Robust Distributed Clustering with Redundant Data Assignment

Saikiran Bulusu*, Venkata Gandikota†, Arya Mazumdar‡, Ankit Singh Rawat§ and Pramod K. Varshney†
*Electrical & Computer Engineering Department, Ohio State University, Columbus, OH 43210, bs.kiran.bulsai@gmail.com
*Electrical Engineering & Computer Science Department, Syracuse University, Syracuse, NY 13202, gandikota.venkata@gmail.com, varshney@syr.edu
†The Halıcıoğlu Data Science Institute (HDSI), University of California, San Diego, arya@ucsd.edu
‡Google Research NY, New York, NY 10011, ankitsrawat@google.com

***Abstract*—In this work, we present distributed clustering algorithms that can handle large-scale data across multiple machines in the presence of faulty machines. These faulty machines can either be straggling machines that fail to respond within a stipulated time or Byzantines that send arbitrary responses. We propose redundant data assignment schemes that enable us to obtain clustering solutions based on the entire dataset, even when some machines are stragglers or adversarial in nature. Our proposed robust clustering algorithms generate a constant factor approximate solution in the presence of stragglers or Byzantines. We also provide various constructions of the data assignment scheme that provide resilience against a large fraction of faulty machines. Simulation results show that the distributed algorithms based on the proposed assignment scheme provide good-quality solutions for a variety of clustering problems.**

I. Introduction

Clustering is one of the basic unsupervised learning tasks used to infer informative patterns in data. The goal of clustering algorithms is to find a subset of data points, called cluster centers, that provide a good representation of the given dataset. The cluster centers provide a partition of the given set of data points that maximize similarity within a group and minimize similarity across the groups. The quality of the clusters is measured using a cost function of which, the k-means and k-median are the most commonly used. The k-median (k-means) clustering problem aims to find a set of k centers that minimize the sum of the distances (sum of the squared distances) of the individual points to their closest cluster center. Since computing an optimal clustering solution is an NP-hard problem [3], we will focus on approximate solutions that aim to find a set of k-centers whose cost is at most a small constant factor larger than the optimal [4].

Most widely used centralized clustering algorithms assume that the entire data fits in a single machine and are no longer desirable with the increasing size of the datasets. Hence, there has been a significant interest in designing efficient distributed algorithms for the clustering problem. The goal is to design algorithms that can work with multiple machines having access only to their respective local datasets. Under the data-distributed setup, we assume one fusion center (FC) and m machines such that the dataset P consisting of n data points is partitioned arbitrarily and distributed across the machines. We denote these partitions by $\{P_1, \ldots, P_m\} \subseteq P$ and assign each of these subsets to a different machine. The individual machines perform computation on the locally available data points and transmit the obtained results to the FC. The FC then aggregates these results to obtain the final clustering result. Recent works have provided clustering algorithms in such data-distributed setup with provably constant factor approximate solutions [5]–[10].

Although the distributed model of computation improves computational efficiency, it makes the system vulnerable to faulty machines. The faulty machines may send information with delay, may completely crash, or may send arbitrary (possibly adversarial) information, thereby drastically affecting the quality of the computed solutions. In this work, we consider two kinds of faulty machines which (i) may send information with delay (or not send anything at all) (*stragglers*), or (ii) may send arbitrary information (*Byzantines*).

Clustering with Stragglers: The stragglers correspond to the machines that take significantly more time than expected to respond. Several issues could lead to this behavior in the machines, like power outages, congested communication networks, or software updates running on the machines. One naïve approach to handling straggling machines in certain distributed tasks is to ignore them or rely on asynchronous methods. There are established tradeoffs between the loss of information due to ignoring the stragglers and the efficiency of specific tasks such as computing distributed gradients [11]–[16]. However, considering the presence of stragglers in distributed clustering has received much less attention.

Clustering with Byzantines: Another challenge in the distributed setup is the presence of adversarial machines, also known as Byzantines [17]. An adversarial attack usually has the ability to influence the centers in one (or more) of the clusters. Instead of sending the correct result of the computation to the FC, a Byzantine may send arbitrary values. A naïve approach is to rely on simple distributed clustering methods even when Byzantines are present [5], [6]. However, this may lead to extremely poor-quality solutions being computed by the distributed clustering algorithm due to the arbitrary information being sent by the Byzantines. Another approach

Dr. Mazumdar was supported by National Science Foundation under Grants 2217058 and 2112665.

is to provide filters to identify and remove the Byzantines in the setup as proposed in the Byzantine machine learning literature [18]–[22].

An alternate solution, that we adopt, is to introduce redundancy in the data distributed to the machines. This ensures that the information obtained from a subset of machines is sufficient to compute the desired function on the entire dataset. Multiple coding-based redundant data distribution schemes have been proposed to mitigate the effect of stragglers [12], [13], [23]–[26] and Byzantines [27]–[30] for computing linear functions such as gradient aggregation in first-order optimization methods. However, these techniques do not translate well for clustering tasks where, unlike the prior works, the responses from different machines may not be related.

In this work, we propose a data distribution scheme for distributed clustering problems in the presence of stragglers and Byzantines. The stragglers send the correct information albeit with a delay. Hence, the FC knows the identities of the stragglers. However, in the case of clustering with Byzantines, the formulation deals with a more general scenario where a subset of the machines are adversarial and can send arbitrary information. Moreover, the identity of these adversarial machines (Byzantines) is not known to the FC which constitutes the main bottleneck in obtaining Byzantine resilient clustering algorithms. We show that our proposed data distribution scheme allows us to compute provably good-quality cluster centers even in the presence of a relatively large number of stragglers and/or Byzantines.

A. Our Results

In this work, we provide robust distributed clustering approaches that generate solutions with a cost at most $c \cdot$OPT, for a small approximation factor $c \geq 1$, where, OPT denotes the cost of the best clustering solution. Our algorithms are resilient to machines that are either (i) stragglers, or (ii) Byzantines.

We propose a redundant data distribution scheme that allocates a data point to multiple machines to mitigate the loss of information (or misinformation) that arises due to the presence of faulty machines. Following are our major contributions.

- We establish sufficient conditions on the data assignment scheme that enables us to mitigate the effect of stragglers and Byzantines to compute good-quality clusters (Property III.1 and Property III.2).
- We design robust k-median and k-means clustering approaches that generate a constant factor approximate solution in the presence of stragglers. Our approach also extends to a more general class of squared ℓ_2-fitting problems known as subspace clustering. Theorem IV.2 shows that we can achieve an approximation factor of roughly 3 for k-median clustering. This approach extends naturally to k-means clustering (Theorem IV.3) and gives an approximation factor close to 10 for k-means. The results for k-means can be improved and generalized to subspace clustering using a slight variant of the above approach which is formalized in Theorem IV.5.
- Byzantines are much harder to handle since their identity is unknown. Using a stronger data assignment scheme compared to its straggler counterpart (Remark 1), we obtain k-median (Theorem V.2) and k-means (Theorem V.6) approaches that are guaranteed to achieve approximation factors close to 3 and 10 respectively. We also improve upon the suggested algorithms to make them computationally and storage efficient in Theorems V.3 and Theorem V.7.
- We provide various constructions of the assignment scheme that satisfy the established conditions (Properties III.1 and III.2) and provide resilience against a large fraction of faulty machines while incurring little redundancy.
- We also consider the random straggler model motivated by practical applications to obtain better trade-offs between the load per machine and the number of faults tolerable. The various constructions and the tradeoffs they present are summarized in Table I, where m are the number of machines, n are the total number of points for clustering, and t are the number of stragglers/Byzantines.
- Simulation results illustrate the excellent performance of our algorithm.

	Construction	Fault model	Load per machine
Thm VI.1	Random	Adversarial	$O(\frac{nt \log m}{(m-t)})$
Thm VI.2	Explicit	Adversarial	$O(\frac{n \log m}{m})$
Thm VII.1	Random	Random	$O(\frac{n \log (n)}{m-t})$
Thm VII.2	Explicit	Random	$O(\frac{n \log m}{m-t})$

TABLE I: Summary of constructions of data assignment schemes. Number of machines (m), Total number of data points (n), number of faulty machines (t).

Setting	Approximation Factor
Straggler resilient k-median (Thm IV.2)	$3(1+\delta)$
Straggler resilient k-means (Thm IV.3)	$10(1+\delta)$
Straggler resilient (r,k)-subspace clustering (Thm IV.5)	$(1+4\delta)$
Byzantine resilient k-median (Thm V.2)	$3(1+\delta)$
Improved Byzantine resilient k-median (Thm V.3)	$\left(\frac{2}{1-1/k}+\frac{1}{1-\delta}\right)(1+3\delta)$
Byzantine resilient k-means (Thm V.6)	$10(1+\delta)$
Improved Byzantine resilient k-means (Thm V.7)	$\left(\frac{8}{1-1/k}+\frac{2}{1-\delta}\right)(1+3\delta)$

TABLE II: Approximation factors ($\delta > 0$).

B. Comparison with our Previous Works

In previous works [1], [2], we assumed that the machines had the ability to compute exact solutions to the clustering problem on the local datasets. In this work, we consider the case when the machines can no longer compute the exact solution to the clustering problem. This reduces the computational load at each machine with an increased approximation factor (see Sections IV and V).

Moreover, in [2], we assumed that the FC computes the local summaries to evaluate the quality of the data sent by each local machine. Hence, the FC required the access to the entire dataset P and had to estimate the cost of computing a cluster on the local dataset P_i using the summaries sent by each machine. In resource-constrained settings, such assumptions can increase the computational load at the FC. In this work, we assume that the FC does not have access to the entire dataset, and hence, can not estimate the cost of computing a cluster

on the local dataset P_i using the summaries sent by each machine. Hence, the analyses in [2] can no longer be utilized. The first challenge in this work is the computation of coresets by the FC as surrogates of the respective local datasets P_i. These are efficiently computed in a streaming fashion using the sensitivity sampling technique [31]. We utilize these coresets to approximate the cost of clustering using the local datasets P_i at the FC. Another challenge is the filtration step. In [2], the filtering step depended on the cost of clustering computed on the local datasets P_i. However, in this work, the filtering is performed utilizing the cost of clustering using the coresets computed by the FC and the local summaries sent by the machines. Moreover, in [2], the weights for each of the points in the summaries sent by the machines were obtained on the respective local datasets P_i. However, crucially in this work, the weights for each of the points in the local summaries sent by the machines are estimated using the coresets computed by the FC. Therefore, the third challenge is the estimation of these weights. We show that these estimated weights are a constant factor away from their intended value with high probability. With these challenges, the analysis is no longer straightforward and needs additional tools (see Sections V-B and V-C). Thus, as described in Section I-A, our work in this paper is much more general and extends our prior work [1], [2] quite significantly.

