Robust Communication-Optimal Distributed Clustering Algorithms

Pranjal Awasthi
Rutgers University, Piscataway, NJ, USA
pranjal.awasthi@rutgers.edu

Ainesh Bakshi
Carnegie Mellon University, Pittsburgh, PA, USA
abakshi@cs.cmu.edu

Maria-Florina Balcan
Carnegie Mellon University, Pittsburgh, PA, USA
ninamf@cs.cmu.edu

Colin White
Carnegie Mellon University, Pittsburgh, PA, USA
crwhite@cs.cmu.edu

David P. Woodruff
Carnegie Mellon University, Pittsburgh, PA, USA
dwoodruf@cs.cmu.edu

Abstract

In this work, we study the k-median and k-means clustering problems when the data is distributed across many servers and can contain outliers. While there has been a lot of work on these problems for worst-case instances, we focus on gaining a finer understanding through the lens of beyond worst-case analysis. Our main motivation is the following: for many applications such as clustering proteins by function or clustering communities in a social network, there is some unknown target clustering, and the hope is that running a k-median or k-means algorithm will produce clusterings which are close to matching the target clustering. Worst-case results can guarantee constant factor approximations to the optimal k-median or k-means objective value, but not closeness to the target clustering.

Our first result is a distributed algorithm which returns a near-optimal clustering assuming a natural notion of stability, namely, *approximation stability* [12], even when a constant fraction of the data are outliers. The communication complexity is $\tilde{O}(sk+z)$ where s is the number of machines, k is the number of clusters, and z is the number of outliers. Next, we show this amount of communication cannot be improved even in the setting when the input satisfies various non-worst-case assumptions. We give a matching $\Omega(sk+z)$ lower bound on the communication required both for approximating the optimal k-means or k-median cost up to any constant, and for returning a clustering that is close to the target clustering in Hamming distance. These lower bounds hold even when the data satisfies approximation stability or other common notions of stability, and the cluster sizes are balanced. Therefore, $\Omega(sk+z)$ is a communication bottleneck, even for real-world instances.

2012 ACM Subject Classification Theory of computation → Unsupervised learning and clustering

Keywords and phrases robust distributed clustering, communication complexity

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.18

Category Track A: Algorithms, Complexity and Games

Related Version A full version of the paper is available at `https://arxiv.org/abs/1703.00830`.

Acknowledgements This work was supported in part by NSF grants CCF-1422910, CCF-1535967, IIS-1618714, an Office of Naval Research (ONR) grant N00014-18-1-2562, an Amazon Research Award, a Microsoft Research Faculty Fellowship, and a National Defense Science & Engineering Graduate (NDSEG) fellowship. Part of this work was done while Ainesh Bakshi and David Woodruff were visiting the Simons Institute for the Theory of Computing.

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).
Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;
Article No. 18; pp. 18:1–18:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Clustering is a fundamental problem in machine learning with applications in many areas including computer vision, text analysis, bioinformatics, and so on. The underlying goal is to group a given set of points to maximize similarity inside a group and dissimilarity among groups. A common approach to clustering is to set up an objective function and then approximately find the optimal solution according to the objective. Common examples of these objective functions include k-median and k-means, in which the goal is to find k centers to minimize the sum of the distances (or sum of the squared distances) from each point to its closest center. Motivated by real-world constraints, further variants of clustering have been studied. For instance, in k-clustering with outliers, the goal is to find the best clustering (according to one of the above objectives) after removing a specified number of data points, which is useful for noisy data. Finding approximation algorithms to different clustering objectives and variants has attracted significant attention in the computer science community [7, 23, 24, 25, 28, 31, 41].

As datasets become larger, sequential algorithms designed to run on a single machine are no longer feasible for real-world applications. Additionally, in many cases data is naturally spread out among multiple locations. For example, hospitals may keep records of their patients locally, but may want to cluster the entire spread of patients across all hospitals in order to do better data analysis and inference. Therefore, distributed clustering algorithms have gained popularity in recent years [18, 20, 42, 32, 40, 29, 27]. In the distributed setting, it is assumed that the data is partitioned arbitrarily across s machines, and the goal is to find a clustering which approximates the optimal solution over the entire dataset while minimizing communication among machines. Recent work in the theoretical machine learning community establishes guarantees on the clusterings produced in distributed settings for certain problems [18, 20, 42]. For example, [42] provides distributed algorithms for k-center and k-center with outliers, and [20] introduces distributed algorithms for capacitated k-clustering under any ℓ_p objective. Along similar lines, the recent work of [32] provides constant-factor approximation algorithms for k-median and k-means with z outliers in the distributed setting. The work of Guha et al. also provides the best known communication complexity bound of $O(sk + z)$ where s is the number of machines, and z is the number of outliers.

Although the above results provide a constant-factor approximation to k-median or k-means objectives, many real-world applications desire a clustering that is close to a 'ground truth' clustering in terms of the structure, i.e., the way the points are clustered rather than in terms of cost. For example, for applications such as clustering proteins by function or clustering communities in a social network, there is some unknown target clustering, and the hope is that running a k-median or k-means algorithm will produce clusterings which are close to matching the target clustering. While in general having a constant factor approximation provides no guarantees on the closeness to the optimal clustering, a series of recent works has established that this is possible if the data has certain structural properties [10, 11, 12, 16, 21, 30, 39, 46]. For example, the $(1 + \alpha, \epsilon)$-*approximation stability* condition defined by [12] states that any $(1 + \alpha)$-approximation to the clustering objective is ϵ-close to the target clustering. For such instances, it is indeed possible to output a clustering close to the ground truth in polynomial time, even for values of α such that computing a $(1+\alpha)$-approximation is NP-hard. We follow this line of research and ask whether distributed clustering is possible for non worst-case instances, in the presence of outliers.

1.1 Our contributions

A distributed clustering instance consists of a set of n points in a metric space partitioned arbitrarily across s machines. The problem is to optimize the k-median/k-means objective while minimizing the amount of communication across the machines. We consider algorithms that approximate the optimal cost as well as computing a clustering close to the target clustering in Hamming distance. Our contributions are as follows:

1. In Section 3, we give a centralized clustering algorithm whose output is ϵ-close to the target clustering, in the presence of z outliers, assuming the data satisfies $(1+\alpha,\epsilon)$-approximation stability and assuming a lower bound on the size of the optimal clusters. To the best of our knowledge, this is the first polynomial time algorithm for clustering approximation stable instances in the presence of outliers. Our results hold for arbitrary values of z, including when a constant fraction of the points are outliers, as long as there is a lower bound on the minimum cluster size.
2. We then give a distributed algorithm whose output is close to the target clustering, assuming the data satisfies $(1+\alpha,\epsilon)$-approximation stability. The communication complexity is $\widetilde{O}\left(sk\right)$, where s is the number of servers and k is the number of clusters. We also extend this to handle z outliers, with a communication complexity $\widetilde{O}\left(sk+z\right)$. This matches the worst-case communication of [32], while outputting a near-optimal clustering by taking advantage of new structural guarantees specific to approximation stability with outliers.
3. While the above algorithms improve over worst-case distributed clustering algorithms in terms of quality of the returned clustering, our algorithms use the same amount of communication as the worst case protocols. In Section 4, we show that the $\Omega(sk)$ and $\Omega(sk+z)$ communication costs for clustering without and with outliers are unavoidable even if data satisfies many types of stability assumptions that have been studied in the literature. Our lower bound of $\Omega(sk+z)$ for obtaining a c-approximation (for any $c\geq 1$) holds even when the data is arbitrarily stable, e.g., $(1+\alpha,\epsilon)$-approximation stable for all $\alpha\geq 0$ and $0\leq\epsilon<1$.
4. We also give an $\Omega(sk+z)$ lower bound for the problem of computing a clustering whose Hamming distance is close to the optimal clustering, even when the data is approximation-stable. Finally, we prove that our above $\Omega(sk+z)$ lower bounds hold for finding a clustering close to the optimal in Hamming distance even when it is guaranteed that the optimal clusters are completely balanced, i.e., each cluster is of size $\frac{n-z}{k}$ (in addition to the guarantee that the clustering satisfies approximation stability), implying our algorithms from Section 3 are optimal. Therefore, $\Omega(sk+z)$ is a fundamental communication bottleneck, even for real-world clustering instances.

