilon - 5\epsilon/\alpha)$, run
    Step 3 of Algorithm 1 and output the clusters $C''_1, \ldots, C''_k$ produced.
```

3.3 An Algorithm for the General Case

The algorithm in the previous section required the minimum cluster size in the target to be large (of size $\Omega(\epsilon n)$). In this section, we show how this requirement can be removed using a different algorithm that finds a clustering that is $O(\epsilon/\alpha)$-close to the target; while the algorithm is just as simple, we need to be a bit more careful in the analysis.

```
Algorithm 3 $k$-median Algorithm: General Case
Input: $\epsilon \leq 1$, $\alpha > 0$, $k$.
Initialization: Run a constant-factor $k$-median approximation algorithm to compute a value $w \in [w_{avg}, \beta w_{avg}]$ for, say, $\beta = 4$.
Step 1: Construct the $\tau$-threshold graph $G_\tau$ with $\tau = \frac{1}{5}\frac{\alpha w}{\beta \epsilon}$.
Step 2: For $j = 1$ to $k$ do:
    Identify the vertex $v_j$ of highest degree in $G_\tau$.
    Remove $v_j$ and its neighborhood from $G_\tau$ and call this cluster $C(v_j)$.
Step 3: Output the $k$ clusters $C(v_1), \ldots, C(v_{k-1}), S - \cup_{i=1}^{k-1} C(v_i)$.
```

Theorem 3.7 (k-Median: General Case). *There is an efficient algorithm that if the given instance $(\mathcal{M}, S)$ satisfies $(1 + \alpha, \epsilon)$-approximation-stability for the k-median objective, will produce a clustering that is $O(\epsilon + \epsilon/\alpha)$-close to $\mathcal{C}_T$.*

Proof. The algorithm is as given above in Algorithm 3. First, if we are not given the value of w_{avg}, we run a constant-factor k-median approximation algorithm (e.g., [Arya et al. 2004]) to compute an estimate $\hat{w} \in [w_{avg}, \beta w_{avg}]$ for, say, $\beta =$

4.[4] Redefining $\hat{d}_{crit} = \frac{\alpha \hat{w}}{10\beta\epsilon} \leq \frac{\alpha w_{avg}}{10\epsilon}$, as in the proof of Proposition 3.2(ii), but using Lemma 3.1(b) with $t = 10\beta$, we have that the set $B = \{x \in S \mid w(x) \geq \hat{d}_{crit}$ or $w_2(x) - w(x) < 5\hat{d}_{crit}\}$ of bad points has size $|B| \leq (6 + 10\beta/\alpha)\epsilon n$. If we again define $X_i = C_i^* \setminus B$, we note that Lemma 3.3 continues to hold with $\tau = 2\hat{d}_{crit}$: the graph G_τ satisfies properties *(i),(ii)* that all pairs of points in the same X_i are connected by an edge and all pairs of points in different X_i, X_j have no edge and no neighbors in common. In summary, the situation is much as if we knew w_{avg} exactly, except that the number of bad points is slighly greater.

We now show that the greedy method of Step 2 above correctly captures most of the cliques $X_1, X_2, \ldots, X_k$ in G_τ—in particular, we show there is a bijection $\sigma : [k] \rightarrow [k]$ such that $\sum_i |X_{\sigma(i)} \setminus C(v_i)| = O(b)$, where $b = |B|$. Since the b bad points may potentially all be misclassified, this gives an additional error of b.

Let us think of each clique X_i as initially "unmarked", and then "marking" it the first time we choose a cluster $C(v_j)$ that intersects it. We now consider two cases. If the j^{th} cluster $C(v_j)$ intersects some *unmarked* clique X_i, we will assign $\sigma(j) = i$. (Note that it is not possible for $C(v_j)$ to intersect two cliques X_i and $X_{j \neq i}$, since by Lemma 3.3*(ii)* these cliques have no common neighbors.) If $C(v_j)$ misses r_i points from X_i, then since the vertex v_j defining this cluster had maximum degree and X_i is a clique, $C(v_j)$ must contain at least r_i elements from B. Therefore the total sum of these r_i can be at most $b = |B|$, and hence $\sum_j |X_{\sigma(j)} \setminus C(v_j)| \leq b$, where the sum is over j's that correspond to the first case.

The other case is if $C(v_j)$ intersects a previously marked clique X_i. In this case we assign $\sigma(j)$ to any arbitrary clique $X_{i'}$ that is not marked by the end of the process. Note that the total number of points in such $C(v_j)$'s must be at most the number of points remaining in the marked cliques (i.e., $\sum_j r_j$), and possibly the bad points (at most b of them). Since the cliques $X_{i'}$ were unmarked at the end, the size of any such $X_{i'}$ must be bounded by the sizes of its matched $C(v_j)$—else we would have picked a vertex from this clique rather than picking v_j. Hence the total size of such $X_{i'}$ is bounded by $|B| + \sum_i r_i \leq 2b$; in turn, this shows that $\sum_j |X_{\sigma(j)} \setminus C(v_j)| \leq \sum_j |X_{\sigma(j)}| \leq 2b$, where this sum is over j's that correspond to the second case. Therefore, overall, the total error over all $C(v_j)$ with respect to the k-median optimal is the two sums above, plus potentially the bad points, which gives us at most $4b$ points. Adding in the extra $\epsilon^* n$ to account for the distance between the k-median optimum and the target clustering yields the claimed $4b + \epsilon^* n = O(\epsilon + \epsilon/\alpha)n$ result. □

4. THE K-MEANS PROBLEM

Algorithm 3 in Section 3.3 for the k-median problem can be easily altered to work for the k-means problem as well. Indeed, if we can prove the existence of a structure like that promised by Lemma 3.1 and Lemma 3.3 (albeit with different parameters), the same algorithm and proof would give a good clustering for any objective function.

Given some optimal solution for k-means define $w(x) = \min_i d(x, c_i)$ to be the

[4] The reason we need to do this, rather than simply increasing an initial low guess of w_{avg} as in the proof of Theorem 3.6, is that we might split some large cluster causing substantial error, and not be able to recognize our mistake (because we only miss small clusters which do not result in very many points being left over).

distance of x to its center, which is *the square root of* x's contribution to the k-means objective function; hence $\text{OPT} = \sum_x w(x)^2$. Again, let $w_2(x) = \min_{j \neq i} d(x, c_j)$ be the distance to the second-closest center, and let $\epsilon^* = dist(\mathcal{C}_T, \mathcal{C}^*)$.

LEMMA 4.1. *If the instance $(\mathcal{M}, S)$ satisfies $(1+\alpha, \epsilon)$-approximation-stability for the k-means objective, then*

(a) If each cluster in $\mathcal{C}_T$ has size at least $2\epsilon n$, then less than $(\epsilon - \epsilon^*)n$ points $x \in S$ on which $\mathcal{C}_T$ and $\mathcal{C}^*$ agree have $w_2(x)^2 - w(x)^2 < \frac{\alpha \text{OPT}}{\epsilon n}$.

(a') For the case of general cluster sizes in $\mathcal{C}_T$, less than $6\epsilon n$ points $x \in S$ have $w_2(x)^2 - w(x)^2 < \frac{\alpha \text{OPT}}{2\epsilon n}$.

Also, for any $t > 0$ we have:

(b) at most $t(\epsilon n/\alpha)$ points $x \in S$ have $w(x)^2 \geq \frac{\alpha \text{OPT}}{t\epsilon n}$.

The proof is identical to the proof for Lemma 3.1, and is omitted here. We now give some details for what changes are needed to make Algorithm 2 from Section 3.3 work here. Again, we use a β-approximation to k-means for some constant β to get $\widehat{OPT} \in [\text{OPT}, \beta\text{OPT}]$. Define the critical distance $\hat{d}_{crit}$ as $(\frac{\alpha\widehat{OPT}}{25\epsilon\beta n})^{1/2}$ in the case of large clusters, or $(\frac{\alpha\widehat{OPT}}{50\epsilon\beta n})^{1/2}$ in the case of general cluster sizes—these are at most 1/5 times the square-roots of the expressions in (a) and (a') above. Call point $x \in S$ *good* if both $w(x) < d_{crit}$ and $w_2(x) \geq 5d_{crit}$, and *bad* otherwise; let B be the bad points. The following proposition has a proof very similar to Proposition 3.2(b).

PROPOSITION 4.2. *If the instance $(\mathcal{M}, S)$ satisfies $(1+\alpha, \epsilon)$-approximation-stability for the k-means objective, then $|B| < (6 + 50\beta/\alpha)\epsilon n$.*

Now the rest of the proof for Theorem 3.7 goes through unchanged in the k-means case as well; indeed, first we note that Lemma 3.3 is true, because it only relies on the good points being at distance $< d_{crit}$ to their center, and being at distance $\geq 5d_{crit}$ to any other center, and the rest of the proof only relies on the structure of the threshold graph. The fraction of points we err on is again $\epsilon^* + 4|B|/n = O(\epsilon + \epsilon/\alpha)$. Summarizing, we have the following result.

THEOREM 4.3 (k-MEANS: GENERAL CASE). *There is an efficient algorithm that if the given instance $(\mathcal{M}, S)$ satisfies $(1+\alpha, \epsilon)$-approximation-stability for the k-means objective, will produce a clustering that is $O(\epsilon + \epsilon/\alpha)$-close to $\mathcal{C}_T$.*

4.1 An Algorithm for Large Clusters

Unfortunately, the argument for exact ϵ-closeness for k-median in the case of large target clusters does not extend directly, because Lemma 4.1(a) is weaker than Lemma 3.1(a)—the latter gives us bounds on the difference in distances, whereas the former only gives us bounds on the difference in the squared distances. Instead, however, we will use the same algorithm style to identify most of the bad points (by outputting "don't know" on some $O(\epsilon/\alpha)$ of the points) and output a clustering on the remaining $1 - O(\epsilon/\alpha)$ fraction of the points which makes at most ϵn errors on these points.