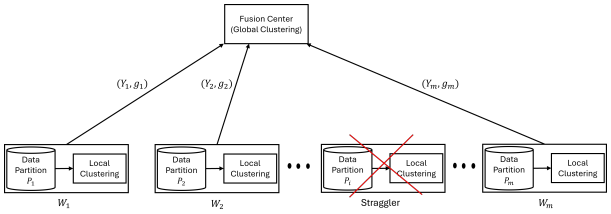
C. Outline and Notation

The system model is described in Section II. Sufficient conditions on the data assignment scheme are presented in Section III. The proposed algorithm for straggler resilience is given in Section IV. The k-median clustering problem in the presence of stragglers is considered in Section IV-A and extended for k-means clustering in Section IV-B. An improvement and generalization of the k-means algorithm to (r, k)-subspace clustering is presented in Section IV-C. The algorithm for Byzantine resilience is given in Section V. The k-median clustering problem is considered in Section V-A. In Section V-B, we present a computationally efficient version of the Byzantine resilient clustering algorithms. Here, we address a few drawbacks of the previous approach and make the algorithm more suitable for practice. These algorithms are then extended to obtain Byzantine-resilient k-means algorithm in Section V-C. Constructions of assignment matrices are presented in Section VI. Extensions of the results to random straggler model are presented in Section VII. Simulation results are provided in Section VIII, followed by our conclusions in Section IX.

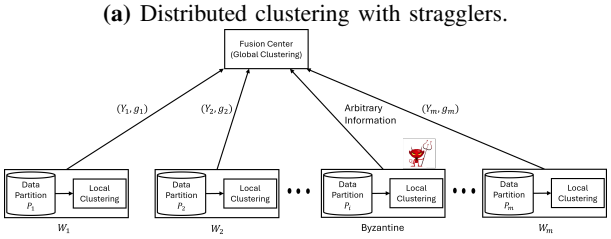
Notations: All vectors are denoted in boldface. We have $[n] = \{1, \ldots, n\}$, and $\mathbf{1}_n$ denotes a vector of all 1's of length n. $d(\mathbf{x}, \mathbf{y})$ denotes the Euclidean distance between two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

II. System Model

Given a dataset with n points $P = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\} \subseteq \mathbb{R}^d$, distributed among m machines, the goal in clustering is to find a set of k cluster centers $C = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\} \subseteq \mathbb{R}^d$ that closely represent the entire dataset. The quality of these



(a) Distributed clustering with stragglers.



(b) Distributed clustering with Byzantines.

Fig. 1: System Model. (Identities of the Byzantines are not known to the FC.)

centers is usually measured by a cost function $\text{cost}(P, C)$. For k-median, the cost function is defined as $\text{cost}(P, C) = \sum_{\mathbf{x} \in P} d(\mathbf{x}, C)$, where $d(\mathbf{x}, C) := \min_{\mathbf{c} \in C} d(\mathbf{x}, \mathbf{c})$. The k-means cost function for clustering is given by $\text{cost}(P, C) = \sum_{\mathbf{x} \in P} d^2(\mathbf{x}, C)$. If the dataset P is weighted with an associated non-negative weight function $g : P \to \mathbb{R}^+$, the k-median cost for the weighted dataset (P, g) is then defined as $\text{cost}(P, g, C) = \sum_{\mathbf{x} \in P} g(\mathbf{x}) d(\mathbf{x}, C)$. The k-means cost for (P, g, C) is defined analogously. Our goal is, therefore, to obtain a set of k centers C that minimizes $\text{cost}(P, C)$. For any data point $\mathbf{x} \in P$, and any set of centers C, we denote its cluster center by $C(\mathbf{x}) := \arg\min_{\mathbf{c} \in C} d(\mathbf{x}, \mathbf{c})$. Also, for any point set P, we denote the cluster of P associated with a center $\mathbf{c} \in C$ by $\text{cluster}(\mathbf{c}, P) := \{\mathbf{x} \in P | C(\mathbf{x}) = \mathbf{c}\}$.

We consider the data-distributed clustering framework with m machines $W_1, \ldots, W_m$. Let $P_i \subseteq P$ be the set of points assigned to the machine W_i. To compute the cluster centers in such data-distributed setups, the machines transmit a summary of their local data to the fusion center (FC). For the simplicity of presentation, we assume each machine computes the optimal clustering solution on its assigned data points.

Problem Statement: In this paper, the main goal is to design data-distributed robust clustering approaches. Specifically, given a dataset P of n points in $\mathbb{R}^d$, and distributed setup with m machines where at most t machines are faulty (either stragglers or Byzantines), the goal is to design a clustering approach that generates a solution with the cost at most $c \cdot \text{OPT}$, with a small approximation factor $c \geq 1$ for the k-median and the k-means clustering problems. Here OPT denotes the optimal cost of clustering the entire dataset.

We will consider the following two models of faults:

► Adversarial Fault Model: In this model, we assume that any arbitrary set of at most t machines can be faulty (either stragglers or Byzantines).

► Random Fault Model: In this model, we assume that each machine can behave as a straggler (or Byzantine) independently with some fixed probability. This model is more applicable in real-world settings to model faults as network

congestion or job scheduling on individual machines can be considered to be an independent stochastic process.

While the focus of this work is on the adversarial fault model, some of our results also extend to random faults model (see Section VII for details).

In the presence of stragglers, the FC combines the local summaries obtained from non-straggler machines to obtain the summary of the global dataset which gives a constant factor approximate solution (Fig. 1a). To mitigate the effect of Byzantines, the FC ranks the received local summaries to evaluate the quality of the data summary sent by each local machine. An approximate solution to the clustering problem can then be computed at the FC by aggregating the subset of best summaries (Fig. 1b). We note that the identity of the Byzantine machines is not known to the FC. Similar to the result of [32], we show that the set of k-centers computed by the machines summarizes their local dataset. Our results also extend trivially when machines provide *approximate* clustering solutions as a summary.

Next, we provide some definitions and results that are helpful for the presentation in the rest of this paper.

A. Preliminaries

Definition II.1 ((r,k)-subspace clustering)**.** Given a dataset $P \subset \mathbb{R}^d$ find a set of k-subspaces (linear or affine) $\mathcal{L} = \{L_i\}_{i=1}^k$, each of dimension r, that minimizes $\text{cost}(P,\mathcal{L}) := \sum_{i=1}^n \min_{L\in\mathcal{L}} d^2(\mathbf{p}_i, L)$.

Note that for $r = 0$, this is exactly the k-means problem described above. Another special case, when $k = 1$, is known as principal component analysis (PCA). Another closely related problem is the k-medians problem defined as follows:

Definition II.2 (k-medians clustering)**.** Given a dataset $P \subset \mathbb{R}^d$ find a set of k-centers $C = \{\mathbf{c}_i\}_{i=1}^k$, each of that minimizes $\text{cost}(P,C) := \sum_{i=1}^n \min_{\mathbf{c}\in C} d(\mathbf{p}_i, L)$.

For any $\alpha \geq 1$, we define an α-approximate solution to a clustering problem with cost function defined by $\text{cost}(\cdot,\cdot)$ as follows:

Definition II.3 (α-approximate solution)**.** For any $\alpha > 1$, the set of k cluster centers C, $|C| = k$, is an α-approximate solution to the k-center clustering problem if the cost of clustering P with C, $\text{cost}(P,C)$, is at most α times the cost of clustering with optimal set k-centers, $\text{cost}(P,C) \leq \alpha \cdot \text{OPT}$.

The quality of the data summaries is captured by the notion of a coreset. Informally, a coreset is a small weighted set of representative points of the dataset that closely approximates the cost of clustering on any set of k centers.

Definition II.4 (ϵ-coreset, [33])**.** Let $\epsilon \geq 0$. For a dataset P, an ϵ-coreset with respect to a cost function $\text{cost}(\cdot,\cdot)$ is a weighted dataset S with an associated weight function $g : S \to \mathbb{R}^+$ such that, for any set of k centers C, we have

$$(1-\epsilon)\text{cost}(P,C) \leq \text{cost}(S,g,C) \leq (1+\epsilon)\text{cost}(P,C).$$

Using any off-the-shelf α-approximate solution to the clustering problem on an ϵ-coreset of the dataset P yields a good approximate solution on the entire dataset. This fact is formalized by the following Theorem.

Theorem II.1 ([34])**.** *Let (S,g) be an ϵ-coreset for a dataset P with respect to the cost function $\text{cost}(\cdot,\cdot)$. Any α-approximate solution to the clustering problem on input S, is an $\alpha(1+3\epsilon)$-approximate solution to the clustering problem on P.*

Next, we present our approach to assigning data to different machines with redundancy.

III. Data Assignment

The first step to obtaining robust distributed clustering in the presence of faulty machines is the initial data assignment to the machines. Specifically, every data point in the dataset P is mapped to multiple machines by carefully employing redundancy in the assignment process. Hence, each data point affects the local computation performed on multiple machines and the final clusters at the FC are obtained by taking into account the contributions of most of the data points in P even though some of the machines are faulty. We introduce the data assignment scheme along with the resilience properties below. This property enables the aggregation of local computations from the non-straggling or honest machines at the FC and preserves the relevant information present in the dataset P for the clustering problems. The assignment scheme is utilized to obtain good-quality solutions to k-median and k-means clustering.

A. Straggler-resilient Data Assignment

Let $A \in \{0,1\}^{m\times n}$ be the binary assignment matrix where the i-th row, $\mathbf{a}_i$, indicates the set $P_i \subseteq P$ of points assigned to machine W_i. Let $\mathcal{R} \subset [m]$ denote the set of non-straggler machines. We assume that $|\mathcal{R}| \geq m-t$, where $t < m$ denotes an upper bound on the number of stragglers in the system. For any such set of non-straggler machines $\mathcal{R}$, we require the assignment matrix A to satisfy the following property.

Property III.1 ((t,δ)-Straggler resilience property)**.** Let $\delta > 0$ be a given constant. The assignment matrix $A \in \{0,1\}^{m\times n}$ has (t,δ)-straggler resilience if for every subset of $m-t$ rows $\mathcal{R} \subseteq [m]$, $\exists$ a recovery vector, $\mathbf{b} = (b_1,\ldots,b_{|\mathcal{R}|})^T \in \mathbb{R}^{|\mathcal{R}|}, b_i > 0, \forall i \in |\mathcal{R}|$, such that for all $i \in [n]$,

$$\mathbf{1}_n^T \leq \sum_{i\in\mathcal{R}} b_i \mathbf{a}_i \leq (1+\delta)\mathbf{1}_n^T, \tag{1}$$

where $\leq$ indicates coordinate-wise inequality.

We remark that the straggler resilience property is significantly different from that in [12] where the property utilizes the fact that the gradients are related to each other across different machines. For instance, in gradient coding, the recovery vector $\mathbf{b}$ can be arbitrary, whereas, we require it to be strictly non-negative. Furthermore, for gradient coding, the error is measured in terms of ℓ_2 norm whereas we need an ℓ_∞ bound on the error.

Utilizing the combinatorial characterization for the assignment scheme given by Property III.1, the information received

from the non-straggler machines can be combined to generate close to optimal clustering solutions using the following result.