1.2 Related Work

There is a long line of work on approximation algorithms for k-median and k-means clustering [24, 36, 41], and the current best approximation ratios are 2.675 [23] and 6.357 [4], respectively. The first constant-factor approximation algorithm for k-median with z outliers was given by Chen [28], and the current best approximation ratios for k-median and k-means with outliers are $7.081+\epsilon$ and $53.002+\epsilon$, respectively, given by Krishnaswamy et al. [38]. There is also a line of work on clustering with balance constraints on the clusters [5, 3, 30]. For k-median and k-means clustering in distributed settings, the work of Balcan et al. showed a coreset construction for k-median and k-means, which leads to a clustering algorithm with $\tilde{O}(skd)$ communication, where d is the dimension, and also studied more general graph topologies for distributed computing [18]. Huang et al. [34] showed a coreset construction for doubling metrics. Malkomes et al. showed a distributed 13- and 4- approximation algorithm for k-center with and without outliers, respectively [42]. Chen et al. studied clustering under the broadcast model of distributed computing, and also proved a communication complexity lower bound of $\Omega(sk)$ for distributed clustering [27], building on a recent lower bound for set-disjointness in the message-passing model [22]. Recently, [32] showed a distributed algorithm with $\tilde{O}(sk+z)$ communication for computing a constant-factor approximation to k-median clustering with z outliers. They also provide bicriteria approximations that remove $(1+\epsilon)z$ outliers to get a clustering of cost $O\left(1+\frac{1}{\epsilon}\right)$ times the cost of the optimal k-median clustering with z outliers, for any $\epsilon > 0$. Even more recently, [40] showed that there exists a bi-criteria algorithm with communication independent of z that achieves a constant approximation to the cost. In particular, their algorithm outputs $(1+\epsilon)z$ outliers and achieves a $(24+\epsilon)$-approximation with $O\left(\frac{sk}{\epsilon}+\frac{s\log\Delta}{\epsilon}\right)$ communication, where Δ is the aspect ratio of the metric.

In recent years, there has also been a focused effort towards understanding clustering for non worst-case models [43, 1, 21, 39]. The work of Balcan et al. defined the notion of approximation stability and showed an algorithm which utilizes the structure to output a nearly optimal clustering [12]. Approximation stability has been studied in a wide range of contexts, including clustering [15, 17, 14], the k-means++ heuristic [2], social networks [33], and computing Nash-equilibria [9]. A recent paper by Chekuri and Gupta introduces the model of clustering with outliers under perturbation resilience, a notion of stability which is related to approximation stability [26].

2 Preliminaries

Given a set V of points of size n, a distance metric d, and an integer k, let $\mathcal{C}$ denote a clustering of V, which we define as a partition of V into k subsets $C_1,\ldots,C_k$. Each cluster C_i contains a center c_i. When d is an arbitrary distance metric, we must choose the centers from the point set. If $V\subseteq\mathbb{R}^d$ and the distance metric is the Euclidean distance, then the centers can be any k points in $\mathbb{R}^d$. In fact, this distinction only changes the cost of the optimal clustering by at most a factor of 2 by the triangle inequality for any p (see, e.g., [8]).

The k-median and the k-means costs are $\sum_i\sum_{v\in C_i} d(c_i,v)$, and $\sum_i\sum_{v\in C_i} d(c_i,v)^2$ respectively. For k clustering with z outliers, the problem is to compute the minimum cost clustering over $n-z$ points, e.g., we must decide which z points to remove, and how to cluster the remaining points, to minimize the cost. We will denote the optimal k-clustering with z outliers by $\mathcal{OPT}$, and we denote the set of outliers for $\mathcal{OPT}$ by Z. We often overload notation and let $\mathcal{OPT}$ denote the objective value of the optimal clustering as well. We denote the optimal clusters as $C_1^*,\ldots,C_k^*$, with centers $c_1,\ldots,c_k$. We say that two clusterings $\mathcal{C}$ and $\mathcal{C}'$ are δ-close if they differ by only $\delta(n-z)$ points, i.e., $\min_\sigma\sum_{i=1}^k |C_i\setminus C'_{\sigma(i)}|<\delta(n-z)$. Let $C^*_{\min}=\min_{j\in[k]}|C_j^*|$, i.e., the minimum cluster size. Given a point $c\in V$, we define $V_c\subset V$ to be the closest set of $C^*_{\min}$ points to c.

We study a notion of stability called approximation stability. Intuitively, a clustering instance satisfies this assumption if all clusterings close in *value* to $\mathcal{OPT}$ are also close in terms of the clusters themselves. This is a desirable property when running an approximation algorithm, since in many applications, the k-means or k-median costs are proxies for the final goal of recovering a clustering that is close to the desired "target" clustering. Approximation stability makes this assumption explicit. This was first defined for clustering with $z=0$ [12], however, we generalize the definition to the setting with outliers.

▶ **Definition 1** (approximation stability). *A clustering instance satisfies* $(1+\alpha, \epsilon)$-approximation stability *for k-median or k-means with z outliers if for all k-clusterings with z outliers, denoted by $\mathcal{C}$, if cost$(\mathcal{C}) \leq (1+\alpha) \cdot \mathcal{OPT}$, then $\mathcal{C}$ is ϵ-close to $\mathcal{OPT}$.*

This definition implies that all clusterings close in cost to $\mathcal{OPT}$ must have nearly the same set of outliers, because if $\mathcal{C}$ contains more than $\epsilon(n-z)$ points from Z, then $\mathcal{C}$ and $\mathcal{OPT}$ cannot be ϵ-close. This is similar to related models of stability for clustering with outliers, e.g. [26]. Note it is standard in this line of work to assume the value of α is known [12].

We will study distributed algorithms under the standard framework of the *coordinator model*. There are s servers, and a designated coordinator. Each server can send messages back and forth with the coordinator. This model is very similar to the *message-passing model*, also known as the *point-to-point* model, in which any pair of machines can send messages back and forth. In fact, the two models are equivalent up to constant factors in the communication complexity [22]. Most of our algorithms can be applied to the mapreduce framework with a constant number of rounds. For more details, see [20, 42].

For our communication lower bounds, we work in the multi-party message passing model, where there are s players, $P_1, P_2, \ldots, P_s$, who receive inputs X^1, X^2, $\ldots X^s$ respectively. They have access to private randomness as well as a common publicly shared random string R, and the objective is to communicate with a central coordinator who computes a function $f : X^1 \times X^2 \ldots \times X^s \rightarrow \{0,1\}$ on the joint inputs of the players. The communication has multiple rounds and each player is allowed to send messages to the coordinator. Note, we can simulate communication between the players by blowing up the rounds by a factor of 2. Given X^i as an input to player i, let Π be the random variable that denotes the transcript between the players and the referee when they execute a protocol Π. For $i \in [s]$, let Π_i denote the messages sent by P_i to the referee.