THEOREM 4.4. *There is an efficient algorithm that if the given instance $(\mathcal{M}, S)$ satisfies $(1+\alpha, \epsilon)$-approximation-stability for the k-means objective, and each cluster in $\mathcal{C}_T$ has size at least $(4 + 75/\alpha)\epsilon n + 2$, will produce a clustering in which at most $O(\epsilon n/\alpha)$ points are labeled as "don't know", and on the remainder the clustering is ϵ-close to $\mathcal{C}_T$.*

PROOF. Let us first assume that we know the value of OPT; we will discharge this assumption later. Define the *critical distance* $d_{crit} := \frac{1}{5}(\frac{\alpha \mathrm{OPT}}{\epsilon n})^{1/2}$. As in Theorem 3.5, we categorize the points in a more nuanced fashion: a point $x \in S$ is called "red" if it satisfies condition (a) of Lemma 4.1 (i.e., if $w_2(x)^2 - w(x)^2 < 25d_{crit}^2$), "yellow" if it is not red and has $w(x) \in [d_{crit}, 5d_{crit}]$, "orange" if it is not red and has $w(x) > 5d_{crit}$, and "green" otherwise. Hence, Lemma 4.1(a) tells us that there are at most $(\epsilon - \epsilon^*)n$ points on which $\mathcal{C}^*$ and $\mathcal{C}_T$ agree, and that are red; at most $25\epsilon/\alpha$ fraction are either yellow or orange (by setting $t = 25$); at most ϵ/α fraction of the points are orange (by setting $t = 1$); the rest are green. Let all the non-green points be called bad, and denoted by the set B. Let us define $b := (1 + 25/\alpha)\epsilon n$; note that $|B| \leq b$.

Now, as in Theorem 3.5, if the cluster sizes in the target clustering are at least $2b + \epsilon n + 2$, then constructing the threshold graph G_τ with $\tau = 2d_{crit}$ and applying Lemma 3.4 we can find a k-clustering $\mathcal{C}'$ where each $X_i := C_i^* \setminus B$ is contained with a distinct cluster, and only the $O(\epsilon + \epsilon/\alpha)$ bad (i.e., non-green) points are possibly in the wrong clusters. We now want to label some points as "don't knows", and construct another clustering $\mathcal{C}''$ where we correctly cluster the green and yellow points.

Again, this is done as in the k-median case: for each point x and each cluster C'_j, compute the median distance $d_{med}(x, C'_j)$ from x to the points in C'_j. If the minimum median distance $\min_{j\in[k]} d_{med}(x, C'_j)$ is greater than $4d_{crit}$, then label the point x as "don't know"; else insert x into the cluster C''_i for $i = \mathrm{argmin}_j d_{med}(x, C'_j)$.

First, we claim that the points labeled "don't know" contain all the orange points. Indeed, for any orange point x, the distance to each optimal cluster center is at least $5d_{crit}$; moreover, since the target clusters are large, a majority of the points in each cluster C'_j are green, which are all within distance d_{crit} of the optimal cluster center. Using the triangle inequality, the median distance of an orange point to every cluster center will be at least $4d_{crit}$, and hence it will be classified as "don't know". There may be more points classified this, but using a similar argument we can deduce that all such points must have $w(x) \geq 3d_{crit}$, and Lemma 4.1(b) implies that there are at most $\frac{25\epsilon n}{9\alpha}$ such "don't know" points.

Next, we show that the yellow and green points will be correctly classified. Note that each non-red point x satisfies $w_2(x)^2 - w(x)^2 \geq 25d_{crit}^2$, all yellow/green points satisfy $w(x)^2 \leq 25d_{crit}^2$, and all green points g satisfy $w(g) < d_{crit}$. We show that this means that any yellow/green point x must satisfy the property that for a green point g_1 in the *same* cluster as x in $\mathcal{C}^*$, and for a green point g_2 in a *different* cluster than x in $\mathcal{C}^*$, we have $d(x, g_1) < d(x, g_2)$. Indeed, $d(x, g_1) < w(x) + d_{crit}$ and $d(x, g_2) > w_2(x) - d_{crit}$, and hence it suffices to show that $w_2(x) \geq w(x) + 2d_{crit}$

for x being yellow or green. To show this, note that

$$\begin{aligned} w_2(x)^2 &\geq w(x)^2 + 25d_{crit}^2 \\ &\geq w(x)^2 + 4d_{crit}^2 + 4 \cdot d_{crit} \cdot (5d_{crit}) \\ &\geq w(x)^2 + 4d_{crit}^2 + 4 \cdot d_{crit} \cdot w(x) \\ &\geq (w(x) + 2d_{crit})^2 \end{aligned}$$

where we use the fact that $w(x) \leq 5d_{crit}$ for green and yellow points. Again, since each yellow or green point is closer to a strict majority of green points in their "correct" cluster in $\mathcal{C}'$, we will correctly classify them. Finally, we finish the argument as before: ignoring the $O(\epsilon n/\alpha)$ "don't knows", $\mathcal{C}''$ may disagree with $\mathcal{C}^*$ on only the $(\epsilon - \epsilon^*)n$ red points where $\mathcal{C}^*$ and $\mathcal{C}_T$ agree, and the $\epsilon^* n$ points where $\mathcal{C}^*$ and $\mathcal{C}_T$ disagree, which is ϵn as claimed.

One loose end remains: we assumed we knew OPT and hence d_{crit}. To remove this assumption, we can again try multiple guesses for the value of d_{crit} as in Theorem 3.6. The argument in that theorem continues to hold, as long as the size of the clusters in the target clustering is at least $3b + \epsilon n + 2 = (4 + 75/\alpha)\epsilon n + 2$, which is what we assumed here. □

5. THE MIN-SUM CLUSTERING PROBLEM

Recall that the min-sum k-clustering problem asks to find a k-clustering $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ to minimize the objective function

$$\Phi(\mathcal{C}) = \sum_{i=1}^{j} \sum_{x \in C_i} \sum_{y \in C_i} d(x, y).$$

In this section, we show how assuming $(1 + \alpha, \epsilon)$-approximation-stability for the min-sum clustering problem, and assuming that all the clusters in the target are "large", allows us to find a clustering that is $O(\epsilon)$-close to the target clustering.

5.1 The high-level idea

As one might expect, the general plan is to extend the basic techniques from the previous sections, though the situation is now a bit more delicate. While we can still argue that there cannot be too many points that could be cheaply reassigned to different clusters (since that would violate our basic assumption, though we have to be careful about the somewhat messy issue of multiple reassignments), now the cost of reassigning a point x to cluster C_j is proportional to the number of points in C_j. In particular, the net effect of this cost structure is that unlike the k-median and k-means objectives, there is no longer a uniform threshold or critical distance. Many points in some cluster C_i could be quite close to another cluster C_j if C_j is large. On the other hand, one can show the (good) points in C_j will be even closer to each other. Thus, by slowly growing a threshold distance, we will be able to find the clusters in the order from largest to smallest. We then argue that we can identify points in time when the size of the largest component found is large enough compared to the current threshold to have captured the cluster, allowing us to pull those clusters out before they have had the chance to mistakenly connect to smaller ones. This argument will require an assumption that all clusters are large.

(See subsequent work of [Balcan and Braverman 2009] for an algorithm that allows for general cluster sizes).

5.2 Properties of Min-Sum Clustering

Let the min-sum optimal clustering be $\mathcal{C}^* = \{C_1^*, \dots, C_k^*\}$ with objective function value $\mathrm{OPT} = \Phi(\mathcal{C}^*)$. For $x \in C_i^*$, define

$$w(x) = \sum_{y \in C_i^*} d(x, y)$$

so that $\mathrm{OPT} = \sum_x w(x)$, and let $w_{avg} = \mathrm{avg}_x w(x) = \frac{\mathrm{OPT}}{n}$. Define

$$w_2(x) = \min_{j \neq i} \sum_{y \in C_j^*} d(x, y).$$

A useful fact, following immediately from the triangle inequality, is the following:

Fact 5.1. *For two points x and y, and any cluster C_j^*,*

$$\sum_{z \in C_j^*} (d(x, z) + d(y, z)) \geq |C_j^*| \, d(x, y).$$

We now prove the following lemma.

Lemma 5.2. *If the given instance $(\mathcal{M}, S)$ satisfies $(1+\alpha, \epsilon)$-approximation-stability for the min-sum objective and each cluster in $\mathcal{C}_T$ has size at least $2\epsilon n$, then:*

(a) less than $(\epsilon - \epsilon^*)n$ points $x \in S$ on which $\mathcal{C}_T$ and $\mathcal{C}^*$ agree have $w_2(x) < \frac{\alpha w_{avg}}{4\epsilon}$, and

(b) at most $60\epsilon n/\alpha$ points $x \in S$ have $w(x) > \frac{\alpha w_{avg}}{60\epsilon}$.

Proof. To prove Property (a), assume to the contrary. Then one could take $\mathcal{C}^*$ and move a set S' of $(\epsilon - \epsilon^*)n$ points x that have $w_2(x) < \frac{\alpha w_{avg}}{4\epsilon}$ and on which $\mathcal{C}_T$ and $\mathcal{C}^*$ agree to the clusters that define their w_2 value. We now argue that the resulting increase in min-sum objective value is less than $\alpha\mathrm{OPT}$.