Lemma III.1. *Let $P \subset \mathbb{R}^d$ be a dataset distributed across m machines using a (t, δ)-straggler resilient assignment matrix A that satisfies Property III.1 . Let $\mathcal{R}$ be any set of $m - t$ machines. For any $\delta > 0$, let $\boldsymbol{b} \in \mathbb{R}^{|\mathcal{R}|}$ be the recovery vector corresponding to $\mathcal{R}$. Then, for any set of k centers $C \subset \mathbb{R}^d$, any weight function $g : P \to \mathbb{R}$,*

$$\textit{cost}(P, g, C) \leq \sum_{i \in \mathcal{R}} b_i \textit{cost}(P_i, g, C) \leq (1 + \delta)\textit{cost}(P, g, C).$$

Proof. Here, we prove the result for the $d^2(\cdot, \cdot)$ cost function, and the proof extends similarly to $d(\cdot, \cdot)$ as well. The proof is independent of the choice of the distance function, and we only use properties of the assignment matrix. First note that,

$$\begin{aligned}\sum_{i \in \mathcal{R}} b_i \text{cost}(P_i, g, C) &= \sum_{i \in \mathcal{R}} b_i \sum_{\mathbf{p} \in P_i} g(\mathbf{p}) d^2(\mathbf{p}, C) \\ &= \sum_{i \in \mathcal{R}} b_i \sum_{j \in [n]} A_{i,j} g(\mathbf{p}_j)\ d^2(\mathbf{p}_j, C) \\ &= \sum_{j \in [n]} g(\mathbf{p}_j)\ d^2(\mathbf{p}_j, C) \sum_{i \in \mathcal{R}} b_i A_{i,j}. \end{aligned} \tag{2}$$

From Property III.1 we know that for any $j \in [n]$, $\sum_{i \in \mathcal{R}} b_i A_{i,j} \leq 1 + \delta$. Combining this fact with (2), we obtain

$$\begin{aligned}\sum_{i \in \mathcal{R}} b_i \text{cost}(P_i, g, C) &= \sum_{j \in [n]} g(\mathbf{p}_j)\ d^2(\mathbf{p}_j, C) \sum_{i \in \mathcal{R}} b_i A_{i,j} \\ &\leq (1 + \delta) \sum_{j \in [n]} g(\mathbf{p}_j)\ d^2(\mathbf{p}_j, C) \\ &= (1 + \delta) \cdot \text{cost}(P, g, C)\end{aligned}$$

Similarly, Property III.1 ensures that for any $j \in [n]$, $\sum_{i \in \mathcal{R}} b_i A_{i,j} \geq 1$. Utilizing this fact in (2) gives us the desired lower bound as follows.

$$\begin{aligned}\sum_{i \in \mathcal{R}} b_i \text{cost}(P_i, g, C) &= \sum_{j \in [n]} g(\mathbf{p}_j)\ d^2(\mathbf{p}_j, C) \sum_{i \in \mathcal{R}} b_i A_{i,j} \\ &\geq \sum_{j \in [n]} g(\mathbf{p}_j)\ d^2(\mathbf{p}_j, C) = \text{cost}(P, C).\end{aligned}$$

□

B. Byzantine-resilient Data Assignment

Similar to the straggler resilient data assignment, we propose a modified data assignment to the machines to mitigate the effect of Byzantines. Let $A \in \{0, 1\}^{m \times n}$ denote the binary assignment matrix whose i-th row, $\mathbf{a}_i$, indicates the set $P_i \subseteq P$ of points assigned to machine W_i. Let $\mathcal{R} \subset [m]$ denote the set of honest (non-Byzantine) machines. We assume that $|\mathcal{R}| \geq m - t$, where $t < m$ denotes an upper bound on the number of Byzantines in the system. For any such set of honest machines $\mathcal{R}$, we require the assignment matrix A to satisfy the following property.

Property III.2 ((t, δ)-Byzantine resilience property)**.** Let $\delta > 0$ be a given constant. The assignment matrix $A \in \{0, 1\}^{m \times n}$ has (t, δ)-Byzantine resilience if $\exists$ a reconstruction coefficient $\rho > 0$, such that for any subset of $m - t$ rows $\mathcal{R} \subseteq [m]$,

$$\mathbf{1}_n^T \leq \rho \sum_{i \in \mathcal{R}} \mathbf{a}_i \leq (1 + \delta)\mathbf{1}_n^T, \tag{3}$$

where $\leq$ indicates coordinate-wise inequality.

Remark 1. The Byzantine-resilience property is much stronger than the straggler resilience property introduced in Property III.1. For straggler resilience, it is sufficient to have some non-negative linear combination of the rows (corresponding to the non-straggler machines) that is close to the all-ones vector. However, for Byzantine resilience, we need all these linear combinations to be uniform and non-negative. Further, we also need this reconstruction factor to be the same across all subsets of Byzantines. While the latter condition is not strictly needed, it simplifies the proofs and the algorithm.

The information received from the honest machines is combined using the following lemma to generate a close-to-optimal clustering solution.

Lemma III.2. *Let $\mathcal{R} \subseteq [m]$ be any set of $m - t$ indices. Let ρ be the reconstruction coefficient of the (t, δ)-Byzantine resilient assignment matrix. Then, for any set of centers C, we have $\textit{cost}(P, C) \leq \sum_{i \in \mathcal{R}} \rho\ \textit{cost}(P_i, C) \leq (1 + \delta)\textit{cost}(P, C)$.*

Proof. The proof is analogous to that of Lemma III.1 and follows based on the combinatorial characterization for the assignment scheme enforced by Property III.2. □

IV. Straggler Resilient Clustering

In this section, we present straggler resilient clustering techniques using the redundant data distribution scheme described above. In Section IV-A, we present the k-median clustering algorithm that is extended in a straightforward manner to the k-means setting in Section IV-B. This algorithm is improved in Section IV-C, where we present a general algorithm for the (r, k)-subspace clustering.

A. Straggler-Resilient Distributed k-median Clustering

Dataset P is distributed across m machines using a (t, δ)-straggler resilient assignment matrix A that satisfies Property III.1. Each non-straggling machine sends a set of weighted k-median centers of their local datasets which when aggregated at the FC gives a summary for the entire dataset. Hence, the weighted k-median clustering on this summary at the FC provides a good-quality solution for the entire dataset P. In Algorithm 1, we provide the above-discussed steps in detail.

Before we state the theorem that quantifies the quality of the clustering solution $\hat{C}$ provided by Algorithm 1 on the entire dataset P, we present the following lemma where we show that the cost incurred by the weighted dataset Y is close to the cost incurred by P for any set of k centers C, which is necessary to prove the theorem.

[1]In general, if $\mathbf{y} \in Y_{i_1} \cap Y_{i_r} \cap \ldots \cap Y_{i_r}$, then $g(\mathbf{y}) = \sum_{j=1}^{r} b_{i_j} g_{i_j}(\mathbf{y})$.

Algorithm 1 Straggler-resilient distributed k-median clustering

1: **Initialize:** A collection of n data points $P \subset \mathbb{R}^d$
2: Allocate P to m machines according to a (t, δ)-straggler resilient matrix A.
3: Let $P_i \subset P$ be the set of points assigned to machine W_i
4: Each machine W_i computes a k-median solution, Y_i, on set P_i.
5: Define $g_i : Y_i \to \mathbb{R}$ as $g_i(\mathbf{y}) = |\text{cluster}(\mathbf{y}, P_i)|$, for every $y \in Y_i$
6: FC collects $\{(Y_i, g_i)\}_{i \in \mathcal{R}}$ from the non-straggling machines, for some $\mathcal{R} \subseteq [m], |\mathcal{R}| \geq m - t$
7: Let $Y = \bigcup_{i \in \mathcal{R}} Y_i$. Using the recovery vector $\mathbf{b}$, define $g : Y \to \mathbb{R}$ such that $g(\mathbf{y}) = b_i g_i(\mathbf{y}), \forall \mathbf{y} \in Y_i$ and $i \in \mathcal{R}$[1]
8: **Return** $\hat{C}$, the k-median solution on (Y, g).

Lemma IV.1. *For k-median clustering, for any set of k-centers $C \subset \mathbb{R}^d$, we have*

$$\textit{cost}(P, C) - \sum_{i \in \mathcal{R}} b_i \textit{cost}(P_i, Y_i) \leq \textit{cost}(Y, g, C) \leq 2(1 + \delta)\textit{cost}(P, C).$$

Proof of Lemma IV.1 is presented in Appendix, Section A.

Theorem IV.2. *Let C^* be the optimal set of k-median centers for dataset P. Then, Algorithm 1 on dataset P returns a set of centers $\hat{C}$ such that $\textit{cost}(P, \hat{C}) \leq 3(1 + \delta)\textit{cost}(P, C^*)$.*

Proof of Theorem IV.2. Utilizing the lower bound from Lemma IV.1 with $C = \hat{C}$, we have

$$\begin{aligned} \text{cost}(P, \hat{C}) &\leq \text{cost}(Y, g, \hat{C}) + \sum_{i \in \mathcal{R}} b_i \text{cost}(P_i, Y_i) \\ &\overset{(a)}{\leq} \text{cost}(Y, g, C^*) + \sum_{i \in \mathcal{R}} b_i \text{cost}(P_i, C^*) \\ &\overset{(b)}{\leq} 2(1 + \delta)\text{cost}(P, C^*) + (1 + \delta)\text{cost}(P, C^*), \end{aligned} \tag{4}$$

where (a) follows from the fact that $\hat{C}$ and Y_i are the optimal set of centers for the weighted dataset (Y, g) and the partial dataset P_i, respectively. For (b), we utilize the upper bound in Lemma IV.1 and Lemma III.1 with $C = C^*$. □

Note that the summary computed at the FC uses the weighted set (Y, g) which is constructed only from the information sent by the non-straggling nodes. Also, the data assignment scheme initially used to distribute the data satisfies the Property III.1. Hence, from Theorem IV.2, we observe that the FC is able to construct a good summary of the entire dataset P despite the presence of the stragglers. Moreover, this summary is sufficient to generate a good quality k-median clustering solution corresponding to P, i.e., the summary generates a constant factor approximate solution.

Remark 2. Suppose the honest machines and the FC are unable to compute the exact k-median clustering solution as required in Step 4 and Step 8 of Algorithm 1, but instead produce an α-approximate solution (such as in [35]), then this slight variant of Algorithm 1 returns a set of k-centers $\hat{C}$ such that $\text{cost}(P, \hat{C}) \leq \alpha(1+\delta)(2+\alpha)\text{cost}(P, C^*)$, even in the presence of t stragglers. See Appendix A for the complete proof.

Remark 3 (Time Complexity). The workers and the FC may use the $O(1)$-approximate k-medians clustering algorithm of [36] that runs in time $O(|P_i|d)$ at the worker nodes, and $O(mkd)$ time at the FC.