A protocol Π is called a δ-error protocol for function f if there exists a function Π_{out} such that for every input, $Pr\left[\Pi_{out} = f(X^1, X^2, \ldots X^s)\right] \geq 1-\delta$. The communication cost of a protocol, denoted by $|\Pi|$, is the maximum length of Π over all possible inputs and random coin flips of all the s players and the referee. The randomized communication complexity of a function f, $R_\delta(f)$, is the communication cost of the best δ-error protocol for computing f. For our lower bounds, we also consider that the data satisfies a very strong, general notion of stability which we call c-separation.

▶ **Definition 2** (separation). *Given $c \geq 1$ and a clustering objective, a clustering instance satisfies* c-separation *if $c \cdot \max_i \max_{u,v \in C_i^*} d(u,v) < \min_i \min_{u' \in C_i^*, v' \notin C_i^*} d(u',v')$.*

Intuitively, this definition implies the maximum distance between any two points in one cluster is a factor c smaller than the minimum distance across clusters. This assumption has been used in several papers (for clustering with no outliers) to show guarantees for various algorithms [13, 44, 37]. We note that this notion of stability captures a wide class of previously studied notions including perturbation resilience [21, 10, 16, 6] and approximation stability.

▶ **Definition 3** (perturbation resilience). *For $\beta > 0$, a clustering instance (V,d) satisfies $1+\alpha$-perturbation resilience for the k-means objective, if for any function $d' : V \times V \rightarrow \mathbb{R}_{\geq 0}$, such that for all $p, q \in V$, $d(p,q) \leq d'(p,q) \leq (1+\beta)d(p,q)$, and the optimal clustering under d' is unique and equal to the optimal clustering under d, for the k-means objective.*

We note we can replace the objective with any center based objective such as k-median or k-center. Next, we show that *separation* implies *approximation stability* and *perturbation resilience.* We defer the proof to the Appendix.

▶ **Lemma 4.** *Given $\alpha, \epsilon > 0$, and a clustering objective such as k-median, let (V, d) be a clustering instance which satisfies c-separation, for $c > (1+\alpha)n$, where $n = |V|$. Then (V, d) satisfies $(1+\alpha, \epsilon)$-approximation stability and $(1+\alpha)$-perturbation resilience.*

3 Approximation Stability with Outliers

In this section, we give a centralized algorithm for clustering with z outliers under approximation stability, and then extend it to a distributed algorithm for the same problem. To the best of our knowledge, this is the first result for clustering with outliers under approximation stability, as well as the first distributed algorithm for clustering under approximation stability even without outliers. We defer the details to the Appendix. Our algorithm can handle any fraction of outliers, even when the set of outliers makes up a constant fraction of the input points. For simplicity, we focus on k-median.

▶ **Theorem 5** (Centralized Clustering). *Algorithm 1 runs in poly$\left(n, \left(\frac{\alpha}{\epsilon}\left(k+\frac{1}{\alpha}\right)\right)^{\frac{1}{\alpha}}\right)$ time and outputs a clustering that is ϵ-close to $\mathcal{OPT}$ for k-median with z outliers under $(1+\alpha, \epsilon)$-approximation stability, assuming for all i, $|C_i^*| \geq 2\left(1+\frac{5}{\alpha}\right)\epsilon(n-z)$.*

Note that the runtime is at most poly$\left(n^{\frac{1}{\alpha}}\right)$, and if $\frac{\alpha}{\epsilon} \in \Theta(k)$, the runtime is poly$\left(n, k^{\frac{1}{\alpha}}\right)$. The algorithm has two high-level steps. First, we use standard techniques from approximation stability without outliers to find a list of clusters $\mathcal{X}$, which contains clusters from the optimal solution (with $\leq \left(1+\frac{1}{\alpha}\right)\epsilon(n-z)$ mistakes), and clusters made up mostly of outlier points. We show how all but $1/\alpha$ of the outlier clusters must have high cost if their size were to be extended to the minimum optimal cluster size, and can thus be removed from our list $\mathcal{X}$. Finally, we use brute force enumeration to remove the final $\frac{1}{\alpha}$ outlier clusters, and after another cluster purifying step, we are left with a k clustering which $(1+\alpha)$-approximates the cost and thus is guaranteed to be ϵ-close to optimal.

We begin by outlining the key properties of $(1+\alpha, \epsilon)$-approximation stability. Let w_{avg} denote the average distance from each point to its optimal center, so $w_{avg} \cdot (n-z) = \mathcal{OPT}$. The following lemma is the first of its kind for clustering with outliers and establishes two key properties for approximation stable instances. Intuitively, the first property bounds the number of points that are far away from their optimal center, and follows from Markov's inequality. The second property bounds the number of points that are either closer on average to the center of a non-optimal cluster that the optimal one or are outliers that are close to some optimal center as compared to a point belonging to that cluster.

▶ **Lemma 6.** *Given a $(1+\alpha, \epsilon)$-approximation stable clustering instance (V, d) for k-median such that for all i, $|C_i^*| > 2\epsilon(n-z)$, then* ***Property 1:*** *For all $y > 0$, there exist at most $\frac{y\epsilon}{\alpha}(n-z)$ points, v, such that $d(v, c_v) \geq \frac{\alpha w_{avg}}{y\epsilon}$.* ***Property 2:*** *There are fewer than $\epsilon(n-z)$ total points with one of the following two properties: the point v is in an optimal cluster C_i^*, and there exists $j \neq i$ such that $d(v, c_j) - d(v, c_i) \leq \frac{\alpha w_{avg}}{\epsilon}$, or, the point v is in Z, and there exists i and $v' \in C_i^*$ such that $d(v, c_i) \leq d(v', c_i) + \frac{\alpha w_{avg}}{\epsilon}$ (recall that Z denotes the set of outliers from the optimal clustering).*

We define a point as *bad* if it falls into the bad case of either Property 1 (with $y = 5$) or Property 2, and we denote the set of bad points by B. Otherwise, a point is *good.* From Properties 1 and 2, $|B| \leq \left(1+\frac{5}{\alpha}\right)\epsilon(n-z)$. For each i, let G_i denote the good points from the optimal cluster C_i^*. We consider the graph $G' = (V, E')$ called the *neighborhood graph*, constructed by adding an edge (u, v) iff there are at least $|B|+2$ points w such that $d(u, w), d(v, w) \leq \tau = \frac{2w_{avg}}{5}$. Under approximation stability, the graph G' has the following structure: there is an edge between all pairs of good points from C_i^* and there is no edge between any pair of good points belonging to distinct clusters, C_i^*, C_j^*. Further, these points do not have any common neighbors. Since the set of good points in each cluster, denoted by G_i, form cliques of size $> |B|$ and are far away from one another, and there are $\leq |B|$ bad points, it follows that each G_i is in a unique connected component C_i' of G'.