Let the new clustering be $\mathcal{C}' = (C_1', \dots, C_k')$, where $|C_i' \setminus C_i^*| = \delta_i n$, so that $\sum_i \delta_i = \epsilon - \epsilon^*$. Also, let $C_2(x)$ denote the cluster C_i^* that point $x \in S'$ is moved to. Then, for each point $x \in S'$ moved, the increase to the min-sum objective is at most $2w_2(x) + \sum_{y \in S' : C_2(y) = C_2(x)} d(x, y)$—here the factor of two arises because the min-sum objective counts each pair of points in a cluster twice, once from each end. From Fact 5.1, we know that if $C_2(y) = C_2(x)$ then $d(x, y) \leq \frac{1}{|C_2(x)|}(w_2(x) + w_2(y))$. Thus, we can replace the term $d(x, y)$ in the cost charged to point x with $\frac{2}{|C_2(x)|} w_2(x)$, yielding a total cost charged to point x moved to cluster C_i^* of

$$2w_2(x) + 2w_2(x) \, \frac{\delta_i n}{|C_i^*|}.$$

Summing over all points x moved to all clusters, and using the fact that $w_2(x) < \frac{\alpha w_{avg}}{4\epsilon}$ for all $x \in S'$, we have a total cost increase of less than

$$\begin{aligned}\sum_i (\delta_i n) \frac{2\alpha w_{avg}}{4\epsilon}\left[1 + \frac{\delta_i n}{|C_i^*|}\right] &\leq \epsilon n \frac{\alpha w_{avg}}{2\epsilon} + \frac{\alpha w_{avg}}{2\epsilon}\sum_i \frac{\delta_i^2 n^2}{|C_i^*|} \\ &\leq \frac{\alpha}{2}\mathrm{OPT} + \frac{\alpha w_{avg}}{2\epsilon}\frac{\epsilon^2 n^2}{\min_i |C_i^*|} \\ &\leq \frac{\alpha}{2}\mathrm{OPT} + \frac{\alpha}{4}\mathrm{OPT} < \alpha \mathrm{OPT}.\end{aligned}$$

Finally, property (b) follows immediately from Markov's inequality. □

Let us define the *critical value* $v_{crit} := \frac{\alpha w_{avg}}{60\epsilon}$. We call point x *good* if it satisfies both $w(x) \leq v_{crit}$ and $w_2(x) \geq 15 v_{crit}$, else x is called *bad*; let X_i be the *good* points in the optimal cluster C_i^*, and let $B = S \setminus \cup_i X_i$ be the bad points.

Lemma 5.3 (Structure of Min-Sum optimum). *If the given instance $(\mathcal{M}, S)$ satisfies $(1+\alpha, \epsilon)$-approximation-stability for the min-sum objective then as long as the minimum cluster size is at least $2\epsilon n$ we have:*

(i) For all x, y in the same X_i, we have $d(x,y) < \frac{2v_{crit}}{|C_i^*|}$,

(ii) For $x \in X_i$ and $y \in X_{j \neq i}$, we have $d(x,y) > \frac{14\, v_{crit}}{\min(|C_i^*|,|C_j^*|)}$, and

(iii) The number of bad points $|B| = |S \setminus \cup_i X_i|$ is at most $b := (1 + 60/\alpha)\epsilon n$.

Proof. For part (i), note that Fact 5.1 implies that

$$d(x,y) \leq \frac{1}{|C_i^*|}\sum_{z \in C_i^*}(d(x,z) + d(y,z)) = \frac{1}{|C_i^*|}(w(x) + w(y)).$$

Since $x, y \in X_i$ are both good, we have $w(x), w(y) \leq v_{crit}$, so part (i) follows.

For part (ii), assume without loss of generality that $|C_j^*| \leq |C_i^*|$. Since both $x \in C_i^*, y \in C_j^*$ are good, we have $w_2(x) = \sum_{z \in C_j^*} d(x,z) \geq 15 v_{crit}$ and $w(x) = \sum_{z \in C_j^*} d(y,z) \leq v_{crit}$. By the triangle inequality $d(x,y) \geq d(x,z) - d(y,z)$, we have

$$|C_j^*|\, d(x,y) \geq \sum_{z \in C_j^*}(d(x,z) - d(y,z)) = w_2(x) - w(y) \geq 14 v_{crit}.$$

Finally, part (iii) follows from Lemma 5.2 and a trivial union bound. □

While Lemma 5.3 is similar in spirit to Lemma 3.3, there is a crucial difference: the distance between the good points in X_i and those in X_j is no longer bounded below by some absolute value τ, but rather the bound depends on the sizes of X_i and X_j. However, a redeeming feature is that the separation is large compared to the sizes of *both* X_i and X_j; we will use this feature crucially in our algorithm.

5.3 The Algorithm for Min-Sum Clustering

For the algorithm below, define *critical thresholds* $\tau_0, \tau_1, \tau_2, \ldots$ as: $\tau_0 = 0$ and τ_i is the ith smallest distinct distance $d(x,y)$ for $x, y \in S$. Thus, $G_{\tau_0}, G_{\tau_1}, \ldots$ are the only distinct threshold graphs possible.

THEOREM 5.4. *If the given instance $(\mathcal{M}, S)$ satisfies $(1+\alpha,\epsilon)$-approximation-stability for the min-sum objective and we are given the value of w_{avg}, then so long as each cluster in $\mathcal{C}_T$ has size greater than $(5+120/\alpha)\epsilon n$, Algorithm 4 produces a clustering that is $O(\epsilon/\alpha)$-close to the target. If we are not given w_{avg}, there is an efficient algorithm that uses Algorithm 4 as a subroutine and on any such instance will produce a clustering that is $O(\epsilon/\alpha)$-close to the target.*

Algorithm 4 Min-sum Algorithm

```
Input: $(\mathcal{M}, S)$, $w_{avg}$, $\epsilon \le 1$, $\alpha > 0$, $k$, $b := (1+60/\alpha)\epsilon n$.
Let the initial threshold $\tau = \tau_0$.

Step 1: If $k = 0$ or $S = \emptyset$, stop.
Step 2: Construct the $\tau$-threshold graph $G_\tau$ on the current set $S$ of points.
Step 3: Create a new graph $H$ by connecting two points in $S$ by an edge if they
    share at least $b$ neighbors in common in $G_\tau$.
Step 4: Let $C$ be largest connected component in $H$. If $|C| \ge 3v_{crit}/\tau$,
    then output $C$ as a cluster, set $k \leftarrow k-1$, $S \leftarrow S \setminus C$, and go to Step 1,
    else increase $\tau$ to the next critical threshold and go to Step 1.
```

PROOF. Since each cluster in the target clustering has more than $(5+120/\alpha)\epsilon n = 2b + 3\epsilon n$ points by the assumption, and the *optimal min-sum clustering* $\mathcal{C}^*$ must differ from the target clustering by fewer than ϵn points, each cluster in $\mathcal{C}^*$ must have more than $2b + 2\epsilon n$ points. Moreover, by Lemma 5.2(iii), the bad points B constitute at most b points, and hence each $|X_i| = |C_i^* \setminus B| > b + 2\epsilon n \ge b + 2$.

Analysis under the assumption that w_{avg} is given. Consider what happens in the execution of the algorithm: as we increase τ, the sizes of the H-components increase (since we are adding more edges in G_τ). This happens until the largest H-component is "large enough" (i.e., the condition in Step 4 gets satisfied) and we output a component whose size is large enough; and then we go back to raising τ.

We claim that every time we output a cluster in Step 4, this cluster completely contains some X_i and includes no points in any $X_{j\neq i}$. More specifically, we show that as we increase τ, the condition in Step 4 will be satisfied *after* all the good points in the some cluster have been fully connected, but *before* any edges appear between good points in different clusters. It suffices to show that the first cluster output by the algorithm contains some X_i entirely; the claim for the subsequent output clusters is the same. Assume that $|C_1^*| \ge |C_2^*| \ge \ldots \ge |C_k^*|$. Define $d_i = \frac{2v_{crit}}{|C_i^*|}$ and recall that $\max_{x,y\in X_i} d(x,y) \le d_i$ by Lemma 5.3(i).

We first claim that as long as $\tau \le 3\,d_1$, no two points belonging to different X_i's can lie in the same H-component. By Lemma 5.3(ii) the distance between points in any X_i and $X_{j\neq i}$ is strictly greater than $\frac{14v_{crit}}{\min(|C_i^*|,|C_j^*|)}$, which is strictly greater than 2τ for any $\tau \le 3\,d_1$. Hence every $x \in X_i$ and $y \in X_j$ share no common neighbors, and by an argument identical to that in Lemma 3.4, the nodes x, y belong to different components of H.

Next, we claim that for values of $\tau < \min\{d_i, 3d_1\}$, the H-component containing points from X_i cannot be output by Step 4. Indeed, since $\tau < 3d_1$, no X_i and X_j

belong to the same H-component by the argument in the previous paragraph, and hence any H-component containing points from X_i has size at most $|C_i^*| + |B| < \frac{3|C_i^*|}{2}$; here we used the fact that each $|C_i^*| > 2b$ due to the large cluster assumption. However, the minimum size bound $\frac{3v_{crit}}{\tau}$ in Step 4 is equal to $\frac{3\,d_i\,|C_i^*|}{2\tau} \geq \frac{3|C_i^*|}{2}$ for values of $\tau < d_i$, where we used the definition of d_i, and that $d_i > \tau$. Hence the condition of Step 4 is not satisfied and the H-component will not be output. Moreover, note that when $\tau \geq d_i$, all the points of X_i lie in the same H-component.

The above two paragraphs show that nothing bad happens: no incorrect components are constructed or components outputted prematurely. We finally show that something good happens—in particular, that the condition in Step 4 becomes true for some H-component fully containing some X_i for some value $\tau = [d_1, 3d_1]$. (By the argument in the previous paragraph, $\tau \geq d_i$, and hence the output component will fully contain X_i.) For the sake of contradiction, suppose not. But note at time $\tau = 3d_1$, at least the H-component containing X_1 has size at least $|C_1^*| - |B| > |C_1^*|/2$ and will satisfy the minimum-size condition (which at time $\tau = 3d_1$ requires a cluster of size $\frac{3v_{crit}}{\tau} = \frac{v_{crit}}{d_1} = |C_1^*|/2$), giving the contradiction.

To recap, we showed that by time $3d_1$ none of the clusters have merged together, and the Step 4 condition was satisfied for at least the component containing X_1 (and hence for the largest component) at some time prior to that. Moreover, this largest component must fully contain some set X_i and no points in $X_{j\neq i}$. Finally, we can now iterate this argument on remaining set of points to complete the proof for the case that we know w_{avg}.

Analysis if w_{avg} is not given. In this case, we do not want to use a β-approximation algorithm for min-sum to obtain a clustering that is $O(\beta\epsilon/\alpha)$-close to the target, because the minsum clustering problem only has a logarithmic approximation for arbitrary k, and hence our error would blow up by a logarithmic factor. Instead, we use the idea of trying increasing values of w_{avg}. Specifically, the approximation algorithm gives us upper and lower bounds for w_{avg} that differ by a logarithmic factor, hence we can start at the lower bound for w_{avg}, and try increasing powers of 2: this ensures that we would try the process only $O(\log\log n)$ times before we reach the correct value of w_{avg}. Since we don't really know this correct value, we stop the first time we output k clusters that cover at least $n - b = (1 - O(\epsilon/\alpha))n$ points in S. Clearly, if we reached the correct value of w_{avg} we would succeed in covering all the good $n - b$ points using our k clusters; we now argue that we will never mistakenly output a high-error clustering.