B. Straggler-Resilient Distributed k-means Clustering

Observe that the above-described algorithms for distributed k-median clustering in the presence of stragglers can be generalized to other classical cost functions to yield algorithms such as for the k-means clustering algorithm. The key observation that we use to extend the above-described algorithms is that the distance function $d^2(\cdot, \cdot)$ satisfies a scaled version of the triangle inequality, i.e., for any $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^d$,

$$d^2(\mathbf{a}, \mathbf{b}) \leq 2(d^2(\mathbf{a}, \mathbf{c}) + d^2(\mathbf{b}, \mathbf{c})). \tag{5}$$

We use a strategy similar to Algorithm 1 to compute the k-means clustering solution in the presence of stragglers. We observe that if each local machine can compute an exact (or approximate) k-means solution on their local datasets, then it can be suitably combined using the recovery vector to obtain a constant factor approximate k-means solution to the global dataset. Algorithm 2, does exactly this, where in Step 4 of Algorithm 1, each machine W_i sends a k-means solution Y_i corresponding to P_i weighted accordingly.

Algorithm 2 Straggler-resilient distributed k-means clustering

1: **Initialize:** A collection of n data points $P \subset \mathbb{R}^d$
2: Allocate P to m machines according to a (t, δ)-straggler resilient matrix A.
3: Assign the set of points $P_i \subset P$ to machine W_i
4: Each machine W_i computes an α-approximate k-means solution Y_i on set P_i. Let $g_i : Y_i \to \mathbb{R}$ as $g_i(\mathbf{y}) = |\text{cluster}(\mathbf{y}, P_i)|$, for every $y \in Y_i$
5: FC collects $\{(Y_i, g_i)\}_{i \in \mathcal{R}}$ from the non-straggling machines
6: Let $Y = \bigcup_{i \in \mathcal{R}} Y_i$. Using the recovery vector $\mathbf{b}$, define $g : Y \to \mathbb{R}$ such that $g(\mathbf{y}) = b_i g_i(\mathbf{y}), \forall \mathbf{y} \in Y_i$ and $i \in \mathcal{R}$
7: **Return** $\hat{C}$, the α-approximate k-means solution on (Y, g).

The performance guarantees of Algorithm 2 can be stated as follows:

Theorem IV.3. *Let C^* be the optimal set of k-means centers for dataset P. Then, Algorithm 2 on dataset P returns a set of centers $\hat{C}$ such that $\textit{cost}(P, \hat{C}) \leq 2\alpha(3 + 2\alpha)(1 + \delta)\textit{cost}(P, C^*)$.*

The proof is very similar to that of Theorem IV.2, and can be found in Appendix B.

C. Straggler-Resilient Distributed (r,k)-Subspace Clustering

Note that the approximation factor of over 10 obtained using Algorithm 2 is quite prohibitive. We observe that Algorithm 2 succeeds because the weighted centers (Y_i, g_i) sent by the

local machines W_i are in fact a coreset of the local dataset P_i in a weak sense [2]. We leverage this observation to present a variant of Algorithm 2 that is computationally more efficient and also guarantees an improved approximation factor. In Algorithm 3, each machine sends a δ-coreset of its local dataset instead of an exact k-means solution. We now show that this small change can yield a better approximation factor and more general results.

In this section, we present a straggler resilient algorithm for a general class of squared ℓ_2 fitting problems, known as (r, k) subspace clustering problems where the goal is to find k subspaces each of dimension at most r that best fit the data. Note that the subspace clustering problem covers both the k-means and the principal component analysis (PCA) problems as special cases.

Algorithm 3 Straggler-resilient distributed (r, k)-subspace clustering

1: **Initialize:** A collection of n data points $P \subset \mathbb{R}^d$
2: Allocate P to m machines according to a (t, δ)-straggler resilient matrix A.
3: Assign the set of points $P_i \subset P$ to machine W_i
4: Each machine W_i computes δ-coreset (Y_i, g_i) of P_i.
5: FC Collects $\{(Y_i, g_i)\}_{i\in\mathcal{R}}$ from the non-straggling machines
6: Let $Y = \bigcup_{i\in\mathcal{R}} Y_i$. Using the recovery vector $\mathbf{b}$, define $g : Y \to \mathbb{R}$ such that $g(\mathbf{y}) = b_i g_i(\mathbf{y}), \forall \mathbf{y} \in Y_i$ and $i \in \mathcal{R}$
7: **Return** $\hat{C}$, the set of r-subspaces that is an α-approximate solution to the (r, k)-subspace clustering on input (Y, g).

In the following lemma, we show that the cost incurred by the aggregated weighted dataset (Y, g) is close to the cost incurred by P for any set of k centers C. In other words, (Y, g) is a coreset of P.

Lemma IV.4. *Let $\delta \in (0, 1)$. For any set of k-centers $C \subset \mathbb{R}^d$, we have*

$$(1-\delta)\mathit{cost}(P, C) \le \mathit{cost}(Y, g, C) \le (1+3\delta)\mathit{cost}(P, C).$$

Proof. The proof is relegated to Appendix C. □

The following result quantifies the quality of the clustering solution $\hat{C}$ provided by Algorithm 3 on the entire dataset P.

Theorem IV.5. *Let $\delta \in (0, 1)$. Let C^* be the optimal solution for (r, k)-subspace clustering on dataset P. Then, Algorithm 3 on dataset P returns a set of k subspaces $\hat{C}$ such that $\mathit{cost}(P, \hat{C}) \le \alpha(1 + 4\delta)\mathit{cost}(P, C^*)$.*

Proof. From the bounds in Lemma IV.4, we have

$$\begin{aligned}\text{cost}(P, \hat{C}) &\overset{(a)}{\le} \frac{\text{cost}(Y, g, \hat{C})}{1-\delta} \overset{(b)}{\le} \frac{\alpha}{1-\delta}\text{cost}(Y, g, C^*) \\ &\overset{(c)}{\le} \frac{\alpha(1+3\delta)}{1-\delta}\text{cost}(P, C^*) \le \alpha(1+4\delta)\text{OPT}, \quad (6)\end{aligned}$$

where (a) and (c) follow from Lemma IV.4, and (b) follows from the fact that FC computes an α-approximate k-means clustering on (Y, g). □

Coreset constructions for various clustering algorithms with squared ℓ_2 cost were considered in [37], [38]. There is a long line of work that has focused on constructing coresets for subspace clustering and for the k-means problems [34], [37], [39], [40]. Prior to the work of [41], the size of the coresets was dependent on the dimension of the problem d. However, in [34], first coresets of dimension independent sizes were provided. In particular, [34] constructed ϵ-coresets of size $O(k/\epsilon)$ and $\tilde{O}(k^3/\epsilon^4)$ for subspace and k-means clustering, respectively. Later, [42], [43] improved the coreset sizes to poly(k/ϵ) for the subspace clustering problem and was further reduced to $\tilde{O}(k/\epsilon^4)$ for k-means and k-median problems by [44]. The current state-of-the-art coreset sizes are $\tilde{O}(k\epsilon^{-2} \cdot \min\{\epsilon^{-z}, k\})$ where $z = 2$ for k-means and $z = 1$ for k-median problems as given in [45].

Therefore, Algorithm 3 obtains an approximation factor of $(1+4\delta)$ when each machine communicates $\tilde{O}(k/\epsilon^4)$ points to the FC. Whereas, in Algorithm 2 each machine communicates k points to obtain an approximation factor of $10(1+\delta)$. The observation indicates that with an increase in communication, we can obtain better accuracy.

V. Byzantine Resilient Clustering

A. Byzantine Resilient Distributed k-Median Clustering

In this section, we design distributed clustering methods that are robust to the presence of Byzantines. Since the Byzantines can send arbitrary information, naive clustering algorithms may lead to solutions that may be of poor quality (illustrated in Section VIII).

We first present a simple solution that assumes sufficient storage and computational power of the FC. Note that such an assumption is not unrealistic as central servers are usually quite powerful. However, the proposed algorithm is quite computationally and storage intensive which makes it prohibitive for practical applications with limited resources. We later propose techniques to address this difficulty by incurring slightly larger approximation factors.

The dataset P is distributed among the m machines using the assignment matrix A which satisfies Property III.2. The basic idea of the proposed algorithm is that each honest machine sends a set of k-median centers of their respective data subsets. Next, the FC combines the set of k-median centers from all the machines and computes the respective cost of clustering on them to gauge the quality of the centers sent by each machine. The FC then computes a good-quality[3] clustering solution for the entire dataset by filtering out the summaries with larger cost. We present the aforementioned steps in detail in Algorithm 4.

Remark 4 (Time Complexity). The workers and the FC use the $O(1)$-approximate k-medians clustering algorithm of [36] that runs in time $O(|P_i|d)$ at the worker nodes, and $O(mkd)$

[2] the cost of clustering using the weighted set (Y, g) is close to the cost of clustering using the entire dataset P albeit with an offset (Lemma A.3)

[3] good approximation factor

Algorithm 4 Byzantine-resilient distributed k-median clustering

1: **Initialize:** A collection of n vectors $P \subset \mathbb{R}^d$
2: Allocate P to m machines according to a (t, δ)-Byzantine resilient matrix A.
3: Assign the set of points $P_i \subset P$ to machine W_i
4: Each honest worker W_i computes an α-approximate k-median solution Y_i on set P_i
5: Each honest worker W_i sends the set of points Y_i to FC
6: Byzantine workers send an arbitrary set of k points.
7: FC computes & arranges received point sets in non-decreasing order of $\text{cost}(P_i, Y_i)$.
8: Without loss of generality, assume $\text{cost}(P_1, Y_1) \leq \text{cost}(P_2, Y_2) \leq \ldots \leq \text{cost}(P_m, Y_m)$.
9: For each point $\mathbf{y} \in Y_i$, FC computes weight $g_i(\mathbf{y}) = |\text{cluster}(\mathbf{y}, P_i)|$.
10: Let $Y = \bigcup_{i\in[m-t]} Y_i$. Using ρ, define $g : Y \to \mathbb{R}$ such that $g(\mathbf{y}) = \rho g_i(\mathbf{y}), \forall \mathbf{y} \in Y_i$
11: **Return** $\hat{C}$, the α-approximate k-median solution on (Y, g).

time at the FC. The FC takes additional $O(mdk \log k)$ time to perform an the filtering in Step 7-9 .

We present the following intermediate result, which shows that the cost incurred by the weighted summary Y_i of machine W_i on any set of k centers C is bounded by the cost of clustering the local dataset P_i with C, and the *quality* of the summary Y_i. Note that these bounds hold irrespective of the machine W_i being honest or Byzantine and rely on the fact that the FC can correctly compute the weights $g_i(\mathbf{y})$.

Lemma V.1. *For any $i \in [m]$, the weighted point set (Y_i, g_i) satisfies*

$$\begin{aligned} \textit{cost}(P_i, C) - \textit{cost}(P_i, Y_i) &\leq \textit{cost}(Y_i, g_i, C) \\ &\leq \textit{cost}(P_i, C) + \textit{cost}(P_i, Y_i). \quad (7) \end{aligned}$$

Proof. The proof is relegated to Appendix D. □

Lemma V.1 shows that the cost of clustering the weighted data subset (Y_i, g_i) (where summary Y_i is obtained from W_i and weight function g_i is computed at the FC), with any set of k centers C, $\text{cost}(Y_i, g_i, C)$ deviates from $\text{cost}(P_i, C)$ by an additive term of $\text{cost}(P_i, Y_i)$. The latter term quantifies the quality of the summary Y_i obtained from W_i. We assume that this quantity can be computed (or approximated) by the FC. This information is then used to filter out the summaries that contribute to large cost of clustering. From these observations, we get our main result that evaluates the quality of the clustering solution, $\hat{C}$, obtained by Algorithm 4 on the entire dataset P.