In the setting without outliers, the list of connected components of size greater than $\left(1+\frac{5}{\alpha}\right)\epsilon n$ is exactly $\{C_1', \ldots, C_k'\}$. However, in the setting with outliers, we can only return a set $\mathcal{X}$ which includes $\{C_1', \ldots, C_k'\}$ but also may include many other outlier clusters which are hard to distinguish from the optimal clusters. Although approximation stability tells us that any set Z' of outliers must have a much higher cost than any optimal cluster C_i^* (since we can arrive at a contradiction by replacing the cluster C_i^* with the cluster Z'), this is not true when the size of Z' is even slightly smaller than C_i^*. Since the good clusters returned are only $O\left(\frac{\epsilon}{\alpha}\right)$-close to optimal, many good clusters may be smaller than outlier clusters, and so a key challenge is to distinguish outlier clusters Z' from good clusters C_i'.

To accomplish this task, we compute the minimum cost of each cluster, pretending that its size is at least $C_{\min}^*$ (the size of the minimum optimal cluster, which we can guess in polynomial time). In our key structural lemma (Lemma 7), we show that nearly all outlier components will have large cost. Given a set of points Q, we define $\text{cost}_{\min}(Q)$ to be the minimum cost of Q if it were extended to $C_{\min}^*$ points. Note, $\text{cost}_{\min}(Q)$ can be computed in polynomial time by iterating over all points $c \in Q$, for each such point constructing V_c by adding the the $C_{\min}^* - |Q|$ points closest to c, computing the resulting cost, and taking the minimum over all such costs.

Algorithm 1 k-median with z-outliers under Approximation Stability.

`Input`: Clustering instance (V, d), cost w_{avg}, value C^*_{min}, integer $x > 0$.

1. Create the neighborhood graph on V with parameters $\tau = \frac{2w_{avg}}{5\epsilon}$ and $b = C^*_{min} - (1 + \frac{5}{\alpha})\epsilon(n-z)$ as follows: for each $u, v \in V$, add an edge (u,v) iff there exist $\geq b$ points $w \in V$ such that $d(u,w), d(w,v) \leq \tau$. Denote the connected components by $\mathcal{X} = \{Q_1, \ldots, Q_d\}$.
2. For each Q_i, compute $\text{cost}_{\min}(Q_i) = \min_{c \in Q_i} \min_{V_c} \sum_{v \in V_c} d(c,v)$, where V_c must satisfy $|V_c| \geq C^*_{min}$ and $Q_i \subseteq V_c$. Create a new set $\mathcal{X}' = \{Q_i \mid \text{cost}_{min}(Q_i) < \left(3 + \frac{2\alpha}{5}\right)\frac{1}{x} \cdot \mathcal{OPT}\}$.
.
3. For all $0 \leq t \leq x$, for each size t subset $\mathcal{X}_t' \subseteq \mathcal{X}'$ and size $(k - |\mathcal{X}'| - t)$ subset $\mathcal{X}_t \subseteq (\mathcal{X} \setminus \mathcal{X}')$,
   a. Create a new clustering $\mathcal{C} = \mathcal{X}' \cup \mathcal{X}_t \setminus \mathcal{X}_t'$.
   b. For each point $v \in V$, define $I(v)$ as the index of the cluster in $\mathcal{C}$ with minimum median distance to v, e.g., $I(v) = \text{argmin}_i\,(d_{\text{med}}(v, Q_i))$ where $d_{\text{med}}(v, Q_i)$ denotes the median distance from v to Q_i.
   c. Let $V' \subseteq V$ denote the $n - z$ points with the smallest values of $d(v, c_{I(v)})$. For all i, set $Q_i' = \{v \in V' \mid I(v) = i\}$.
   d. If $\sum_i \text{cost}(Q_i') \leq (1+\alpha)\mathcal{OPT}$, return $\{Q_1, \ldots, Q_k\}$.

▶ **Lemma 7.** *Given an instance of k-median clustering with z outliers such that each optimal cluster $|C_i^*| > 2\left(1+\frac{5}{\alpha}\right)\epsilon(n-z)$, for any $x \in \mathbb{N}$, the instance satisfies $(1+\alpha, \epsilon)$-approximation stability for $\alpha > \frac{35}{5x-4}$, and there are at most x disjoint sets of outliers Z' such that $|Z'| > \min_i |C_i^*| - \left(1+\frac{5}{\alpha}\right)\epsilon(n-z)$ and $\text{cost}_{\min}(Z') \leq \left(3 + \frac{2\alpha}{5}\right)\frac{1}{x}\mathcal{OPT}$.*

The key ideas behind the proof are as follows. If there are two sets of outliers Z_1 and Z_2 both with fewer than $C_{\min}^*$ points, then we can obtain a contradiction by taking into account both sets of outliers. Set $1 \leq z_1, z_2 \leq \left(1+\frac{5}{\alpha}\right)\epsilon(n-z)$ such that $|Z_1| = C_{\min}^* - z_1$ and

$|Z_2| = C^*_{\min} - z_2$, and assume without loss of generality that $z_1 < z_2$. We design a different clustering $\mathcal{C}'$ by first replacing the minimum-sized cluster in the optimal clustering with Z_1. The cost of the points in Z_2 is low by assumption. However, we have now potentially assigned more than z points to be outliers by an additive z_1 amount. Hence, in order to create a valid clustering that is far from $\mathcal{OPT}$ we need to add back at least z_1 more outlier points. We do this by choosing z_1 outlier points from Z_2 that are closest to an optimal center in $\mathcal{OPT}$. To bound the additional cost incurred, we use the fact that Z_2 must be close to at least z_2 points from $V \setminus Z$, by the assumption that $\text{cost}_{\min}(Z_2)$ is low, and use these points to bound the distance from centers in $\mathcal{OPT}$ to the z_1 points that were added back. In the full proof, we extend this idea to x sets $Z_1, \ldots, Z_x$ to achieve a tradeoff between x and α.

From Lemma 7, we show a threshold of $\text{cost}_{\min}$ for the components of $\mathcal{X}$, such that all but x optimal clusters are below the cost threshold, and all but x outlier clusters are above the cost threshold. Then we can brute force over all ways of excluding x low-cost sets and including x high-cost sets, and we will be guaranteed that one combination contains a clustering which is $O\left(\frac{\epsilon}{\alpha}\right)$-close to the optimal. However, we still need to recognize the right clustering when we see it. To do this, we show that after performing one more cluster purifying step which is inspired by arguments in [12] - reassigning all points to the component with the minimum median distance - we will reduce our error to $\epsilon(n-z)$ in Hamming distance and we show how to bound the total cost of these mistakes by $\frac{4\alpha}{5}\mathcal{OPT}$. Therefore, during brute force enumeration, we return immediately when we find a clustering with cost at most $(1+\alpha)\mathcal{OPT}$ (and thus must be ϵ-close to $\mathcal{OPT}$). Then we can try all possible values of C^*_{min} while only incurring a polynomial increase in the runtime of the algorithm. For w_{avg}, we first run an approximation algorithm for k-median with z outliers to obtain a constant approximation to w_{avg} (e.g., [38]). The constant in the minimum allowed optimal cluster size then increases by a factor of 7. This is because we need to use a smaller value of τ when constructing the neighborhood graph G', and so the number of "bad" points increases. In order to show all the good connected components from G' contain a majority of good points, we merely increase the bound on the minimum cluster size.

Distributed Setting. Next, we give a distributed algorithm for approximation stability with outliers using $\tilde{O}\left(sk+z\right)$ communication. However, as opposed to worst case, we can get close to the ground truth (target) clustering. In Section 4, we show a matching lower bound.