The argument is as follows. Let us say we *mark* X_i the first time we output a cluster containing at least one point from it. There are three possible sources of mistakes: (a) we may output a cluster prematurely: it may contain some but not all points from X_i, (b) we may output a cluster which contains points from one or more previously marked sets X_j (but no unmarked X_i), or (c) we may output a cluster with points from an unmarked X_i and one or more previously marked X_j. In case (a), if we end up clustering all but an $O(\epsilon/\alpha)$-fraction of the points, we did not miss too many points from the X_i's, so our error is $O(\epsilon/\alpha)$. In case (b), such an event would use up an additional cluster and therefore would end with missing some X_i completely, which would result in more than b unclustered points, and we would try a larger guess for w_{avg}. The dangerous case is case (c), but we claim case (c) in fact cannot happen. Indeed, the value of τ at which we would form connected components containing points from both X_i and X_j is a constant times larger than the value τ at which all of X_i would be in a single H-component. Moreover, since our guess for w_{avg} is too small, this H-component would certainly satisfy the condition of Step 4 and be output as a cluster instead. □

6. RELATIONSHIP TO ϵ-SEPARATION CONDITION

Ostrovsky et al. [2006] consider k-means clustering in Euclidean spaces, and define and analyze a very interesting separation condition that provides a notion of how "naturally clustered" a given dataset is. Specifically, they call a k-means instance *ϵ-separated* if the optimal k-means cost is at most ϵ^2 times the cost of the optimal $(k-1)$-means solution. Under this assumption on the input, they show how to seed Lloyd's method to obtain a $1+f(\epsilon)$ approximation in d-dimensional Euclidean space in time $O(nkd + k^3d)$, and a $(1+\delta)$-PTAS with run-time $nd2^{k(1+\epsilon^2)/\delta}$. This notion of ϵ-separation, namely that any $(k-1)$-means solution is substantially more expensive than the optimal k-means solution, is in fact related to (c, ϵ)-approximation-stability. Indeed, in Theorem 5.1 of their paper, they show that their ϵ-separatedness assumption implies that any near-optimal solution to k-means is $O(\epsilon^2)$-close to the k-means optimal clustering. However, the converse is not necessarily the case: an instance could satisfy approximation-stability without being ϵ-separated.[5] We present here a specific example with $c = 2$, in fact of a point set in Euclidean space. Consider $k = 2$ where target cluster C_1 has $(1-\delta)n$ points and target cluster C_2 has δn points, with δ a parameter to be fixed later. Suppose that any two points inside the same cluster C_i have distance 1 and any two points inside different clusters have distance $1 + 1/\epsilon$. Choosing any $\delta \in (\epsilon, 1-\epsilon)$, the resulting example satisfies $(2, \epsilon)$-approximation-stability for k-median, and choosing any $\delta \in (\epsilon^2, 1-\epsilon^2)$, the resulting example satisfies $(2, \epsilon^2)$-approximation-stability for k-means. However, it need not satisfy the ϵ-separation property: for $\delta = 2\epsilon$, the optimal 2-median solution has cost $n-2$, but the optimal 1-median solution picks a center at any point in the cluster of size $(1-2\epsilon)n$ and hence has cost $(1-2\epsilon)n - 1 + (2\epsilon n)(1 + 1/\epsilon) = 3n - 1$. Likewise for $\delta = 2\epsilon^2$, the optimal 2-means solution has cost $n-2$, but the optimal 1-means solution has cost less than $(3+4\epsilon)n$. Thus, in both cases the ratio of costs between $k = 1$ and $k = 2$ is not so large.

In fact, for the case that k is much larger than $1/\epsilon$, the difference between the two properties can be more substantial. Suppose ϵ is a small constant, and consider a clustering instance in which the target consists of $k = \sqrt{n}$ clusters with $\sqrt{n}$ points each, such that all points in the same cluster have distance 1 and all points in different clusters have distance $D+1$ where D is a large constant. Then, merging two clusters increases the cost additively by $\Theta(\sqrt{n})$, since D is a constant. Consequently, the optimal $(k-1)$-means/median solution is just a factor $1 + O(1/\sqrt{n})$ more expensive than the optimal k-means/median clustering. However, for D sufficiently large compared to $1/\epsilon$, this example satisfies $(2, \epsilon)$-approximation-stability or even

[5] [Ostrovsky et al. 2006] shows an implication in this direction (Theorem 5.2); however, this implication requires a substantially stronger condition, namely that data satisfy (c, ϵ)-approximation-stability for $c = 1/\epsilon^2$ (and that target clusters be large). In contrast, our primary interest is in the case where c is below the threshold for existence of worst-case approximation algorithms.

$(1/\epsilon, \epsilon)$-approximation-stability (for proof, see Appendix B).

7. SUBSEQUENT WORK

In this section we describe work subsequent to the initial conference publication of our results [Balcan et al. 2009] that has gone on to further expand understanding of the algorithmic implications of approximation-stability, explore relaxations of the approximation-stability condition, and use approximation-stability to develop fast, effective algorithms for clustering biological data. Additionally, we discuss subsequent work exploring other deterministic stability and separation conditions.

7.1 Algorithmic Results under Approximation-Stability

Further guarantees for min-sum clustering. Balcan and Braverman [2009] further analyze the min-sum problem and show how to handle the presence of small target clusters. To achieve this they derive new structural properties implied by $(1+\alpha, \epsilon)$-approximation-stability. In the case where k is small compared to $\log n / \log\log n$ they output a single clustering which is $O(\epsilon/\alpha)$-close to the target, while in the general case their algorithm outputs a small list of clusterings with the property that the target clustering is close to one of those in the list. They further show that if we the target clusters are large (of size at least $100\epsilon n/\alpha^2$), they can reduce the approximation error from $O(\epsilon/\alpha)$ down to $O(\epsilon)$.

Further guarantees for k-median and k-means clustering. Schalekamp et al. [2010] show that Algorithm 1 additionally achieves a good approximation to the k-median objective in the case that target clusters are large. We note that our approximation hardness result for clustering under (c, ϵ)-approximation-stability (which appears as Theorem A.2 in Appendix A) requires the target to have small clusters. They also discuss implementation issues and perform a number of experimental comparisons between various algorithms.

Awasthi et al. [2010] go further and provide a PTAS for k-median, as well as for k-means in Euclidean space, when all target clusters have size $> \epsilon n$ and $\alpha > 0$ is a constant. One implication of this is that when $\alpha > 0$ is a constant, they improve the "largeness" condition needed to efficiently get ϵ-close for k-median from $O((1 + 1/\alpha)\epsilon n)$ to ϵn. Another implication is that they are able to get the same guarantee for k-means as well, when points lie in R^n, improving on the guarantees in Section 4.1 for points in Euclidean space. Note that while α does not appear in the "largeness" condition, their algorithm has running time that depends exponentially on $1/\alpha$, whereas ours does not depend on $1/\alpha$ at all.

Correlation clustering. Balcan and Braverman [2009] also analyze the correlation clustering problem under the (c, ϵ)-approximation-stability assumption. For correlation clustering, the input is a graph with edges labeled $+1$ or -1 and the goal is to find a partition of the nodes that best matches the signs of the edges [Blum et al. 2004]. Usually, two versions of this problem are considered: minimizing disagreements and maximizing agreements. In the former case, the goal is to minimize the number of -1 edges inside clusters plus the number of $+1$ edges between clusters, while in the latter case the goal is to maximize the number of $+1$ edges inside the cluster plus the number of -1 edges between. These are equivalent at optimality but differ in their difficulty of approximation. Balcan and Braverman [2009] show that for the objective of minimizing disagreements, $(1+\alpha,\epsilon)$-approximation-stability implies $(2.5, O(\epsilon/\alpha))$-approximation-stability, so one can use a state-of-the-art 2.5-approximation algorithm for minimizing disagreements in order to achieve an accurate clustering.[6] This contrasts sharply with the case of objectives such as k-median, k-means and min-sum (see Theorem A.1).

7.2 Relaxations of Approximation-Stability

Stability with noise and outliers. Balcan, Roeglin, and Teng [2009] consider a relaxation of (c,ϵ)-approximation-stability that allows for the presence of noisy data—data points for which the (heuristic) distance measure does not reflect cluster membership well—that could cause stability over the full dataset to be violated. Specifically, they define (ν, c, ϵ)-approximation-stability which requires that the data satisfies (c,ϵ)-approximation-stability only after a ν fraction of the data points have been removed. Balcan et al. [2009] show that in the case where the target clusters are large (have size $\Omega((\epsilon/\alpha+\nu)n)$) the large-clusters algorithm we present in this paper can be used to output a clustering that is $(\nu+\epsilon)$-close to the target clustering. They also show that in the more general case there can be multiple significantly different clusterings that can satisfy (ν, c, ϵ)-approximation-stability (since two different sets of outliers could result in two different clusterings satisfying the condition). However, if *most* of the points come from large clusters, they show one can in polynomial time output a small list of k-clusterings such that any clustering that satisfies the property is close to one of the clusterings in the list.

Deletion-stability. Awasthi et al. [2010] consider instances satisfying the condition that deleting any center in the k-median/k-means-optimal solution, and reassigning its points to one of the $k-1$ other centers, raises the objective value by at least a $1+\alpha$ factor. This can be viewed as a relaxation of $(1+\alpha,\epsilon)$-approximation-stability in the event that target clusters have size greater than ϵn (since in that case no solution with $k-1$ clusters can be ϵ-close to the target). It also can be viewed as a relaxation of the condition of [Ostrovsky et al. 2006]. They then show how to obtain a PTAS under this condition when $\alpha > 0$ is a constant. Note, however, that their running time is exponential in $1/\alpha$; in contrast, our algorithms have running times polynomial in n and k and independent of $1/\alpha$.