Theorem V.2. *Let C^* be the optimal solution to the k-median problem on point set P. Then, Algorithm 4 on dataset P returns a set of k-centers $\hat{C}$ such that $\textit{cost}(P, \hat{C}) \leq 3\alpha^2(1 + \delta)\textit{cost}(P, C^*)$, even in the presence of t Byzantines.*

Proof of Theorem V.2. Let $\hat{C}$ be the set of k-centers returned by Algorithm 4. From Lemma III.2, we have

$$\text{cost}(P, \hat{C}) \leq \sum_{i=1}^{m-t} \rho \ \text{cost}(P_i, \hat{C}),$$

utilizing the result from Lemma V.1 with $C = \hat{C}$, we get

$$\begin{aligned} \text{cost}(P, \hat{C}) &\leq \sum_{i=1}^{m-t} \rho \ \text{cost}(P_i, \hat{C}) \\ &\leq \sum_{i=1}^{m-t} \rho \ \text{cost}(P_i, Y_i) + \sum_{i=1}^{m-t} \rho \ \text{cost}(Y_i, g_i, \hat{C}). \end{aligned}$$

Next, we note that for every Byzantine in $j \in [m-t]$, there is an honest machine $i \in \mathcal{R}$ with a higher cost, i.e. $\text{cost}(P_i, Y_i) \geq \text{cost}(P_j, Y_j)$, which yields the following.

$$\text{cost}(P, \hat{C}) \leq \sum_{i\in\mathcal{R}} \rho \ \text{cost}(P_i, Y_i) + \sum_{i=1}^{m-t} \rho \ \text{cost}(Y_i, g_i, \hat{C}).$$

Since Y_i is an α-approximate k-median solution on the partial dataset P_i, we have $\text{cost}(P_i, Y_i) \leq \alpha \ \text{cost}(P_i, C^*)$. Hence, we have

$$\text{cost}(P, \hat{C}) \leq \alpha \sum_{i\in\mathcal{R}} \rho \ \text{cost}(P_i, C^*) + \sum_{i=1}^{m-t} \rho \ \text{cost}(Y_i, g_i, \hat{C}).$$

We apply the result from Lemma III.2 to the first term. Utilizing the definition of the cost function on a weighted point set, $\text{cost}(Y, g, \hat{C})$ and the α approximate solution $\hat{C}$ of the weighted dataset (Y, g) in the second term, we obtain

$$\text{cost}(P, \hat{C}) \leq \alpha(1 + \delta)\text{cost}(P, C^*) + \alpha \ \text{cost}(Y, g, C^*).$$

From the definition of the cost function, $\text{cost}(Y, g, C^*)$, we get

$$\begin{aligned} \text{cost}(P, \hat{C}) &\leq \alpha(1 + \delta)\text{cost}(P, C^*) + \alpha \sum_{i=1}^{m-t} \text{cost}(Y_i, \rho g_i, C^*) \\ &\leq \alpha(1 + \delta)\text{cost}(P, C^*) + \alpha \sum_{i=1}^{m-t} \rho \ \text{cost}(Y_i, g_i, C^*). \end{aligned}$$

Next, applying the result from Lemma V.1 to the second term above, we have

$$\begin{aligned} \text{cost}(P, \hat{C}) \leq \alpha(1 + \delta)\text{cost}(P, C^*) &+ \alpha \sum_{i=1}^{m-t} \rho \ \text{cost}(P_i, Y_i) \\ &+ \alpha \sum_{i=1}^{m-t} \rho \ \text{cost}(P_i, C^*). \end{aligned}$$

For the second term above, using a similar manipulation as before, we obtain

$$\begin{aligned} \text{cost}(P, \hat{C}) \leq \alpha(1 + \delta)\text{cost}(P, C^*) &+ \alpha^2 \sum_{i\in\mathcal{R}} \rho \ \text{cost}(P_i, C^*) \\ &+ \alpha \sum_{i=1}^{m-t} \rho \ \text{cost}(P_i, C^*), \end{aligned}$$

applying Lemma III.2 to the second and third terms, we obtain

$$\begin{aligned}\text{cost}(P,\hat{C}) &\leq \alpha(1+\delta)\text{cost}(P,C^*) + \alpha^2(1+\delta)\text{cost}(P,C^*) \\ &\quad + \alpha(1+\delta)\text{cost}(P,C^*) \\ &\leq 3\alpha^2(1+\delta)\text{cost}(P,C^*).\end{aligned}$$

□

B. Improved Byzantine Resilient Distributed k-Median Clustering

Recall that in the previous section, we assumed that the FC can compute the local summaries to evaluate the quality of the data sent by each local machine. In particular, we required the FC to have access to the entire dataset P. The FC needs them to estimate the cost of computing cluster P_i using Y_i sent by the machine W_i (Step 8 in Algorithm 4). This assumption is generally reasonable since in most applications, the FC is quite powerful and has access to the entire dataset. However, in resource-constrained settings such assumptions increase the computational load at the FC can be rather restrictive.

In this section, we discuss a simple technique to relax this assumption. For any $\delta \in (0,1)$, let $(\tilde{P}_i, w_i)$ denote a δ-coreset computed by the FC of dataset $P_i, i \in [m]$. One possible approach to compute this efficiently in a streaming fashion is by using the uniform sampling where a δ-coreset of a set of n points is computed by only storing poly$(k\epsilon^{-1})$ points as given in [46]. Another possible approach is to use sensitivity sampling techniques of [31], [47]. Specifically, the algorithm of [31] only stores a small set of points, and computes a good coreset using only the stored points. They show that to compute a δ-coreset of a set of n points, it is sufficient to only store $O(\delta^{-2}dk\log k)$ points. Therefore, the FC will only need to store $O(mdk\log k)$ points in total. Using standard dimension reduction techniques, we can further without loss of generality assume that $d = O(\log n)$.

To improve Algorithm 4, the coreset $(\tilde{P}_i, w_i)$ computed on each dataset P_i (using sensitivity sampling [31]) is utilized to approximate the cost of clustering pointset P_i with $Y_i, i \in [m]$. Furthermore, the weights $g_i(\mathbf{y})$ for each $\mathbf{y} \in Y_i$ are also estimated using only the coreset points. This reduces the computational load at the FC. In particular, estimating cost(P_i, Y_i), takes only $O(k^2 \log |P_i|)$ time instead of $O(k|P_i|)$. In the following, we show that we still obtain a good approximation for k-median clustering in the presence of Byzantines using $(\tilde{P}_i, w_i)$ instead of P_i. Furthermore, this coreset computation at the FC can be done while assigning them to the machines. The modified algorithm for distributed k-median clustering in the presence of Byzantines is presented in Algorithm 5. For the simplicity of presentation, we assume that machines and the FC can compute the exact (i.e., $\alpha = 1$) k-median solution on a small dataset. The results extend trivially when in Step 5 and Step 12, the machines and the FC compute an α-approximate solution.

Theorem V.3. *Let $\delta \in (0,1)$. Let C^* be the optimal solution to the k-median problem on point set P. Then, Algorithm 5 returns a set of k-centers $\hat{C}$ such that cost$(P,\hat{C}) \leq \left(\frac{2}{1-1/k} + \frac{1}{1-\delta}\right)(1+3\delta)$cost$(P,C^*)$, even in the presence of t Byzantines with probability $1 - \frac{1}{k}$.*

Algorithm 5 Computationally-efficient Byzantine-resilient distributed k-median clustering

1: **Initialize:** A collection of n vectors $P \subset \mathbb{R}^d$
2: Allocate P to m machines according to A with Property III.2.
3: FC computes δ-coreset $(\tilde{P}_i, w_i)$ from the streaming data with respect to each P_i
4: Assign the set of points $P_i \subset P$ to machine W_i
5: Each honest worker W_i computes k-median solution Y_i on set P_i
6: Each honest worker W_i sends the set of points Y_i to FC
7: Byzantine workers send an arbitrary set of k points.
8: FC computes & arranges received point sets in non-decreasing order of cost$(\tilde{P}_i, w_i, Y_i)$.
9: Without loss of generality, assume cost$(\tilde{P}_1, w_1, Y_1) \leq$ cost$(\tilde{P}_2, w_2, Y_2) \leq \ldots \leq$ cost$(\tilde{P}_m, w_m, Y_m)$.
10: For each point $\mathbf{y} \in Y_i$, FC computes weight $\tilde{g}_i(\mathbf{y}) = \sum_{p \in \text{cluster}(\mathbf{y}, \tilde{P}_i)} w_i(p)$.
11: Let $Y = \bigcup_{i \in [m-t]} Y_i$. Using ρ, define $\tilde{g} : Y \to \mathbb{R}$ such that $\tilde{g}(\mathbf{y}) = \rho\tilde{g}_i(\mathbf{y}), \forall \mathbf{y} \in Y_i$
12: **Return** $\hat{C}$, the k-median solution on $(Y, \tilde{g})$.

We now briefly sketch the proof of Theorem V.3. The formal proof is presented in Appendix E.

The two main differences in Algorithm 5 compared to Algorithm 4 are

1) The filtering of Byzantines in Step 8 is done with respect to cost$(\tilde{P}_i, w_i, Y_i)$ instead of cost(P_i, Y_i). Since $(\tilde{P}_i, w_i)$ is a δ-coreset of P_i, we incur at most a factor of $(1+\delta)$ in cost by making this change.
2) For any $i \in [m-t]$, and $\mathbf{y} \in Y_i$, the quantity $g_i(\mathbf{y}) = |\text{cluster}(\mathbf{y}, P_i)|$ is computed using the coreset $\tilde{P}_i$ instead of the actual pointset P_i. We show that by adopting a particular sensitivity-based i.i.d. sampling technique of coreset construction, the estimate of $\tilde{g}_i$ is at most some $(1+\gamma)$ factor away from its intended value with very high probability, for some appropriately chosen value of γ.

We now formalize the above two statements in the following Lemmas and Observations.