▶ **Theorem 8** (Distributed Clustering). *Given a $(1+\alpha, \epsilon)$-approximation stable clustering instance, there exists an algorithm that runs in poly$\left(n^{\frac{1}{\alpha}}\right)$ time and with high probability outputs a clustering that is $O(\epsilon)$-close to $\mathcal{OPT}$ for k-median with $\tilde{O}\left(sk+z\right)$ communication if each optimal cluster C^*_i has cardinality at least* $\max\left\{2\left(1+\frac{22}{\alpha}\right)\epsilon(n-z), \Omega\left(\frac{(n-z)}{sk}\right)\right\}$.

We start by giving intuition for our algorithm where there are no outliers. The high-level structure of the algorithm can be thought of as a two-round version of the centralized algorithm from approximation stability with no outliers [12]. Each machine effectively creates a coreset of its input, consisting of a weighted set of points, and sends these weighted points to the coordinator. The coordinator runs the same algorithm on these sets of weighted centers, to output the final solution.

In the analysis, we define good and bad points using Property *(1)* above with $y = 20$ as opposed to $y = 5$, so that there are more bad points than in the non-distributed setting, $|B| = \left(1+\frac{1}{20}\right)\epsilon(n-z)$, but for each optimal cluster C^*_i, the good points G_i are even more tightly concentrated. In the first round, each machine computes the neighborhood graph described above with parameter $\tau = \frac{w_{avg}}{10}$. This more stringent definition of τ ensures that

Claims *(1)* and *(2)* above are not only true for the input point set, but also true for a summarized version of the point set, where each point represents a ball of data points within a radius of τ. Therefore, there is still enough structure present such that the coordinator can compute a near-optimal clustering, and finally the coordinator sends the k resulting (near optimal) centers to each machine.

Now we expand this approach to the case with outliers. The starting point of the algorithm is the same: we perform two rounds of the sequential approximation stability algorithm with no outliers, so that each machine computes a summary of its point set, and the coordinator clusters the points it receives. Recall that in the centralized setting, running the non-outlier algorithm produces a list of clusters $\mathcal{X}$, some of which are near-optimal and some of which are outlier clusters, and then we crucially computed the $\text{cost}_{\min}$ of each potential cluster to distinguish the near-optimal clusters from the outlier clusters. In the distributed setting, we can construct the set $\mathcal{X}$ using the two-round approach.

However, the $\text{cost}_{\min}$ computation is sensitive to small sets of input points, and, as a result, the coresets will not give the coordinator enough information to perform this step correctly. In particular, this involves finding the closest points to a component that increase the cardinality to $C^*_{\min}$, and these points may be arbitrarily partitioned across the machines. Furthermore, the centralized algorithm can easily try all possible centers to compute the minimum cost of a given component Q, but it is much harder in the distributed setting to even find an approximately optimal center. Even with a center c chosen, the coordinator needs a near-exact estimate of the minimum cost of Q, however, it does not know the $C^*_{\min}$ closest points to c. Therefore our distributed algorithm must balance accuracy with communication.

For each component Q, the coordinator simulates $\log n$ random draws from Q by querying its own weighted points, and then querying the machine of the corresponding point. This allows the coordinator to find a center c whose cost is only a constant factor away from the best center. To compute $\text{cost}_{\min}(c)$, the coordinator runs a binary-search procedure with all machines to find the minimum distance t such that $B_t(c)$ contains more than $C^*_{\min}$ points.

Given a random point v from Q, by a Markov inequality, there is a $1/2$ chance that the cost of center v on V_c is at most twice the cost with center c. From a Chernoff bound, by sampling $10 \log n$ points for each component, each component will find a good center with high probability. Therefore, the coordinator can evaluate the cost of each component up to a factor of 2, which is sufficient to (nearly) distinguish the outlier clusters from the near-optimal clusters. The rest of the algorithm is similar to the centralized setting. We brute-force all combinations of removing x low-cost clusters from $\mathcal{X}$ and adding back x high-cost clusters from x. We perform one more cluster purifying step, and then check the cost of the resulting clustering. If the cost is smaller than $(1+\alpha)w_{avg}(n-z)$, then we return this clustering.

Similar to the centralized setting, we can use existing algorithms (e.g. [32]) to approximate w_{avg}, and we can use binary search to find $C^*_{\min}$. The algorithm communicates $\tilde{O}(sk+z)$ bits to approximate w_{avg}. The communication in the first step is $O(sk \log n)$, since there are at most $\min\left\{\frac{s}{\epsilon}, O(sk)\right\}$ sets of size at least $\max\left\{\frac{\epsilon n}{s}, \frac{n}{sk}\right\}$, each of which are communicated to the coordinator. The total communication to compute $\text{cost}_{\min}$ for every component is $\tilde{O}(sk)$. The binary search wrapper to find $C^*_{\min}$ adds a $\log n$ multiplicative factor. Therefore, the total communication is $\tilde{O}(sk+z)$.

4 Communication Complexity Lower Bounds

In this section, we show lower bounds for the communication complexity of distributed clustering with and without outliers. We prove $\Omega(sk+z)$ lower bounds for two types of clustering problems: computing a clustering whose cost is at most a c-approximation to the optimal (or even just to determine the cost up to a factor of c) for any $c \geq 1$, and computing a clustering which is δ-close to $\mathcal{OPT}$, for any $\delta < \frac{1}{4}$. This shows prior work of [32] is tight.

Our lower bounds hold even under c-separation (Definition 2). Furthermore, our lower bounds for δ-close clustering hold even under a weaker version of clustering, which we call *locally-consistent clustering.* In this problem, instead of assigning a globally consistent index in $\{1, \ldots, k\}$ for each point, each player only needs to assign indices to its points that is consistent in a local manner, e.g., the assignment of index set $\{1, \ldots, k\}$ to clusters $\{C_1, \ldots, C_k\}$ chosen by player 1 might be a permutation of the assignment chosen by player 2. We work in the communication model described in Section 2.

▶ **Definition 9** (Multi-party set disjointness ($\mathsf{DISJ}_{s,\ell}$)). *Given s players, denoted by P_1, P_2, $\ldots P_s$, player P_j receives as input a bit vector X^j of length ℓ. Let X denote the a binary matrix such that each X^j is a column of X. Let X_i denote the i-th row of X and $X^j[i]$ denote the (i,j)-th entry of X. Then, $\mathsf{DISJ}_{s,\ell} = \bigvee_{i\in[\ell]} \bigwedge_{j\in[s]} X^j[i]$, i.e., $\mathsf{DISJ}_{s,\ell} = 0$ if at least one row of X corresponds to the all ones vector and 1 otherwise.*

We note that set disjointness is a fundamental problem in communication complexity and we use the following lower bound for $\mathsf{DISJ}_{s,\ell}$ in the message-passing model by [22]:

▶ **Theorem 10** (Communication complexity of $\mathsf{DISJ}_{s,\ell}$ [22]). *For any $\delta > 0$, $s = \Omega(\log(n))$ and $\ell \geq 1$, the randomized communication complexity of multi-party set disjointness, $R_\delta(\mathsf{DISJ}_{s,\ell})$, is $\Omega(s\ell)$.*

We use the above theorem to show a lower bound of $\Omega(sk+z)$ for distributed clustering algorithms that attain an approximation to the cost of the optimal clustering under center-based clustering objectives such as k- median and k-means even if the instance satisfies strong beyond-worse case stability assumptions. We note that our first reduction is a slight modification of the reduction that appears in [27] and we show how to extend the reduction to stable instances and to account for outliers. Intuitively, the parameters of the reduction are carefully chosen so that the clustering instance created either has k or $k+1$ distinct locations, toggled by the disjointness instance being yes or no. The lower bound for outliers requires starting with a two player, balanced instance of set disjointness, introduced by [45].