The Inductive model. Balcan et al. [2009] and Balcan and Braverman [2009] also show how to cluster well under approximation-stability in the *inductive* clustering setting. Here, rather than being given the entire set S up front, the algorithm is provided only a small random sample $\tilde{S}$ of it. Our goal is then to use $\tilde{S}$ to produce a hypothesis $h : X \to Y$ which implicitly represents a clustering of the whole set S and which has low error on it. Balcan et al. [2009] show how in the large clusters case the analysis in our paper can be adapted to the inductive model for k-median and k-means, and Balcan and Braverman [2009] have shown how to adapt their minsum algorithm to the inductive setting as well.

[6] Note that the maximizing agreement version of correlation clustering is less interesting in our framework since it admits a PTAS.

7.3 Practical Application of Approximation-Stability

Motivated by clustering applications in computational biology, Voevodski et al. [2010; 2012] analyze (c, ϵ)-approximation-stability in a model with unknown distance information where one can only make a limited number of *one versus all* queries. They design an algorithm that, assuming (c, ϵ)-approximation-stability for the k-median objective, finds a clustering that is ϵ-close to the target by using only $O(k)$ one-versus-all queries in the large cluster case, and in addition is faster than the algorithm we present here. In particular, the algorithm for the large clusters case we describe in Section 3 can be implemented in $O(|S|^3)$ time, while the one proposed in [Voevodski et al. 2010; 2012] runs in time $O(|S|k(k + \log |S|))$. They then use their algorithm to cluster biological datasets in the Pfam [Finn et al. 2010] and SCOP [Murzin et al. 1995] databases, where the points are proteins and distances are inversely proportional to their sequence similarity. This setting nicely fits the one-versus all queries model because one can use a fast sequence database search program to query a sequence against an entire dataset. The Pfam [Finn et al. 2010] and SCOP [Murzin et al. 1995] databases are used in biology to observe evolutionary relationships between proteins and to find close relatives of particular proteins. Voevodski et al. [2010; 2012] show that their algorithms are not only fast on these datasets, but also achieve high accuracy. In particular, for one of these sources they obtain clusterings that almost exactly match the given classification, and for the other, the accuracy of their algorithm comparable to that of the best known (but slower) algorithms using the full distance matrix.

7.4 Other Deterministic Separation Conditions

There has also been subsequent work exploring the problem of clustering under other deterministic stability and separation conditions.

Bilu and Linial [2010] consider inputs satisfying the condition that the optimal solution to the objective remains optimal even after bounded perturbations to the input weight matrix. They give an algorithm for maxcut (which can be viewed as a 2-clustering problem) under the assumption that the optimal solution is stable to (roughly) $O(n^{2/3})$-factor multiplicative perturbations to the edge weights. Awasthi et al. [2012] consider this condition for center-based clustering objectives such as k-median and k-means, and give an algorithm that finds the optimal solution when the input is stable to only factor-3 perturbations. This factor is improved to $1 + \sqrt{2}$ by Balcan and Liang [2012], who also design algorithms under a relaxed (c, ϵ)-stability to perturbations condition in which the optimal solution need not be identical on the c-perturbed instance, but may change on an ϵ fraction of the points (in this case, the algorithms require $c = 2 + \sqrt{7}$). Note that for the k-median and min-sum objectives, (c, ϵ)-approximation-stability with respect to $\mathcal{C}^*$ implies (c, ϵ)-stability to perturbations because an optimal solution in a c-perturbed instance is guaranteed to be a c-approximation on the original instance;[7] so, (c, ϵ)-stability to perturbations is a weaker condition. Similarly, for k-means, (c, ϵ)-stability to perturbations is implied by (c^2, ϵ)-approximation-stability. However, as noted above, the values of

[7] In particular, a c-perturbed instance $\tilde{d}$ satisfies $d(x, y) \leq \tilde{d}(x, y) \leq cd(x, y)$ for all points x, y. So, using Φ to denote cost in the original instance, $\tilde{\Phi}$ to denote cost in the perturbed instance and using $\tilde{\mathcal{C}}$ to denote the optimal clustering under $\tilde{\Phi}$, we have $\Phi(\tilde{\mathcal{C}}) \leq \tilde{\Phi}(\tilde{\mathcal{C}}) \leq \tilde{\Phi}(\mathcal{C}^*) \leq c\Phi(\mathcal{C}^*)$.

c known to lead to efficient clustering in the case of stability to perturbations are larger than for approximation-stability, where any constant $c > 1$ suffices.

Kumar and Kannan [2010] consider the problem of recovering a target clustering under deterministic separation conditions that are motivated by the k-means objective and by Gaussian and related mixture models. They consider the setting of points in Euclidean space, and show that if the projection of any data point onto the line joining the mean of its cluster in the target clustering to the mean of any other cluster of the target is $\Omega(k)$ standard deviations closer to its own mean than the other mean, then they can recover the target clusters in polynomial time. This condition was further analyzed and reduced by work of Awasthi and Sheffet [2012]. This separation condition is formally incomparable to approximation-stability (even restricting to the case of k-means with points in Euclidean space). In particular, if the dimension is low and k is large compared to $1/\epsilon$, then this condition can require more separation than approximation-stability (e.g., with k well-spaced clusters of unit radius, similar to the example of Appendix B, approximation-stability would require separation only $O(1/\epsilon)$ and independent of k). On the other hand if the clusters are high-dimensional, then this condition can require less separation than approximation-stability since the ratio of projected distances will be more pronounced than the ratios of distances in the original space.

8. CONCLUSIONS AND OPEN QUESTIONS

8.1 Discussion

The main motivation for this work is that for many unsupervised-learning clustering problems, such as clustering proteins by function or clustering images by subject, the true goal is to partition the points correctly—e.g., to produce a clustering in which proteins are correctly clustered by their function, or all images by who is in them. However, since accuracy typically cannot be measured directly by the clustering algorithm, distance-based objectives such as k-median, k-means, or min-sum are used instead as measurable proxies for this goal.[8] Usually, these distance-based objectives are studied in isolation, for an arbitrary point set, with upper and lower bounds proven on their approximability. In this work, we consider instead the implications of studying them along with the underlying accuracy goal. What our results show is that if we consider the natural inductive bias that would motivate use of a c-approximation algorithm for such problems, namely (c, ϵ)-approximation-stability, we can use it to achieve a clustering of error $O(\epsilon)$ even if we do not have a c-approximation algorithm for the associated objective: in fact, even if achieving a c-approximation is NP-hard. In particular, for the case of the k-median, k-means, and min-sum objectives, we can achieve a low-error clustering on any instance satisfying (c, ϵ)-approximation-stability for any constant $c > 1$ (if additionally the target clusters are large in the case of min-sum).

From the perspective of approximation algorithms, this work suggests a new avenue for making progress in the face of approximation-hardness barriers for prob-

[8] This is similar to the way quantities such as hinge-loss are often used as surrogate losses for error rate or 0-1 loss in the context of *supervised* learning—except that for supervised learning, this is done to make the computational problem more tractable, whereas in the case of clustering it is done because the underlying accuracy goal cannot be directly measured.

lems where the given objective may be a proxy for an underlying accuracy goal. In particular, an appealing aspect of (c, ϵ)-approximation-stability is that it is a property that one would hope to hold in any event for such problems when using an approximation algorithm. That is because if the given instance does not satisfy this condition, then achieving a c-approximation is, by itself, insufficient to ensure producing a desirable solution for that instance. So, an algorithm that guarantees low-error solutions under (c, ϵ)-approximation-stability can be said to perform nearly as well on such problems *as if* one had a generic c-approximation, and as we show, this may be achievable even when achieving a c-approximation is NP-hard. In particular, since we achieve this guarantee for any constant $c > 1$, this means that our performance guarantee in terms of accuracy is nearly as good as if we had a generic PTAS for the k-median, k-means, and min-sum objectives.

Approximation-stability additionally motivates algorithms with interesting and useful properties. For example, it motivates outlier-resilient re-clustering of data as in Algorithm 1 (Section 3) as well as algorithms that aim to explicitly identify and output "I don't know" on outliers in order to achieve especially low error on the remainder (e.g., Theorem 4.4, Section 4). Furthermore, as with approximation ratio, approximation stability can provide a useful and convenient guide for algorithm design in novel data clustering scenarios. For example, as discussed in Section 7, Voevodski et al. [2010] consider the problem of clustering biological datasets in which only limited distance information can be obtained. They find that algorithms designed for approximation-stability—in fact, a variant of the algorithm that we propose for the k-median problem—yield fast and highly accurate results.

8.2 Open questions

One natural open question is whether the $O(\epsilon/\alpha)$ form of the bounds we achieve are intrinsic, or if improved bounds for these objectives are possible. For example, suppose our instance satisfies $(1 + \epsilon, \epsilon)$-approximation-stability for *all* $\epsilon > 0$, say for k-median (e.g., achieving a 1.01-approximation would produce a solution of error 1%, a 1.001-approximation gives error 0.1%, etc.); is such an assumption sufficient to produce a near-optimal solution of some form, either in terms of approximation or in terms of low error? (Note that directly applying our results for $(1 + \alpha, \epsilon)$-approximation-stability yields nothing useful in terms of low error, since our closeness guarantee of $O(\epsilon/\alpha)$ becomes greater than 1 when $\alpha = \epsilon$.) Another natural question is whether one can use this approach for other clustering or partitioning objective functions. For example, the *sparsest cut* problem has been the subject of a substantial body of research, with the best known approximation guarantee a factor of $O(\sqrt{\log n})$ [Arora et al. 2004]. However, in the event this objective is a proxy for a true goal of partitioning a dataset in a nearly-correct manner, it is again natural to consider data satisfying (c, ϵ)-approximation-stability. In this case, given the current state of approximation results, it would be of interest even if c is a large constant. See [Balcan 2009] for more details. The *max-cut* problem would also be of interest, for values c closer to 1 than the Goemans-Williamson bound [Goemans and Williamson 1995] (defining approximation-stability appropriately for maximization problems).