Observation V.1. Let $\delta \in (0,1)$. For any $i \in [m]$ and any set of k centers C, we have

$$|\text{cost}(\tilde{P}_i, w_i, C) - \text{cost}(P_i, C)| \leq \delta\text{cost}(P_i, C)$$

The observation follows from the fact that $(\tilde{P}_i, w_i)$ is a δ-coreset of P_i

Lemma V.4. *Let $\gamma \geq \frac{1}{k}$. For any $i \in [m]$, and $\mathbf{y} \in Y_i$,*

$$\Pr[|\tilde{g}_i(\mathbf{y}) - g_i(\mathbf{y})| \geq \gamma\ g_i(\mathbf{y})] \leq \frac{1}{k}$$

Lemma V.4 therefore ensures the following:

Observation V.2. Let $\gamma \geq \frac{1}{k}$. For any $i \in [m]$ and $\mathbf{y} \in Y_i$, let $g_i(\mathbf{y}) := |\text{cluster}(\mathbf{y}, P_i)|$. Then for any set of k centers C,

$$\begin{aligned}\text{cost}(Y_i, g_i, C) &= \sum_{\mathbf{y}\in Y_i} g_i(\mathbf{y}) d(\mathbf{y}, C) \leq \sum_{\mathbf{y}\in Y_i} \frac{1}{1-\gamma}\tilde{g}_i(\mathbf{y}) d(\mathbf{y}, C) \\ &= \frac{1}{1-\gamma}\text{cost}(Y_i, \tilde{g}_i, C),\end{aligned}$$

with probability at least $1 - 1/k$.

Using the two observations listed above, we get an equivalent of Lemma V.1.

Lemma V.5. *Let $\delta, \gamma \in (0, 1)$. For any $i \in [m]$, the weighted point set $(Y_i, \tilde{g}_i)$ satisfies*

$$\begin{aligned}&(1-\gamma)\textit{cost}(P_i, C) - \frac{1-\gamma}{1-\delta}\textit{cost}(\tilde{P}_i, w_i, Y_i) \leq \textit{cost}(Y_i, \tilde{g}_i, C) \\ &\leq (1+\delta)\textit{cost}(P_i, C) + \textit{cost}(\tilde{P}_i, w_i, Y_i).\end{aligned}$$

The proof of Theorem V.3 then follows similar to the proof of Theorem V.2 using the adjusted Lemma V.5 instead of Lemma V.1.

C. Byzantine Resilient k-means Clustering

Similar to Algorithm 2 for straggler resilient k-means clustering, a simple modification can be made to Algorithm 4 to obtain a Byzantine-resilient distributed k-means algorithm (Algorithm 6) with performance guarantees given in Theorem V.6.

Algorithm 6 Byzantine-resilient distributed k-means clustering

1: **Initialize:** A collection of n vectors $P \subset \mathbb{R}^d$
2: Allocate P to m machines according to A with Property III.2.
3: Assign the set of points $P_i \subset P$ to machine W_i
4: Each honest worker W_i computes k-means solution Y_i on set P_i
5: Each honest worker W_i sends the set of points Y_i to FC
6: Byzantine workers send an arbitrary set of k unweighted points.
7: FC computes & arranges received point sets in non-decreasing order of $\text{cost}(P_i, Y_i)$.
8: Without loss of generality, assume $\text{cost}(P_1, Y_1) \leq \text{cost}(P_2, Y_2) \leq \ldots \leq \text{cost}(P_m, Y_m)$.
9: For each point $\mathbf{y} \in Y_i$, FC computes weight $g_i(\mathbf{y}) = |\text{cluster}(\mathbf{y}, P_i)|$.
10: Let $Y = \bigcup_{i\in[m-t]} Y_i$. Using ρ, define $g : Y \to \mathbb{R}$ such that $g(\mathbf{y}) = \rho g_i(\mathbf{y}), \forall \mathbf{y} \in Y_i$
11: **Return** $\hat{C}$, the k-means solution on (Y, g).

Theorem V.6. *Let C^* be the optimal solution to the k-means problem on point set P. Then, Algorithm 6 returns a set of k-centers $\hat{C}$ such that $\textit{cost}(P, \hat{C}) \leq 10\alpha^2(1+\delta)\textit{cost}(P, C^*)$, even in the presence of t Byzantines.*

Moreover, similar to Algorithm 5, the FC can reduce its computational and storage costs by computing δ-coresets for k-means clustering of each local data set. Coresets obtained by i.i.d. sensitivity sampling of the data in a streaming fashion require $O(\delta^{-2}mdk\log k)$ points to be stored in total.

Algorithm 7 Computationally-efficient Byzantine-resilient distributed k-means clustering

1: **Initialize:** A collection of n vectors $P \subset \mathbb{R}^d$
2: Allocate P to m machines according to A with Property III.2
3: FC computes weighted coreset $(\tilde{P}_i, w_i)$ from the streaming data with respect to each P_i
4: Assign the set of points $P_i \subset P$ to machine W_i
5: Each honest worker W_i computes k-means solution Y_i on set P_i
6: Each honest worker W_i sends the set of points Y_i to FC
7: Byzantine workers send an arbitrary set of k points.
8: FC computes & arranges received point sets in non-decreasing order of $\text{cost}(\tilde{P}_i, w_i, Y_i)$.
9: Without loss of generality, assume $\text{cost}(\tilde{P}_1, w_1, Y_1) \leq \text{cost}(\tilde{P}_2, w_2, Y_2) \leq \ldots \leq \text{cost}(\tilde{P}_m, w_m, Y_m)$.
10: For each point $\mathbf{y} \in Y_i$, FC computes weight $\tilde{g}_i(\mathbf{y}) = \sum_{\mathbf{p}\in\text{cluster}(\mathbf{y},\tilde{P}_i)} w_i(\mathbf{p})$.
11: Let $Y = \bigcup_{i\in[m-t]} Y_i$. Using ρ, define $\tilde{g} : Y \to \mathbb{R}$ such that $\tilde{g}(\mathbf{y}) = \rho\tilde{g}_i(\mathbf{y}), \forall \mathbf{y} \in Y_i$
12: **Return** $\hat{C}$, the k-means solution on $(Y, \tilde{g})$.

Theorem V.7. *Let $\delta \in (0, 1)$. Let C^* be the optimal solution to the k-means problem on point set P. Then, Algorithm 7 returns a set of k-centers $\hat{C}$ such that $\textit{cost}(P, \hat{C}) \leq \left(\frac{8}{1-1/k} + \frac{2}{1-\delta}\right)(1+3\delta)\textit{cost}(P, C^*)$, even in the presence of t Byzantines with probability $1 - \frac{1}{k}$.*

The proofs of both Theorem V.6, and Theorem V.7 are analogous to the k-median proof. In fact, they are verbatim the same except for the use of scaled triangular inequality (Eq. (5)) instead of standard triangle inequality used in the proofs for their k-median counterparts.

VI. Construction of Data Assignment Matrix

In this section, we provide the approach for the construction of the assignment matrix in the presence of stragglers and Byzantines. Since Property III.2 for Byzantine resilience is stronger than Property III.1, we will focus only on the construction of assignment matrices A that satisfy the former. The straggler resilience property of those matrices will follow from the definition.

Let n be the number of data points in P, and m be the number of machines. Let $\mathcal{B} \subset [m]$, $|\mathcal{B}| < t$ denote the set of Byzantines, and let $\mathcal{R} = [m] \setminus \mathcal{B}$ be the set of non-Byzantines. For the simplicity of presentation, we assume $n = m$. This assumption holds without loss of generality as we can arbitrarily partition the dataset into m chunks of size n/m each, and distribute each chunk as a single data point.

We now present the construction of various assignment matrices $A \in \{0, 1\}^{m\times m}$ that satisfy Property III.2, and hence Property III.1 as well. The two parameters of importance when constructing an assignment matrix are the load per machine

which is defined as the number of data points sent to each machine ($\ell = \max_i |P_i|$), and the fraction of faulty machines that can be tolerated ($= t/m$). For each of the constructions provided below, we analyze the tradeoffs between these two parameters.

A. Randomized Construction

In this section, we show that a random Bernoulli assignment matrix satisfies Property III.2 albeit with slightly degraded tradeoffs between ℓ and t.

Consider an $m \times m$ random Bernoulli assignment matrix A where each entry $A_{i,j}$ is set to 1 independently with some probability p, and 0 otherwise.

Theorem VI.1. *For any $\delta > 0$, the Bernoulli assignment matrix A with $p = O(\frac{1}{\log m})$, satisfies Property III.2 with probability at least $1-O(\frac{1}{m})$, and is resilient to $t = O(\frac{m}{\log^2 m})$ Byzantines.*

Proof. Proof is relegated to Appendix F. □

Alternatively, Theorem VI.1 satisfies (t, δ)-Byzantine resilience property with an expected load of $\ell = O(\frac{mt(2+\delta)^2}{\delta^2(m-t)} \log m)$. Note that Theorem VI.1 provides lesser redundancy in the regime when $t = o(m)$ compared to the naïve solution of distributing all the points to all the machines (for which $\ell = m$).

B. Explicit Construction

We now present an explicit construction of an assignment matrix that satisfies Property III.2. The construction is based on expander graphs which were also used to construct explicit data assignment schemes for gradient coding [23], [24].

Let $G = (V, E)$ be a connected d-regular graph on m vertices and let $\mathcal{A}_G$ denote its adjacency matrix. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$ be the m real eigenvalues of $\mathcal{A}_G$. Define the expansion parameter of graph G as $\lambda = \max\{|\lambda_2|, |\lambda_m|\}$. We denote such d-regular graphs on n vertices with expansion parameter λ as (n, d, λ)-expanders.

The double cover of a graph $\tilde{G} = (\tilde{V}, \tilde{E})$ on n vertices, is a bipartite graph $G = (L \cup R, E)$, on $2n$ vertices with $L = R = V$. There is an edge $(u, v) \in L \times R$ in G if and only if $(u, v) \in \tilde{E}$.

To construct our assignment matrix, we consider a bipartite graph $G = (L \cup R, E)$ that is a double cover of an $(m/2, d, \lambda)$-expander. The $m \times m$ assignment matrix A is obtained from G by setting $A_{u,v} = 1$ if and only if there is an edge between $(u, v) \in G$ for any $u \in R$ and, $v \in L$. We now show that the assignment matrix A obtained from G satisfies Property III.2 for any set of t Byzantines.

Theorem VI.2. *For any $\delta > 0$, the assignment matrix A satisfies Property III.2 with $t = \sqrt{\log m / \log\log m}$, and $\ell = O(\log m)$.*

The proof, presented formally in Appendix G, follows from the fact that if $\tilde{G}$ is an expander graph, then its double cover G satisfies the expander Mixing Lemma [48].

Theorem VI.3 (Expander Mixing Lemma [48])**.** *For any sets S and T in a (n, d, λ)-expander, we have $|E(S,T) - \frac{d}{n}|S||T|| \leq \lambda\sqrt{|S||T|}$, where, $E(S,T)$ denotes the number of edges between sets S and T.*

Using Expander Mixing Lemma, we can show that no vertex in L is incident to a large fraction of vertices in any t subset of R. This in turn translates to the fact that no column of A has a large number of 1's in any subset of t rows of A. Therefore, removing any t rows of A keeps all the column weights within a fixed range.

The existence of graphs with appropriate expansion properties then completes the proof. We use the constructions of (n, d, λ)-expanders of [49], to get data assignment schemes that are resilient to $O(\sqrt{\log m})$ Byzantines with an overhead of $O(\log m)$ data points per machine.