▶ **Theorem 11.** *Given $c_1 \geq 1$, $c_2 \geq 0$, the communication complexity for computing a c_1-approximation for k-median, k-means, or k-center clustering is $\Omega(sk)$, even when promised that the instance satisfies c_2-separation. Further, for the case of clustering with z outliers, computing a c_1-approximation to k-median, k-means, or k-center cost, given the promise that the instance satisfies c_2-separation requires $\Omega(sk+z)$ bits of communication.*

We note that thus far we have ruled out a distributed clustering algorithm that has communication complexity less than $\Omega(sk+z)$ to output the exact clustering under strong stability assumptions. Next, we show an $\Omega(sk+z)$ lower bound when the goal is to return a clustering that is $\delta < \frac{1}{4}$-close to optimal in Hamming distance, i.e., outputting a clustering that differs from the optimal clustering in a δ-fraction of the points, given that the instance is $(1+\alpha, \epsilon)$-stable for any setting of these parameters.

We show that our lower bounds hold even when the algorithm outputs a c-approximate solution to the clustering cost of a $\frac{1}{4}$-close clustering. Intuitively, the proof is again a reduction from $\mathsf{DISJ}_{s,\ell}$, similar to the proof of Theorem 11. The main difference is that the coordinator now adds roughly $\frac{n}{2}$ copies of a subset of points in our construction, to make the optimal clustering stand out from the rest. The main technical challenge is to figure out how to add these points such that the optimal clustering stands out in both yes and no instances. Therefore, recovering an approximation to the optimal clustering in Hamming distance provides enough information to solve set disjointness.

▶ **Theorem 12.** *Given $c_1 \geq 1$, $c_2 \geq 0$, and $0 < \delta < \frac{1}{4}$, the communication complexity for computing a c_1-approximation to the k-median, k-means, or k-center objective with z outliers and outputting a clustering that is δ-close to the optimal, $\Omega(sk + z)$, even when promised that the instance satisfies c_2-separation.*

Though the above lower bounds are quite general, it is possible that the hard instances may have the optimal clusters to be very different in cardinality if sk is large. The smallest cluster may be size $O\left(\frac{n}{sk}\right)$, while the largest cluster may be size $\Omega(n)$. Often, real-world instances have roughly balanced clusters. There is a line of work on clustering with balance constraints on the clusters [5, 3, 30], and some of our algorithmic results assume a lower bound on the minimum cluster size.

We note that our previous reduction for proving a lower bound against δ-close clustering algorithms fundamentally relies on testing the cardinality of the clusters. Therefore, we extend our previous lower bounds to the setting where we are promised that the input clusters are well balanced, i.e., have roughly the same cardinality. We still consider algorithms that only get δ-close to the optimal clustering. We begin by defining the following basic notions from information theory:

▶ **Definition 13** (Entropy and conditional entropy). *The entropy of a random variable X drawn from distribution μ, denoted as $X \sim \mu$, with support χ, is given by $H(X) = \sum_{x\in\chi} \Pr_\mu[X = x] \log \frac{1}{\Pr_\mu[X=x]}$. Given two random variable X and Y with joint distribution μ, the entropy of X conditioned on Y is given by $H(X \mid Y) =$ $\mathbb{E}_{y\sim\mu(Y)} \left[\sum_{x\in\chi} \Pr_{\mu(X|Y=y)}[X = x] \log \frac{1}{\Pr_{\mu(X|Y=y)}[X=x]}\right]$.*

Note, the binary entropy function $H_2(X)$ is the entropy function for the distribution $\mu(X)$ supported on $\{0, 1\}$ such that $\mu(X) = 1$ with probability p and $\mu(X) = 0$ otherwise.

▶ **Definition 14** (Mutual information and conditional mutual information). *Given random variables X and Y, the mutual information between X and Y is given by $I(X; Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$. The conditional mutual information between X and Y, conditioned on a random variable Z is given by $I(X; Y \mid Z) = H(X \mid Z) - H(X \mid Y, Z) = H(Y \mid Z) - H(Y \mid X, Z)$.*

Recall, the δ-error randomized communication complexity of $\mathcal{A}$, $R_\delta(\mathcal{A})$, in the message passing model is communication complexity of any randomized protocol Π that solves $\mathcal{A}$ with error at most δ. Let μ be a distribution over $X^1, X^2, \dots X^s$. We call a deterministic protocol (δ, μ)-error if it gives the correct answer for $\mathcal{A}$ on at least a $1-\delta$ fraction of the input, weighted by the distribution μ. Let $D_{\mu,\delta}(\mathcal{A})$ denote the cost of the minimum communication (δ, μ)-error protocol. By Yao's minimax lemma, $R_\delta(\mathcal{A}) \geq \max_\mu D_{\mu,\delta}(\mathcal{A})$. Therefore, in order to lower bound the randomized communication complexity of $\mathcal{A}$, it suffices to construct a distribution μ over the input such that any deterministic protocol that is correct on $1-\delta$ fraction of any input can be analyzed easily. The communication complexity of a protocol Π is also lower bounded by its information complexity.

▶ **Definition 15** (Information complexity of $\mathcal{A}$). *For $i \in [s]$, let Π_i be a random variable that denotes the transcript of the messages sent by player P_i to the coordinator. We overload notation by letting Π denote the concatenation of Π_1 to Π_s. Then, the information complexity of $\mathcal{A}$ is given by $\mathsf{IC}_{\mu,\delta}(\mathcal{A}) = \min_{(\delta,\mu)\text{-}error\ \Pi} I(X_1, X_2, \dots X_s; \Pi)$.*

Since information lower bounds communication (see, e.g., [35]), $R_\delta(\mathcal{A}) \geq \mathsf{IC}_{\mu,\delta}(\mathcal{A})$ in the message passing model. So our proof strategy is to construct a distributed protocol for solving the above problem using an algorithm that obtains a δ-close clustering for balanced clusters. We then design a distribution μ over the input and lower bound the information complexity of the resulting problem by $\Omega(k)$. We then amplify the bound by introducing $s/2$ copies of Alice and Bob (as before). Next, we describe this proof strategy in detail.

We begin with a two-player communication problem, where Alice and Bob receive length ℓ bit vectors, and the objective is to compute the AND function on each index in $[\ell]$. We then construct a gadget that reduces computing AND on any particular index to solving a 2-clustering problem, where each cluster has 2 points (and thus the instance is balanced). The gadget is such that Alice and Bob insert 2 points each, at a fixed set of locations determined by their input, and the optimal 2-clustering places Alice's points in different clusters iff the AND evaluates to true. The same holds for Bob. Therefore, Alice and Bob learn each other's bit simply looking at the output of the clustering algorithm. The players then create this gadget for each index in their input, and place the gadgets sufficiently far from each other.

Observe, a δ-close clustering algorithm must output a $(1-2\delta)$-fraction of the clusters correctly. Using such an algorithm as a distributed protocol enables the players to learn the AND function on a $(1-2\delta)$-fraction of the coordinates. Note the underlying communication problem here does not correspond to well-studied problems such as set disjointness. However, some proofs of the lower bound for multi-party set disjointness do reduce to computing the AND function on every index [19]. Therefore, we relate the communication complexity of the above problem to the amount of information revealed by any protocol that is correct on a large fraction of the input.