More broadly, there are other types of problems, such as evolutionary tree reconstruction, where the measurable objectives typically examined may again only be a proxy for the true goals, e.g., to produce a correct evolutionary tree. It would be interesting to examine whether the approach developed here might be of use in those settings as well.

Acknowledgments. We would like to thank Leonard Schulman and the anonymous referees for their helpful comments. Author M.-F. Balcan was supported in part by NSF grant CCF-0953192, a Microsoft Faculty Fellowship, and a Google Research Grant. Author A. Blum was supported in part by NSF grants CCF-0514922 and CCF-0830540 and by a Google Research Grant. Author A. Gupta was supported in part by NSF grants CCF-0448095 and CCF-0729022, and an Alfred P. Sloan Fellowship.

REFERENCES

Achlioptas, D. and McSherry, F. 2005. On spectral learning of mixtures of distributions. In *Proceedings of the Eighteenth Annual Conference on Learning Theory.*

Agarwal, P. and Mustafa, N. 2004. k-means projective clustering. In *Proceedings of the 23rd Annual Symposium on Principles of Database Systems.*

Alon, N., Dar, S., Parnas, M., and Ron, D. 2000. Testing of clustering. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science.*

Arora, S. and Kannan, R. 2005. Learning mixtures of arbitrary gaussians. In *Proceedings of the 37th ACM Symposium on Theory of Computing.*

Arora, S., Raghavan, P., and Rao, S. 1999. Approximation schemes for Euclidean k-medians and related problems. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing.*

Arora, S., Rao, S., and Vazirani, U. 2004. Expander flows, geometric embeddings, and graph partitioning. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing.*

Arthur, D. and Vassilvitskii, S. 2007. k-means++: the advantages of careful seeding. In *SODA*. 1027–1035.

Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., and Pandit., V. 2004. Local search heuristics for k-median and facility location problems. *SIAM J. Comput. 33,* 3, 544–562.

Awasthi, P., Blum, A., and Sheffet, O. 2010. Stability yields a PTAS for k-median and k-means clustering. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science.*

Awasthi, P., Blum, A., and Sheffet, O. 2012. Center-based clustering under perturbation stability. *Inf. Process. Lett. 112,* 1-2, 49–54.

Awasthi, P. and Sheffet, O. 2012. Improved spectral-norm bounds for clustering. In *APPROX-RANDOM*. 37–49.

Balcan, M. 2009. Better guarantees for sparsest cut clustering. In *Proceedings of the 22nd Annual Conference on Learning Theory.*

Balcan, M., Blum, A., and Gupta, A. 2009. Approximate clustering without the approximation. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms.*

Balcan, M., Blum, A., and Vempala, S. 2008. A discrimantive framework for clustering via similarity functions. In *Proceedings of the 40th ACM Symposium on Theory of Computing.*

Balcan, M. and Braverman, M. 2009. Finding low error clusterings. In *Proceedings of the 22nd Annual Conference on Learning Theory.*

Balcan, M., Roeglin, H., and Teng, S. 2009. Agnostic clustering. In *Proceedings of the 20th International Conference on Algorithmic Learning Theory.*

Balcan, M.-F. and Liang, Y. 2012. Clustering under perturbation resilience. In *ICALP (1)*. 63–74.

Bartal, Y., Charikar, M., and Raz, D. 2001. Approximating min-sum k-clustering in metric spaces. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing.*

Belkin, M. and Sinha, K. 2010. Polynomial learning of distribution families. In *FOCS*. 103–112.

Bilu, Y. and Linial, N. 2010. Are stable instances easy? In *Proceedings of the First Symposium on Innovations in Computer Science*.

Blum, A., Bansal, N., and Chawla, S. 2004. Correlation clustering. *Machine Learning 56*, 89–113.

Brohée, S. and van Helden, J. 2006. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics 7*, 488.

Charikar, M. and Guha, S. 1999. Improved combinatorial algorithms for the facility location and k-median problems. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*.

Charikar, M., Guha, S., Tardos, E., and Shmoy, D. B. 1999. A constant-factor approximation algorithm for the k-median problem. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*.

Czumaj, A. and Sohler, C. 2007. Small space representations for metric min-sum k-clustering and their applications. In *Proceedings of the 24th International Symposium on Theoretical Aspects of Computer Science*.

Dasgupta, S. 1999. Learning mixtures of gaussians. In *Proceedings of The 40th Annual Symposium on Foundations of Computer Science*.

de la Vega, W. F., Karpinski, M., Kenyon, C., and Rabani, Y. 2003. Approximation schemes for clustering problems. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*.

Devroye, L., Gyorfi, L., and Lugosi, G. 1996. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag.

Duda, R. O., Hart, P. E., and Stork, D. G. 2001. *Pattern Classification*. Wiley.

Feige, U. 1998. A threshold of $\ln n$ for approximating set cover. *J. ACM 45,* 4, 634–652.

Finn, R., Mistry, J., Tate, J., Coggill, P., andJ.E. Pollington, A. H., Gavin, O., Gunesekaran, P., Ceric, G., Forslund, K., Holm, L., Sonnhammer, E., Eddy, S., and Bateman, A. 2010. The pfam protein families database. *Nucleic Acids Research 38*, D211–222.

Goemans, M. X. and Williamson, D. P. 1995. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach. 42,* 6, 1115–1145.

Gollapudi, S., Kumar, R., and Sivakumar, D. 2006. Programmable clustering. In *Proceedings of the 25th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*.

Guha, S. and Khuller, S. 1999. Greedy strikes back: Improved algorithms for facility location. *Journal of Algorithms 31,* 1, 228–248.

Indyk, P. 1999. Sublinear time algorithms for metric space problems. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*.

Jain, K., Mahdian, M., and Saberi, A. 2002. A new greedy approach for facility location problems. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*.

Jain, K. and Vazirani, V. V. 2001. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *JACM 48,* 2, 274 – 296.

Kannan, R., Salmasian, H., and Vempala, S. 2005. The spectral method for general mixture models. In *Proceedings of The Eighteenth Annual Conference on Learning Theory*.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. 2004. A local search approximation algorithm for k-means clustering. *Comput. Geom. 28,* 2-3, 89–112.

Kleinberg, J. 2002. An impossibility theorem for clustering. In *Proceedings of the Neural Information Processing Systems*.

Kumar, A. and Kannan, R. 2010. Clustering with spectral norm and the k-means algorithm. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*.

Kumar, A., Sabharwal, Y., and Sen, S. 2004. A simple linear time $(1+\epsilon)$-approximation algorithm for k-means clustering in any dimensions. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*. Washington, DC, USA.

Lloyd, S. 1982. Least squares quantization in PCM. *IEEE Trans. Inform. Theory 28,* 2, 129–137.

Mahajan, M., Nimbhorkar, P., and Varadarajan, K. 2009. The planar k-means problem is np-hard. In *WALCOM*. 274–285.

Manning, C., Raghavan, P., and Schütze, H. 2008. *Introduction to information retrieval.* Cambridge University Press.

Megiddo, N. and Supowit, K. 1984. On the complexity of some common geometric location problems. *SIAM J. Comput. 13,* 1, 182–196.

Meila, M. 2003. Comparing clusterings by the variation of information. In *Proceedings of the The Sixteenth Annual Conference on Learning Theory.*

Meila, M. 2005. Comparing clusterings – an axiomatic view. In *International Conference on Machine Learning.*

Meila, M. 2006. The uniqueness of a good clustering for k-means. In *Proceedings of the 23rd International Conference on Machine Learning.*

Meila, M. 2012. Local equivalences of distances between clusterings - a geometric perspective. *Machine Learning 86,* 3, 369–389.

Moitra, A. and Valiant, G. 2010. Settling the polynomial learnability of mixtures of gaussians. In *FOCS*. 93–102.

Murzin, A., Brenner, S. E., Hubbard, T., and Chothia, C. 1995. Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology 247*, 536–540.

Ostrovsky, R., Rabani, Y., Schulman, L., and Swamy, C. 2006. The effectiveness of lloyd-type methods for the k-means problem. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science.*

Sahni, S. and Gonzalez, T. 1976. P-complete approximation problems. *J. ACM 23,* 3, 555–565.

Schalekamp, F., Yu, M., and van Zuylen, A. 2010. Clustering with or without the approximation. In *Proceedings of the 16th Annual International Computing and Combinatorics Conference.*

Schulman, L. 2000. Clustering for edge-cost minimization. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing.* 547–555.

Vempala, S. and Wang, G. 2004. A spectral algorithm for learning mixture models. *JCSS 68,* 2, 841–860.

Voevodski, K., Balcan, M., Roeglin, H., Teng, S., and Xia, Y. 2010. Efficient clustering with limited distance information. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence.*

Voevodski, K., Balcan, M., Roeglin, H., Teng, S., and Xia, Y. 2012. Active clustering of biological sequences. *Journal of Machine Learning Research*.

Yan, D., Huang, L., and Jordan, M. I. 2009. Fast approximate spectral clustering. In *KDD*. 907–916.

A. ADDITIONAL PROOFS

Theorem A.1. *For any $1 \leq c_1 < c_2$, any $\epsilon, \delta > 0$, for sufficiently large k, there exists a family of metric spaces G and target clusterings that satisfy (c_1, ϵ)-approximation-stability for the k-median objective (likewise, k-means and min-sum) and yet do not satisfy even $(c_2, 1/2 - \delta)$-approximation-stability for that objective.*

Proof. We focus first on the k-median objective. Consider a set of n points such that the target clustering consists of one cluster C_1 with $n(1-2\delta)$ points and $k-1$ clusters $C_2, \ldots, C_k$ each with $\frac{2\delta n}{k-1}$ points. All points in the same cluster have distance 1. The distance between points in any two distinct clusters C_i, C_j

for $i, j \geq 2$ is D, where $D > 1$ will be defined below. Points in C_1 are at distance greater than $c_2 n$ from any of the other clusters.