Theorem VI.4 ([49])**.** *There exists a polynomial time algorithm to construct $(n, d, \lambda) = (2^\ell, \ell - 1, \sqrt{\ell \log^3 \ell})$.*

In particular, the matrix obtained using a double cover of an expander graphs of [49] gives a (t, δ)-Byzantine resilient assignment matrix with $\delta = \gamma/(1-\gamma)$, where $\gamma = \frac{t}{m} + \sqrt{\frac{t \log^3 \log m}{\log m}}$, and $\ell = O(\log m)$.

VII. Random Straggler Model

In this section, we consider the random fault model for stragglers. In this model, we assume that each machine W_i, for $i \in [m]$ behaves as a straggler independently with some fixed (known) probability p_t.

Property VII.1 ((t, δ)- Random straggler resilience property)**.** Let $\delta > 0$ be a given constant. The assignment matrix $A \in \{0, 1\}^{m \times n}$ has (t, δ)- random straggler resilience if for a random subset of $m - t$ rows $\mathcal{R} \subseteq [m]$ chosen i.i.d with probability $1 - p_t$, $\exists$ a recovery vector, $\mathbf{b} = (b_1, \ldots, b_{|\mathcal{R}|})^T \in \mathbb{R}^{|\mathcal{R}|}, b_i > 0, \forall i \in |\mathcal{R}|$, such that for all $i \in [n]$,

$$\mathbf{1}_n^T \leq \sum_{i \in \mathcal{R}} b_i \mathbf{a}_i \leq (1 + \delta)\mathbf{1}_n^T, \tag{8}$$

with probability at least $1 - 1/m$.

We note that the proofs of Lemma III.1, and hence Lemma IV.1 hold with high probability for a random set $\mathcal{R}$ given an assignment matrix A that satisfies Property VII.1. Therefore, the guarantees of Theorems IV.2, Theorem IV.3, and Theorem IV.5 continue to hold with high probability.

Constructions of matrices satisfying Property VII.1 are presented in Section VII-A and Section VII-B. We note that these constructions provide better trade-off between the load per machine to tolerate a constant fraction of random stragglers.

We also remark that the equivalent of Property III.2 for random Byzantines is not sufficient to get provable guarantees for the clustering algorithms described in Section V as we strictly require Lemma III.2 to hold for any arbitrary set of $m - t$ indices for the proof of Theorem V.2. Therefore, the random Byzantine model does not give us any advantage over the adversarial constructions considered in Section VI.

A. Randomized Construction for Random Stragglers

We present a randomized construction of the assignment matrix that satisfies Property III.1. For the construction of the matrix, we assume a random straggler model, where every machine acts as a straggler independently with probability p_t. Hence, the local computation from each machine is received at the FC with probability $1 - p_t$.

For some ℓ (to be chosen later), the (i, j)-th entry of the assignment matrix, based on the random construction discussed above, is defined as

$$A_{i,j} = \begin{cases} 1 & \text{with probability } p_a = \frac{\ell}{m} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

For an appropriate choice of ℓ (and, hence, p_a), we show that the random matrix A satisfies Property III.1 with high probability.

Theorem VII.1. *For any $\delta > 0$, the randomized assignment matrix in (9) with $\ell = \frac{6(2+\delta)^2}{\delta^2} \cdot \frac{\log(\sqrt{2}m)}{1-p_t}$ satisfies Property III.1 with probability at least $1 - \frac{1}{m}$ under the random straggler model.*

Proof. Proof is relegated to Appendix H. □

Therefore, the random Bernoulli construction gives a (t, δ)-random straggler resilient matrix with $\mathbb{E}[t] = mp_t = O(m)$ for a constant straggler probability for which $\mathbb{E}[\ell] = O(\frac{(2+\delta)^2}{\delta^2} \log(\sqrt{2}m))$.

B. Explicit Construction for Random Stragglers

Fractional Repetition Codes (FRC) have been well-studied in [50] for straggler resilient gradient computations. In this section, we show that the FRC scheme also satisfies Property III.1 for random stragglers with high probability, and hence provides redundant data assignment for straggler-resilient clustering problems.

For simplicity, let us assume that we have m data points and m machines. In FRC, the m data points are partitioned into groups of size s (assume that s divides m), and each group of data points is replicated across s machines. The assignment matrix A for this scheme is given by

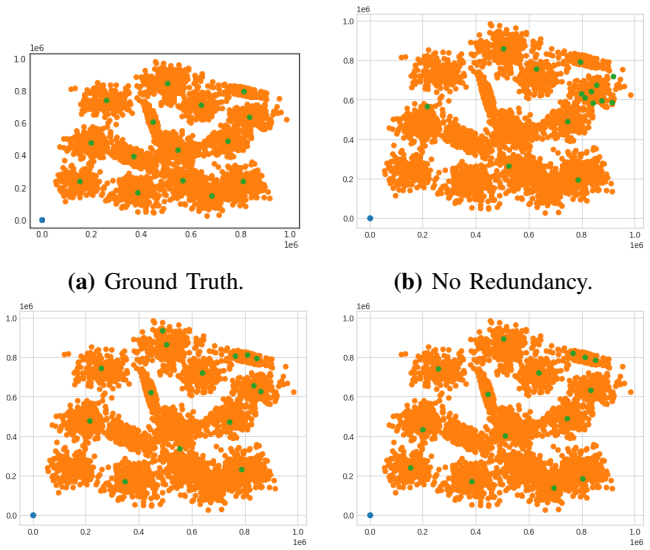
$$A = \begin{pmatrix} \mathbf{1_{s\times s}} & \mathbf{0_{s\times s}} & \mathbf{0_{s\times s}} & \dots & \mathbf{0_{s\times s}} \\ \mathbf{0_{s\times s}} & \mathbf{1_{s\times s}} & \mathbf{0_{s\times s}} & \dots & \mathbf{0_{s\times s}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0_{s\times s}} & \mathbf{0_{s\times s}} & \mathbf{0_{s\times s}} & \dots & \mathbf{1_{s\times s}} \end{pmatrix}, \tag{10}$$

where $\mathbf{1_{s\times s}}$ denotes an $s \times s$ matrix of all 1's.

Let $A_{\mathcal{R}}$ of size $|\mathcal{R}| \times m$ denote the submatrix of honest machines obtained by removing t rows from A uniformly at random. We now show that the random matrix $A_{\mathcal{R}}$ satisfies Property III.1 with high probability.

Theorem VII.2. *For any $\delta > 0$, the FRC based assignment matrix A with $\ell = s = O(\log m)$, satisfies Property III.1 with probability at least $1 - O(\frac{1}{m})$ under the random straggler model, and provides resilience against $t = O(m)$ stragglers.*

Proof. The proof is relegated to Appendix I. □



(a) Ground Truth. **(b)** No Redundancy.

(c) $p = 0.1$. **(d)** $p = 0.2$.

Fig. 2: Performance of the proposed Straggler-resilient k-median algorithm.

The FRS of Theorem VII.2 scheme gives a (t, δ)-random straggler resilient matrix with $\ell = O\left(\frac{(2+\delta)^2}{\delta^2} \frac{\log m}{(1-p_t)}\right)$, and $\mathbb{E}[t] = mp_t$. In particular, it provides good trade-off between the load per machine $\ell = O(\log m)$, and the number of Byzantines tolerated, $t = O(m)$.

Next, we empirically evaluate the performance of our algorithms and show that they are robust to Byzantines (or stragglers).

VIII. Simulation Results

In this section, we demonstrate the performance of our distributed k-median clustering algorithms that are resilient to stragglers and Byzantines, respectively. We consider the synthetic Gaussian dataset [51] with $n = 5000$ two-dimensional points that are distributed among $m = 10$ machines.

A. Straggler-resilient Clustering

In this section, we illustrate the performance of our straggler-resilient distributed k-median algorithm and benchmark it with the non-redundant data assignment scheme. We consider $t = 3$ randomly chosen stragglers. We present the results in Figures 2a, 2b, 2c, and 2d.

We plot the ground truth using the centroids provided in the dataset in Fig. 2a with k-median clustering, for $k = 15$. In Fig. 2b, we present the results by ignoring the local computations from the stragglers, i.e., Algorithm 1 is used without any redundant data assignment. We randomly partition the $n = 5000$ data points among $m = 10$ machines. The non-straggler machines send their respective k-median centers to the FC. Then, the FC runs a k-median algorithm on the $k(m - t)$ centers obtained from the non-straggler machines. From Fig. 2b, the set of poor quality k-centers obtained from this scheme is noticeable.

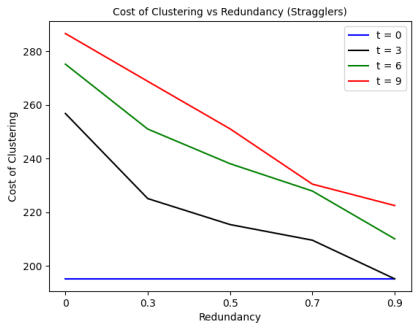


Fig. 3: Cost of clustering in presence of stragglers.

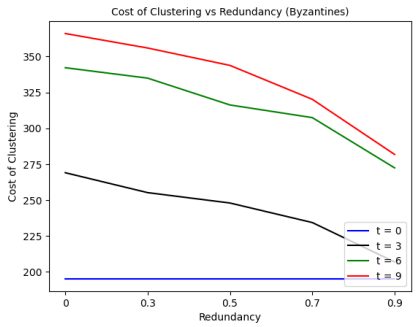
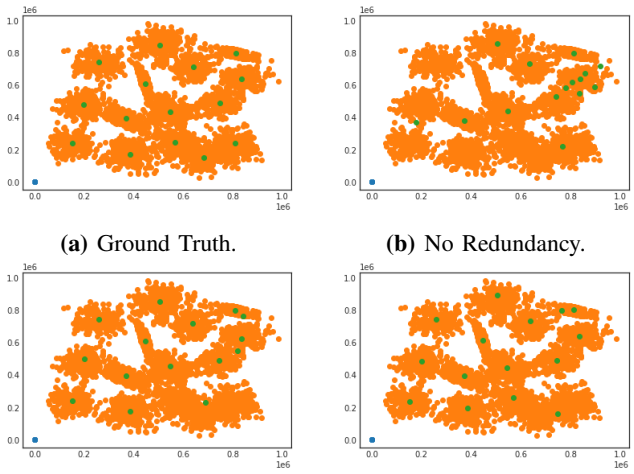


Fig. 5: Cost of clustering in presence of Byzantines.



(a) Ground Truth. **(b)** No Redundancy.

(c) $p = 0.1$. **(d)** $p = 0.2$.

Fig. 4: Performance of the proposed Byzantine-resilient k-median algorithm.

In Fig. 2c, the result obtained by using Algorithm 1 is shown. We choose the assignment matrix randomly with $p = \Pr[A_{i,j} = 1] = 0.1$. Hence, using this assignment matrix ensures that each machine receives 500 data points on an average which results in a non-redundant data assignment. Lastly, in Fig. 2d, we show the effect of increasing the value of p to 0.2. Therefore, the redundancy in the data assignment increases which results in each machine receiving about 1000 data points. We observe that the results are very close to the ground truth clustering presented in Fig. 2a.