We define a distribution μ over the input such that each bit for Alice and Bob is set to be 1 with probability 1/2 independently and 0 otherwise. Here, we observe that the δ-close clustering algorithm implies a $(2\delta, \mu)$-protocol for computing AND on each index. Therefore, we prove that the information complexity of a $(2\delta, \mu)$-protocol is $\Omega(\ell)$. Intuitively, this says any correct deterministic protocol that is correct on a $1-2\delta$ fraction of the input, for the given input distribution μ, must reveal $\Omega(1)$ information on every index that has at least one 1, which amounts to communicating the bit. Since our gadget has 2 clusters for each index, setting $\ell = \Theta(k)$ obtains an $\Omega(k)$ communication lower bound. Using our previous strategy of duplicating the Alice and Bob players $s/2$ times, we obtain the following theorem:

▶ **Theorem 16.** *Given $\delta < \frac{1}{4}$ and the promise that the optimal clusters are balanced, i.e., the cardinality of each cluster is $\frac{n}{k}$, the communication complexity for computing a clustering that is δ-close to the optimal k-means or k-median clustering is $\Omega(sk)$.*

Finally, we extend the above lower bound to clustering instances that are balanced and also satisfy $(1+\alpha, \epsilon)$-approximation stability, again obtaining an $\Omega(sk+z)$ lower bound. Perhaps surprisingly, we show that there is no trade-off between the stability parameters and the communication lower bound even if the clusters are balanced and the algorithm outputs a clustering that is $\delta < \epsilon/4$ close to the optimal clustering. In contrast, our previous result can handle all $\delta < 1/4$. Intuitively, to obtain a clustering instance that is $(1+\alpha, \epsilon)$-approximation stable, we restrict the number of indices on which AND evaluates to 1 to be $O(\epsilon n)$. Therefore we start with a promise version of the multi-party set disjointness problem, where the promise states if the sets intersect, they intersect on exactly one element. Formally,

▶ **Definition 17** (Promise multi-party set disjointness ($\mathsf{PDISJ}_{s,\ell}$)). *Given s players denoted by $P_1, \dots, P_s$, each player receives a bit vector X^j of length ℓ. Let X denote a binary matrix such that each X^j is a column of X. Let X_i denote the i-th row of X and X_i^j denote* *the (i,j)-th entry of X. We are promised that at most one row of X has all ones. Then,* $\mathsf{PDISJ}_{s,\ell} = \bigvee_{i\in[\ell]} \bigwedge_{j\in[s]} X_i^j$, *i.e.,* $\mathsf{PDISJ}_{s,\ell} = 0$ *if any row of X corresponds to the all ones vector and* 1 *otherwise.*

We use a result of [19] to lower bound the communication complexity of set-disjointness in the multi-party communication model.

▶ **Theorem 18** (Communication complexity of $\mathsf{PDISJ}_{s,\ell}$ [19]). *For any $\delta > 0$, $s, \ell \in \mathbb{N}$, the randomized communication complexity of promise multi-party set disjointness, $R_\delta(\mathsf{PDISJ}_{s,\ell})$, is $\Omega(\ell/s^2)$.*

We show any algorithm obtaining a δ-close clustering, given the clusters are balanced and the clustering instance is $(1+\alpha,\epsilon)$-stable can be converted into a randomized communication protocol that solves $\mathsf{PDISJ}_{s,\ell}$. At a high level, Alice and Bob receive length ℓ bit vectors and create a gadget for each index in $[\ell]$. If the number of indices on which the bit vectors intersect is at most ϵk, the instance is $(1+\alpha,\epsilon)$-stable. We ensure this by constructing gadgets that incur an arbitrarily high cost in all other cases (see the Appendix for details).

We note that if our clustering instance has exactly one index on which AND evaluates to 1, it is easy for a randomized protocol to be incorrect with good probability. In order to circumvent this issue and maintain $(1+\alpha,\epsilon)$-stability, Alice and Bob create $\epsilon n - 1 = 2\epsilon k - 1$ dummy indices that are set to 1 for both players. Further, Alice and Bob use public randomness to agree on a uniform permutation of the padded input and apply this permutation before constructing the gadgets and running the clustering algorithm. Intuitively, permuting the indices ensures that the δ-close clustering gets a typical cluster right with reasonable probability, by being oblivious to the dummy clusters that were used as padding.

Since we uniformly permute the indices of the input before running the protocol, for any given index, the corresponding cluster has Hamming distance 0 from the optimal clustering with probability at least $1-\epsilon$. This implies at most an ϵ-fraction of the clusters are incorrect. The protocol outputs a clustering that is known to both Alice and Bob. For each index of their input, they know whether their pair of points lie in the same cluster or different clusters. Let $\mathcal{I}$ be the set of indices for which Alice and Bob's points lie in different clusters. If $\mathcal{I} > 4\epsilon k$, the protocol outputs fail. Otherwise, Alice communicates her input on the set $\mathcal{I}$ to Bob. Bob applies an inverse random permutation to indices in set $\mathcal{I}$, and verifies if the indices correspond to the dummy indices that were added or indeed the sets are not disjoint. Note the verification step requires additional communication. Since $\mathcal{I} \leq 4\epsilon k$, and ϵ is at most a small constant, the total additional communication is $O(k/c)$ for some large constant c.

Consider the case where the sets are not disjoint. Then there is an index i^* such that AND on this index evaluates to 1, and with probability at least $1-\epsilon$, the clustering algorithm correctly clusters the corresponding 2-means gadget. This implies that Alice and Bob know that their pair of points lie in different clusters, thus i^* is in the set $\mathcal{I}$ and Alice communicates her input on index i^* to Bob. Bob can then verify that i^* is not a dummy index and indeed the sets are not disjoint.

The case where the sets are disjoint is more subtle. Now the clustering algorithm may return $4\epsilon k$ indices such that Alice's points belong to separate clusters, i.e., they correspond to a $(1,1)$ input, therefore leading to false positives. However, we observe that we can verify if the sets are disjoint by Alice sending over her input bits on the set $\mathcal{I}$ to Bob. Bob can verify if they correspond to the dummy indices and the sets are indeed disjoint. This increases the over all communication by $O(k/c)$. We recall that the promise problem requires $\Omega(\ell) = \Omega(k)$ communication and thus the communication of the above protocol is $\Omega(k - \epsilon k) = \Omega(k)$. We use the technique of cloning Alice and Bob $s/2$ times, so communicating the solution to each player requires $\Omega(sk)$ bits of communication. Finally, we show how to extend this lower bound to the case of outliers to get an overall $\Omega(sk+z)$ lower bound:

▶ **Theorem 19.** *Given a $(1+\alpha,\epsilon)$-approximation stable instance with z outliers such that $\epsilon = o(1)$ and $\delta < \frac{\epsilon}{4}$, and the promise that the optimal clusters are balanced, i.e., the cardinality of each cluster is $\frac{n-z}{k}$, the communication complexity for computing a clustering that is δ-close to the optimal k-means or k-median clustering is $\Omega(sk+z)$.*

References

1 Margareta Ackerman and Shai Ben-David. Clusterability: A theoretical study. In *Artificial Intelligence and Statistics*, pages 1–8, 2009.

2 Manu Agarwal, Ragesh Jaiswal, and Arindam Pal. k-means++ under Approximation Stability. *Theoretical Computer Science*, 588:37–51, 2015.

3 Gagan Aggarwal, Tomás Feder, Krishnaram Kenthapadi, Samir Khuller, Rina Panigrahy, Dilys Thomas, and An Zhu. Achieving anonymity via clustering. In *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 153–162, 2006.