In this construction, the target clustering *is* the optimal k-median solution, and has a total k-median cost of $n-k$. We now define D so that there (just barely) exists a c_2 approximation that splits cluster C_1. In particular, consider the solution that merges C_2 and C_3 into a single cluster ($C_4, \ldots, C_k$ will each be their own cluster) and uses 2 clusters to evenly split C_1. This clearly has error at least $1/2 - \delta$, and furthermore this solution has a cost of $(\frac{2\delta n}{k-1})(D-1) + n - k$, and we define D to set this equal to $c_2(n-k)) = c_2\text{OPT}$.

Any c_1 approximation, however, must be ϵ-close to the target for $k > 1 + 2\delta/\epsilon$. In particular, by definition of D, any c_1-approximation must have one median inside each C_i. Therefore, it cannot place two medians inside C_1 as in the above c_2-approximation, and so can have error on fewer than $\frac{2\delta n}{k-1}$ points. This is less than ϵn by definition of k.

The same construction, with D defined appropriately, applies to k-means as well. In particular, we just define D to be the square-root of the value used for D above, and the entire argument proceeds as before.

For min-sum, we modify the construction so that distances in C_1 are all equal to 0, so now $\text{OPT} = (k-1)(\frac{2\delta n}{k-1})(\frac{2\delta n}{k-1} - 1)$. Furthermore, we set points in C_1 to be at distance greater than $c_2\text{OPT}$ from all other points. We again define D so that the cheapest way to use $k-2$ clusters for the points in $C_2 \cup \ldots \cup C_k$ has cost exactly $c_2\text{OPT}$. However, because of the pairwise nature of the min-sum objective, this is now to equally distribute the points in one of the clusters $C_2, \ldots, C_k$ among all the others. This has cost $2(\frac{2\delta n}{k-1})^2 D + \text{OPT} - (\frac{2\delta n}{k-1})^2(\frac{k-3}{k-2})$, which as mentioned above we set to $c_2\text{OPT}$. Again, because we have defined D such that the cheapest clustering of $C_2 \cup \ldots \cup C_k$ using $k-2$ clusters has cost $c_2\text{OPT}$, any c_1 approximation must use $k-1$ clusters for these points and therefore again must have error less than $\frac{2\delta n}{k-1} < \epsilon n$. □

Theorem A.2. *For k-median, k-means, and min-sum objectives, for any $c > 1$, the problem of finding a c-approximation can be reduced in polynomial time to the problem of finding a c-approximation under (c, ϵ)-approximation-stability. Therefore, a polynomial-time algorithm for finding a c-approximation under (c, ϵ)-approximation-stability implies a polynomial-time algorithm for finding a c-approximation in general.*

Proof. Given a metric G with n nodes and a value k (a generic instance of the clustering problem) we construct a new instance that is (c, ϵ)-approximation-stable. In particular we create a new graph G' by adding an extra n/ϵ nodes that are all at distance D from each other and from the nodes in G, where D is chosen to be larger than $c\text{OPT}$ on G (e.g., D could be the sum of all pairwise distances in G). We now let $k' = k + n/\epsilon$ and define the target clustering to be the optimal (k-median, k-means, or min-sum) solution on G, together with each of the points in $G' \setminus G$ in its own singleton cluster.

We first claim that G' satisfies (c, ϵ)-approximation-stability. This is because, by definition of D, any solution that does not put each of the new nodes into its own singleton cluster will incur too high a cost to be a c-approximation. So a c-approximation can only differ from the target on G (which has less than an ϵ fraction of the nodes). Furthermore, a c-approximation in G' yields a c-approximation in G because the singleton clusters do not contribute to the overall cost in any of the k-median, k-means, or min-sum objectives. □

The following shows that unlike $(1.01, \epsilon)$-approximation-stability, obtaining an $O(\epsilon)$-close clustering is NP-hard under $(1, \epsilon)$-approximation-stability.

THEOREM A.3. *For any contant c', for any $\epsilon < 1/(ec')$, it is NP-hard to find a clustering of error at most $c'\epsilon$ for the k-median and k-means problem under $(1, \epsilon)$-approximation-stability.*

PROOF. First, let us prove a $(1 + 1/e)$-hardness for instances of k-median where one is allowed to place centers at any point in the metric space. The proof is very similar to the proof from [Guha and Khuller 1999; Jain et al. 2002] which gives a $(1 + 2/e)$-hardness for the case where one can place centers only at a distinguished set of locations in the metric space. We then show how to alter this hardness result to prove the theorem.

Consider the max-k-coverage problem with n elements and m sets: that is, given m subsets of a universe of n elements, find k sets whose union covers as many elements as possible. It is NP-hard to distinguish between instances of this problem where there exist k sets that can cover all the elements, and instances where any k sets cover only $(1 - 1/e)$-fraction of the elements [Feige 1998]. The hard instances have the property that both m and k are a tiny fraction of the number of elements n. For some suitably large constant C, we construct an instance of k-median with $cn+m$ points, one point for each set and c points for each element, assign distance 1 between any two points such that one of them represents an element and the other a set containing that point, and distance 2 to all other pairs.

Note that if there are k sets in the set system that cover all the elements (the "yes" instances), choosing the corresponding k points as centers gives us a solution of cost $cn + 2(m - k) \leq (1 + \delta)cn$ for some arbitrarily small constant $\delta > 0$. On the other hand, given any solution to k-median with cost C, if any of these centers are on points corresponding to elements, we can choose a set-point at unit distance from it instead, thus potentially increasing the cost of the solution by at most m to $C + m$. Hence, if this collection of k sets covers at most $(1 - 1/e)$ fraction of the elements (as in a "no" instance of max-k-coverage), the cost of this solution would be at least $(1 - 1/e)cn + 2/ecn + 2(m - k) = (1 + 1/e)cn + 2(m - k)$; hence C would be at least $(1 + 1/e - \delta)cn$ in this case. This shows that for every δ, there are instances of k-median whose optimal cost is either at most C or at least $(1 + 1/e - \delta)$, such that distinguishing between these two cases is NP-hard.

Let us now add infinitesimal noise to the above instances of k-median to make a unique optimal solution and call this the target; the uniqueness of the optimal solution ensures that we satisfy $(1, \epsilon)$-approximation-stability without changing the hardness significantly. Now, in the "yes" case, any clustering with error $c'\epsilon$ will have cost at most $(1 - c'\epsilon)cn + 2c'\epsilon cn + 2(m - k) \leq (1 + c'\epsilon + \delta)cn$. This is less than the cost of the optimal solution in the "no" case (which is still at least $(1 + 1/e - \delta)cn$) as long as $c'\epsilon \leq 1/e - 2\delta$, and would allow us to distinguish the "yes" and "no" instances. This completes the proof for the k-median case, and the proof can be altered slightly to work for the k-means problem as well. □

A.1 Proof of the Reassignment Lemma

We now prove Lemma 2.2, which we restate here for convenience.

LEMMA 2.2. *Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a k-clustering in which each cluster is nonempty, and let $R = \{(x_1, j_1), (x_2, j_2), \ldots, (x_t, j_t)\}$ be a set of t reassignments of points x_i to clusters C_{j_i} (assume that $x_i \notin C_{j_i}$ for all i). Then there must exist a set $R' \subseteq R$ of size at least $t/3$ such that the clustering $\mathcal{C}'$ produced by reassigning points in R' has distance exactly $\frac{1}{n}|R'|$ from $\mathcal{C}$.*

Note: Before proving the lemma, note that we cannot necessarily just choose $R' = R$ because, for instance, it could be that R moves all points in C_1 to C_2 and all points in C_2 to C_1: in this case, performing all reassignments in R produces the exact same clustering as we started with (just with different indices). Instead, we need to ensure that each reassignment in R' has an associated certificate ensuring that if implemented, it will increase the resulting distance from $\mathcal{C}$. Note also that if $\mathcal{C}$ consists of 3 singleton clusters: $C_1 = \{x\}, C_2 = \{y\}, C_3 = \{z\}$, and if $R = \{(x, 2), (y, 3), (z, 1)\}$, then any subset of reassignments in R will produce a clustering that differs in at most one element from $\mathcal{C}$; thus, the factor of 3 is tight.

Notation. Given a clustering $\mathcal{C}$ and a point x, let $C(x)$ denote the cluster $C_i \in \mathcal{C}$ such that $x \in C_i$.

PROOF. The proof is based on the following lower-bounding technique. Given two clusterings $\mathcal{C}$ and $\mathcal{C}'$, suppose we can produce a list L of disjoint subsets of points $S_1, S_2, \ldots$, such that for each i, all points in S_i are in the *same* cluster in one of $\mathcal{C}$ or $\mathcal{C}'$ and they are all in *different* clusters in the other. Then $\mathcal{C}$ and $\mathcal{C}'$ must have distance at least $\frac{1}{n}\sum_i(|S_i| - 1)$. In particular, any bijection σ on the indices can have agreement between $\mathcal{C}$ and $\mathcal{C}'$ on at most one point from each S_i.

A simpler factor-8 argument: We begin for illustration with a simpler factor-8 argument. For this argument we consider two cases. First, suppose that at least half of the reassignments in R involve points x in clusters of size ≥ 2. In this case, we simply do the following. While there exists some $(x, j) \in R$ such that $|C(x)| \geq 2$, choose some arbitrary point $y \in C(x)$ and add $\{x, y\}$ to L, add (x, j) to R', and then remove (x, j) from R as well as any reassignment of y if one exists; also remove both x and y from the overall point set S. If cluster $C(x)$ has been reduced to a singleton $\{z\}$, then remove z from R and S as well. This process guarantees that all pairs added to L are disjoint, and we remove at most three times as many reassignments from R as we add to R' (one for x, at most one for y, and at most one for z). Thus, since we assumed at least half of R came from clusters of size at least 2, overall we get a factor of 6. The second case is that at least half of the reassignments in R involve points x in clusters of size 1. In that case, randomly color each cluster red or blue: in expectation, 1/4 of these reassignments (at least 1/8 of the total in R) go from red clusters to blue clusters. We now simply put all of these reassignments, namely those involving points moving from singleton red clusters to blue clusters, into R'. Because all such (x, j) for any given j involve different source clusters, for each j such that R' contains at least one pair (x, j), we pick an arbitrary $y \in C_j$ and put in L the witness set $S_j = \{x : (x, j) \in R'\} \cup \{y\}$. All points in S_j were in different clusters in $\mathcal{C}$ and are in the same cluster in $\mathcal{C}'$, and the sets $S_j, S_{j'}$ for $j \neq j'$ are disjoint, so L is a legitimate witness set. Furthermore,

$\sum_j(|S_j|-1) = |R'|$ as desired.