4 Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better Guarantees for k-Means and Euclidean k-Median by Primal-Dual Algorithms. *arXiv preprint*, 2016. `arXiv:1612.07925`.

5 Sara Ahmadian and Chaitanya Swamy. Approximation Algorithms for Clustering Problems with Lower Bounds and Outliers. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 69:1–69:15, 2016. `doi:10.4230/LIPIcs.ICALP.2016.69`.

6 Haris Angelidakis, Konstantin Makarychev, and Yury Makarychev. Algorithms for Stable and Perturbation–Resilient Problems. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, 2017.

7 Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM Journal on Computing*, 33(3):544–562, 2004.

8 Pranjal Awasthi and Maria-Florina Balcan. Center based clustering: A foundational perspective. *CRC*, 2014.

9 Pranjal Awasthi, Maria-Florina Balcan, Avrim Blum, Or Sheffet, and Santosh Vempala. On nash-equilibria of approximation-stable games. In *International Symposium on Algorithmic Game Theory*, pages 78–89. Springer, 2010.

10 Pranjal Awasthi, Avrim Blum, and Or Sheffet. Center-based clustering under perturbation stability. *Information Processing Letters*, 112(1):49–54, 2012.

11 Pranjal Awasthi and Or Sheffet. Improved spectral-norm bounds for clustering. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 37–49. Springer, 2012.

12 Maria-Florina Balcan, Avrim Blum, and Anupam Gupta. Clustering under approximation stability. *Journal of the ACM (JACM)*, 60(2):8, 2013.

13 Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. A discriminative framework for clustering via similarity functions. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, pages 671–680, 2008.

14 Maria-Florina Balcan and Mark Braverman. Finding Low Error Clusterings. In *COLT*, volume 3, pages 3–4, 2009.

15 Maria-Florina Balcan, Nika Haghtalab, and Colin White. k-center Clustering under Perturbation Resilience. In *Proceedings of the Annual International Colloquium on Automata, Languages, and Programming (ICALP)*, 2016.

16 Maria Florina Balcan and Yingyu Liang. Clustering under perturbation resilience. *SIAM Journal on Computing*, 45(1):102–155, 2016.

17 Maria-Florina Balcan, Heiko Röglin, and Shang-Hua Teng. Agnostic Clustering. In *ALT*, pages 384–398. Springer, 2009.

18 Maria-Florina F Balcan, Steven Ehrlich, and Yingyu Liang. Distributed k-means and k-median Clustering on General Topologies. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1995–2003, 2013.

19 Ziv Bar-Yossef, Thathachar S Jayram, Ravi Kumar, and D Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004.

20 Mohammadhossein Bateni, Aditya Bhaskara, Silvio Lattanzi, and Vahab Mirrokni. Distributed Balanced Clustering via Mapping Coresets. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 2591–2599, 2014.

21 Yonatan Bilu and Nathan Linial. Are stable instances easy? *Combinatorics, Probability and Computing*, 21(05):643–660, 2012.

22 Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. A tight bound for set disjointness in the message-passing model. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 668–677. IEEE, 2013.

23 Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proceedings of the Annual Symposium on Discrete Algorithms (SODA)*, pages 737–756. SIAM, 2015.

24 Moses Charikar, Sudipto Guha, Éva Tardos, and David B Shmoys. A constant-factor approximation algorithm for the k-median problem. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, pages 1–10. ACM, 1999.

25 Moses Charikar, Samir Khuller, David M Mount, and Giri Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the Annual Symposium on Discrete Algorithms (SODA)*, pages 642–651. Society for Industrial and Applied Mathematics, 2001.

26 Chandra Chekuri and Shalmoli Gupta. Perturbation Resilient Clustering for k-Center and Related Problems via LP Relaxations. In *Proceedings of the International Workshop on Approximation, Randomization, and Combinatorial Optimization Algorithms and Techniques (APPROX-RANDOM)*, 2018.

27 Jiecao Chen, He Sun, David Woodruff, and Qin Zhang. Communication-Optimal Distributed Clustering. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3720–3728, 2016.

28 Ke Chen. A constant factor approximation algorithm for k-median clustering with outliers. In *Proceedings of the Annual Symposium on Discrete Algorithms (SODA)*, volume 8, pages 826–835, 2008.

29 Michael B Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 163–172. ACM, 2015.

30 Travis Dick, Mu Li, Venkata Krishna Pillutla, Colin White, Maria Florina Balcan, and Alex Smola. Data Driven Resource Allocation for Distributed Learning. In *"Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)"*, 2017.

31 Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

32 Sudipto Guha, Yi Li, and Qin Zhang. Distributed Partial Clustering. In *Proceedings of the Symposium on Parallelism in Algorithms and Architectures (SPAA)*, 2017.

33 Rishi Gupta, Tim Roughgarden, and C Seshadhri. Decompositions of triangle-dense graphs. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 471–482. ACM, 2014.

34 Lingxiao Huang, Shaofeng Jiang, Jian Li, and Xuan Wu. Epsilon-coresets for clustering (with outliers) in doubling metrics. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 814–825. IEEE, 2018.

35 Zengfeng Huang, Bozidar Radunovic, Milan Vojnovic, and Qin Zhang. Communication complexity of approximate matching in distributed graphs. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 30. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

36 Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. A local search approximation algorithm for k-means clustering. In *Proceedings of the eighteenth annual symposium on Computational geometry*, pages 10–18. ACM, 2002.

37 Ari Kobren, Nicholas Monath, Akshay Krishnamurthy, and Andrew McCallum. A Hierarchical Algorithm for Extreme Clustering. In *Proceedings of the KDD International Conference on Knowledge Discovery and Data Mining*, pages 255–264. ACM, 2017.

38 Ravishankar Krishnaswamy, Shi Li, and Sai Sandeep. Constant Approximation for k-Median and k-Means with Outliers via Iterative Rounding. *CoRR*, abs/1711.01323, 2017. `arXiv:1711.01323`.

39 Amit Kumar and Ravindran Kannan. Clustering with spectral norm and the k-means algorithm. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 299–308. IEEE, 2010.

40 Shi Li and Xiangyu Guo. Distributed k-Clustering for Data with Heavy Noise. In *Advances in Neural Information Processing Systems*, pages 7849–7857, 2018.

41 Konstantin Makarychev, Yury Makarychev, Maxim Sviridenko, and Justin Ward. A bi-criteria approximation algorithm for k Means. In *APPROX*, 2016.

42 Gustavo Malkomes, Matt J Kusner, Wenlin Chen, Kilian Q Weinberger, and Benjamin Moseley. Fast Distributed k-Center Clustering with Outliers on Massive Data. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1063–1071, 2015.

43 Rafail Ostrovsky, Yuval Rabani, Leonard J Schulman, and Chaitanya Swamy. The effectiveness of lloyd-type methods for the k-means problem. *Journal of the ACM (JACM)*, 59(6):28, 2012.

44 Kim D Pruitt, Tatiana Tatusova, Garth R Brown, and Donna R Maglott. NCBI Reference Sequences (RefSeq): current status, new features and genome annotation policy. *Nucleic acids research*, 2011.

45 Alexander A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992.

46 Konstantin Voevodski, Maria-Florina Balcan, Heiko Röglin, Shang-Hua Teng, and Yu Xia. Min-sum clustering of protein sequences with limited distance information. In *International Workshop on Similarity-Based Pattern Recognition*, pages 192–206. Springer, 2011.