The factor-3 argument: For the factor-3 argument, we begin constructing R' and L as in the first case above, but using only clusters of size at least 3. Specifically, while there exists a reassignment $(x,j) \in R$ such that x is in a cluster $C(x)$ with at least 3 points: choose an arbitrary point $y \in C(x)$ and add $\{x,y\}$ to L, add (x,j) to R', and remove (x,j) from R as well as any reassignment of y if one exists. In addition, remove x and y from the point set S. This process guarantees that all pairs added to L are disjoint, and we remove at most twice as many reassignments from R as we add to R'. (So, if R becomes empty, we will have achieved our desired result with $|R'| = t/2$). Moreover, because we only perform this step if $|C(x)| \geq 3$, this process does not produce any empty clusters.

We now have that for all reassignments $(x,j) \in R$, x is in a singleton or doubleton cluster. Let R_{single} be the set of reassignments $(x,j) \in R$ such that x is in a singleton cluster. Viewing these reassignments as directed edges, R_{single} forms a graph on the clusters C_i where each node has outdegree ≤ 1. Therefore, each component of this graph must be an arborescence with possibly one additional edge from the root. We now proceed as follows. While R_{single} contains a source (a node of outdegree 1 and indegree 0), choose an edge (x,j) such that (a) x is a source and (b) for all other edges (y,j), y is either a source or part of a cycle. We then consider two cases:

(1) Node j is not a sink in R_{single}: that is, there exists an edge $(z,j_z) \in R_{single}$ for $z \in C_j$. In this case, we add to R' the edge (x,j) and all other edges (y,j) such that y is a source, and we remove from R (and from R_{single}) the edges (z,j_z), (x,j), and all edges (y,j) (including the at most one edge (y,j) such that y is part of a cycle). We then add to L the set $\{x\} \cup \{z\} \cup \{y : (y,j) \text{ was just added to } R'\}$ and remove these points from S. Note that the number of edges removed from R is at most the number of edges added to R' plus 2, giving a factor of 3 in the worst case. Note also that we maintain the invariant that no edges in R_{single} point to empty clusters, since we deleted all edges into C_j, and the points x and y added to L were sources in R_{single}.

(2) Otherwise, node j is a sink in R_{single}. In this case, we add to R' the edge (x,j) along with all other edges $(y,j) \in R_{single}$ (removing those edges from R and R_{single}). We choose an arbitrary point $z \in C_j$ and add to L the set $\{x\} \cup \{z\} \cup \{y : (y,j) \text{ was just added to } R'\}$, removing those points from S. In addition, we remove from R all (at most two) edges exiting from C_j (we are forced to remove any edge exiting from z since z was added to L, and there might be up to one more edge if C_j is a doubleton). Again, the number of edges removed from R is at most the number of edges added to R' plus 2, giving a factor of 3 in the worst case.

At this point, if R_{single} is nonempty, its induced graph must be a collection of disjoint cycles. For each such cycle, we choose every other edge (half the edges in an even-length cycle, at least 1/3 of the edges in an odd cycle), and for each edge (x,j) selected, we add (x,j) to R', remove (x,j) and (z,j_z) for $z \in C_j$ from R and R_{single}, and add the pair $\{x,z\}$ to L.

Finally, R_{single} is empty and we finish off any remaining doubleton clusters using the same procedure as in the first part of the argument. Namely, while there exists a reassignment $(x, j) \in R$, choose an arbitrary point $y \in C(x)$ and add $\{x, y\}$ to L, add (x, j) to R', and remove (x, j) from R as well as any reassignment involving y if one exists.

By construction, the set R' has size at least $|R|/3$, and the set L ensures that each reassignment in R' increases the resulting distance from $\mathcal{C}$ as desired. $\square$

B. ANALYSIS OF EXAMPLE IN SECTION 6

In Section 6, an example is presented of $\sqrt{n}$ clusters of $\sqrt{n}$ points each, with distance 1 between points in the same target cluster, and distance $D + 1$ between points in different target clusters. We prove here that for any $\epsilon < 1/2$, this satisfies $(D\epsilon/2, \epsilon)$-approximation-stability for both k-median and k-means objectives. Thus, if $D > 4/\epsilon$, then this is $(2, \epsilon)$-approximation-stable.

Let $\mathcal{C}$ be a clustering of distance at least ϵ from the target clustering $\mathcal{C}_T = \mathcal{C}^*$. Since $\mathcal{C}^*$ has both k-median and k-means cost equal to $n - \sqrt{n}$, we need to show that $\mathcal{C}$ has k-median cost at least $(D\epsilon/2)(n - \sqrt{n})$ (its k-means cost can only be larger).

We do this as follows. First, define the "non-permutation distance" from $\mathcal{C}$ to $\mathcal{C}^*$ as $npdist(\mathcal{C}, \mathcal{C}^*) = \frac{1}{n}\sum_{i=1}^{k} \min_j |C_i - C_j^*|$. That is, we remove the restriction that different clusters in $\mathcal{C}$ cannot be mapped to the same cluster in $\mathcal{C}^*$. This is non-symmetric, but clearly satisfies the condition that $npdist(\mathcal{C}, \mathcal{C}^*) \leq dist(\mathcal{C}, \mathcal{C}^*)$. We observe now that the k-median cost of $\mathcal{C}$ is equal to $Dn \cdot npdist(\mathcal{C}, \mathcal{C}^*) + (n - \sqrt{n})$. In particular, the optimal median for each cluster C_i in $\mathcal{C}$ is a point in whichever cluster C_j^* of $\mathcal{C}^*$ has the largest intersection with C_i. This causes each point in $C_i - C_j^*$ to incur an additional cost of D over its cost in $\mathcal{C}^*$, and so the overall increase over the cost of $\mathcal{C}^*$ is $Dn \cdot npdist(\mathcal{C}, \mathcal{C}^*)$. Thus, it remains just to show that $npdist(\mathcal{C}, \mathcal{C}^*)$ cannot be too much smaller than $dist(\mathcal{C}, \mathcal{C}^*)$.

We now show that $npdist(\mathcal{C}, \mathcal{C}^*) \geq dist(\mathcal{C}, \mathcal{C}^*)/2$. We note that this will rely heavily on the fact that all clusters in $\mathcal{C}^*$ have the same size: if C^* contained clusters of very different sizes, the statement would be false. Since this inequality may be of interest more generally (it is not specific to this example), we formalize it in Lemma B.1 below.

LEMMA B.1. *For any clustering $\mathcal{C}$, if all clusters of $\mathcal{C}^*$ have size n/k, then we have $npdist(\mathcal{C}, \mathcal{C}^*) \geq dist(\mathcal{C}, \mathcal{C}^*)/2$.*

PROOF. Let $p_i = |C_i|/n$ and $p = (p_1, \ldots, p_k)$. Let $u = (1/k, \ldots, 1/k)$ and define $\Delta(p, u) = \sum_{i:p_i > u_i} p_i - u_i$ to be the variation distance between p and u. Then, $npdist(\mathcal{C}, \mathcal{C}^*) \geq \Delta(p, u)$ because a cluster in $\mathcal{C}$ of size $p_i n > n/k$ contributes at least $p_i - 1/k$ to the non-permutation distance. Let $\Delta_i = \max(1/k - p_i, 0)$. Since variation distance is symmetric, we have $\Delta(p, u) = \sum_i \Delta_i$.

Now, fix some mapping of clusters C_i to clusters C_j^* yielding the non-permutation distance from $\mathcal{C}$ to $\mathcal{C}^*$. Let T_j denote the set of indices i such that C_i is mapped to C_j^* and let $t_j = |T_j|$. Let S denote the set of indices j such that $t_j \geq 2$ (if this were a permutation then S would be empty). We can now lower-bound the non-permutation distance as

$$\begin{aligned} \mathit{npdist}(\mathcal{C},\mathcal{C}^*) &\geq \sum_{j\in S}\left[\left(\sum_{i\in T_j} p_i\right) - 1/k\right] \\ &\geq \sum_{j\in S}\left[\frac{t_j-1}{k} - \sum_{i\in T_j}\Delta_i\right] \\ &\geq \left(\sum_{j\in S}\frac{t_j-1}{k}\right) - \Delta(p,u). \end{aligned}$$

Therefore, we have

$$\mathit{npdist}(\mathcal{C},\mathcal{C}^*) + \Delta(p,u) \geq \sum_{j\in S}\frac{t_j-1}{k}. \tag{B.1}$$

We now claim we can convert this mapping into a permutation without increasing the distance by too much. Specifically, for each j such that $t_j \geq 2$, keep only the $i \in T_j$ such that C_i has highest overlap with C_j^* and assign the rest to (arbitrary) unmatched target clusters. This reassignment can increase the distance computation by at most $\frac{1}{k}(\frac{t_j-1}{t_j}) \leq \frac{1}{k}(\frac{t_j-1}{2})$. Therefore, we have

$$\mathit{dist}(\mathcal{C},\mathcal{C}^*) \leq \mathit{npdist}(\mathcal{C},\mathcal{C}^*) + \sum_{j\in S}\frac{t_j-1}{2k}. \tag{B.2}$$

Combining (B.1) and (B.2) we have $\mathit{dist}(\mathcal{C},\mathcal{C}^*) - \mathit{npdist}(\mathcal{C},\mathcal{C}^*) \leq \frac{1}{2}(\mathit{npdist}(\mathcal{C},\mathcal{C}^*) + \Delta(p,u))$, and since $\Delta(p,u) \leq \mathit{npdist}(\mathcal{C},\mathcal{C}^*)$, this yields $\mathit{dist}(\mathcal{C},\mathcal{C}^*) \leq 2\mathit{npdist}(\mathcal{C},\mathcal{C}^*)$ as desired. □

Finally, as noted above, by Lemma B.1 we have that the cost of clustering $\mathcal{C}$ in the construction is at least $Dn\epsilon/2$ as desired.