In Fig. 3, the cost of clustering at the FC as a function of the redundancy of the data assignment scheme for different values of stragglers in the network is shown. We set the total number of machines as 10. For this experiment, we consider BIRCH [52] 2-dimensional dataset with 100,000 points and $k = 100$. The scheme with no stragglers ($t = 0$) acts as a benchmark for other schemes is shown in the plot as the 'blue' curve. We observe that the cost of clustering increases as the number of stragglers in the network is increased from 3 to 9 as expected. Further, we observe that the cost of clustering in the presence of stragglers decreases as the redundancy increases from 0 to 0.9.

B. Byzantine-resilient Clustering

In this section, we illustrate the performance of our Byzantine-resilient distributed k-median algorithm and benchmark it with the non-redundant data assignment scheme. We consider $t = 3$ randomly chosen Byzantines. We present the results in Figures 4a, 4b, 4c, and 4d.

We plot the ground truth using the centroids provided in the dataset in Fig. 4a with k-median clustering, for $k = 15$. In Fig. 4b, we present the results by ignoring the local computations from the Byzantines, i.e., Algorithm 4 is used without any redundant data assignment. We randomly partition the $n = 5000$ data points among $m = 10$ machines. The honest machines send their respective k-median centers to the FC. Then, the FC runs a k-median algorithm on the $(m - t)$ centers obtained from the honest machines. From Fig. 4b, the set of poor quality k-centers obtained from this scheme is noticeable.

In Fig. 4c, the result obtained by using Algorithm 4 is shown. We choose the assignment matrix randomly with $p = \Pr[A_{i,j} = 1] = 0.1$. Hence, using this assignment matrix ensures that each machine receives 500 data points on an average which results in a non-redundant data assignment. Lastly, in Fig. 4d, we show the effect of increasing the value of p to 0.2. Therefore, the redundancy in the data assignment increases which results in each machine receiving about 1000 data points. We observe that the results are very close to the ground truth clustering presented in Fig. 4a.

In Fig. 5, the cost of clustering at the FC as a function of the redundancy of the data assignment scheme for different values of Byzantines in the network is shown. Similar to the previous experiments, we set the total number of machines as 10. Moreover, we consider BIRCH [52] 2-dimensional dataset with 100,000 points and $k = 100$. The scheme with no Byzantines ($t = 0$) acts as a benchmark for other schemes is shown in the plot as the 'blue' curve. We observe that the cost of clustering increases as the number of Byzantines in the network is increased from 3 to 9 as expected. Further, we observe that the cost of clustering in the presence of Byzantines decreases as the redundancy increases from 0 to 0.9. Also, we observe that the cost of clustering in the presence of Byzantines is higher than the cost of clustering in the presence of stragglers.

IX. Conclusion

In this paper, we provided $O(1)$-approximate solutions for the distributed k-median and k-means clustering problems in the presence of stragglers. These algorithms were then extended to the case where Byzantines were present in the system. Note that the approach for k-means (Algorithm 2 and Algorithm 6) used in this work can be generalized to obtain straggler-resilient and Byzantine-resilient algorithms for a larger class of ℓ_2 fitting problems such as (r, k)-subspace clustering solutions. We also provided computationally efficient constant factor approximate solutions for the distributed clustering problems in the presence of Byzantines.

An alternate viable approach to tackle Byzantines is to use some outlier robust clustering at the FC to filter out Byzantines. At a high level, Algorithm 4 achieves that by filtering out all the points that incur large cost on the partial data sets. This ensures that the Byzantines cannot send arbitrary points. Finally, another interesting direction to explore would be to reduce communication costs between the machines and the FC resulting in communication-efficient clustering algorithms in the presence of stragglers and Byzantines.

Appendix

Outline of Appendix

We present all the missing proofs in the appendix. Appendix A – Appendix C deal with straggler resilient clustering algorithms

- In Appendix A, we present the proof of Lemma IV.1. Here, we present the general algorithm where each machine computes and sends to the FC an α-approximate k-medians solution instead of the exact solution.
- In Appendix B, we present the proof of Theorem IV.3 which contains all the required Lemmas and the analysis for k-means clustering in presence of stragglers.
- Appendix C contains the proof of Lemma IV.4. In this section, we present the proof for an important Lemma required for bounding the approximation factor for the (r, k)-subspace clustering.

Appendix D and Appendix E deal with Byzantine resilient clustering algorithms

- In Appendix D, we present the proof of Lemma V.1. In this section, we present the analysis for the Lemma required to obtain the cost of clustering using the k-median solution in the presence of Byzantines.
- In Appendix E, we present the proof of Theorem V.3. In this section, we present the analysis for required Lemmas and the cost of clustering in the presence of Byzantines when using the computationally efficient k-median algorithm.

In Appendix F, and Appendix G we present two constructions (one randomized, and another explicit construction) of (t, δ)-Byzantine (and straggler) resilient assignment matrices.

- In Appendix F, we present the proof of Theorem VI.1. In this section, we present the analysis for random Bernoulli assignment matrix for adversarial Byzantines.
- In Appendix G, we present the proof of Theorem VI.2. In this section, we present the analysis for expander graph based assignment matrix for adversarial Byzantines.

Finally, in Appendix H, and Appendix I we deal with the constructions of assignment matrices for the random straggler model.

- In Appendix H we present the proof of Theorem VII.1. In this section, we present the analysis for random Bernoulli assignment matrix for random Byzantines.
- In Appendix I, we present the proof of Theorem VII.2. In this section, we present the analysis for FRC based assignment matrix for random Byzantines.

Straggler Resilient Clustering

A. Straggler-Resilient Distributed k-median Clustering

In this section, we present the general algorithm mentioned in Remark 2.

Algorithm 8 Straggler-resilient distributed k-median

1: **Initialize:** A collection of n data points $P \subset \mathbb{R}^d$
2: Allocate P to m machines according to a (t, δ)-straggler resilient matrix A.
3: Let $P_i \subset P$ be the set of points assigned to machine W_i
4: Each machine W_i computes an α-approximate k-median solution, Y_i, on set P_i.
5: Define $g_i : Y_i \to \mathbb{R}$ as $g_i(\mathbf{y}) = |\text{cluster}(\mathbf{y}, P_i)|$, for every $y \in Y_i$
6: FC collects $\{(Y_i, g_i)\}_{i\in\mathcal{R}}$ from the non-straggling machines, for some $\mathcal{R} \subseteq [m], |\mathcal{R}| \geq m - t$
7: Let $Y = \bigcup_{i\in\mathcal{R}} Y_i$. Using the recovery vector $\mathbf{b}$, define $g : Y \to \mathbb{R}$ such that $g(\mathbf{y}) = b_i g_i(\mathbf{y}), \forall \mathbf{y} \in Y_i$ and $i \in \mathcal{R}$
8: **Return** $\hat{C}$, an α-approximate k-median solution on (Y, g).

Theorem A.1. *Let C^* be the optimal set of k-median centers for dataset P. Then, Algorithm 8 on dataset P returns a set of centers $\hat{C}$ such that cost$(P, \hat{C}) \leq \alpha(1+\delta)(2+\alpha)$cost$(P, C^*)$.*

Similar to the proof of Theorem IV.2, the proof of Theorem A.1 is established through Lemma A.2. The proof of Lemma IV.1 also follows from Lemma A.2 as a special case when $\alpha = 1$.

Lemma A.2. *For k-median clustering, for any set of k-centers $C \subset \mathbb{R}^d$, we have*

$$\text{cost}(P, C) - \sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, Y_i) \leq \text{cost}(Y, g, C) \leq (1+\alpha)(1+\delta)\text{cost}(P, C).$$

Proof of Lemma A.2. We prove each part of the inequality separately. First, we prove the upper bound on cost(Y, g, C) followed by the lower bound.

Upper Bound: We first show that for any set of k-centers $C \subset \mathbb{R}^d$, and for any $i \in [m]$, cost$(Y_i, g_i, C) \leq (1+\alpha)$cost$(P_i, C)$ which ensures that the weighted k-centers (Y_i, g_i) are a good representation of the partial dataset P_i. Consider the following

$$\begin{aligned}\text{cost}(Y_i, g_i, C) &= \sum_{\mathbf{y}\in Y_i} g_i(\mathbf{y}) d(\mathbf{y}, C) \\ &= \sum_{\mathbf{y}\in Y_i} |\text{cluster}(\mathbf{y}, P_i)| d(\mathbf{y}, C) \\ &\quad \text{(by definition of } g_i\text{)} \\ &= \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in \text{cluster}(\mathbf{y}, P_i)} d(\mathbf{y}, C). \end{aligned} \tag{11}$$

For any $\mathbf{x} \in \mathbb{R}^d$, recall that $C(\mathbf{x})$ denotes its closest center in C. From the above equality, we have

$$\begin{aligned}\text{cost}(Y_i, g_i, C) &= \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in \text{cluster}(\mathbf{y}, P_i)} d(\mathbf{y}, C(\mathbf{y})) \\ &\overset{(a)}{\leq} \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in \text{cluster}(\mathbf{y}, P_i)} d(\mathbf{y}, C(\mathbf{x})) \\ &\overset{(b)}{\leq} \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in \text{cluster}(\mathbf{y}, P_i)} (d(\mathbf{x}, \mathbf{y}) + d(\mathbf{x}, C(\mathbf{x}))) \\ &= \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in \text{cluster}(\mathbf{y}, P_i)} d(\mathbf{x}, \mathbf{y}) + \sum_{\mathbf{x}\in P_i} d(\mathbf{x}, C(\mathbf{x})) \\ &= \text{cost}(P_i, Y_i) + \text{cost}(P_i, C) \\ &\overset{(c)}{\leq} (1+\alpha)\text{cost}(P_i, C), \end{aligned} \tag{12}$$

where (a) follows from the definition of $C(\mathbf{x})$ and (b) follows from triangular inequality. (c) follows from the fact that the k-centers Y_i on the partial dataset P_i is an α-approximate solution, and therefore, $\text{cost}(P_i, Y_i) \leq \alpha\ \text{cost}(P_i, C)$. Next, we have

$$\begin{aligned}\text{cost}(Y, g, C) &= \sum_{i\in\mathcal{R}} \text{cost}(Y_i, b_i \cdot g_i, C) \\ &= \sum_{i\in\mathcal{R}} b_i \text{cost}(Y_i, g_i, C) \\ &\leq (1+\alpha) \sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, C) \\ &\leq (1+\alpha)(1+\delta)\text{cost}(P, C), \end{aligned} \tag{13}$$

where the first inequality follows from (12) and the second inequality follows from Lemma III.1.

Lower Bound: From Lemma III.1 for any set of k-centers C, we have

$$\begin{aligned}\text{cost}(P, C) &\leq \sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, C) \\ &= \sum_{i\in\mathcal{R}} b_i \sum_{\mathbf{x}\in P_i} d(\mathbf{x}, C(\mathbf{x})). \end{aligned} \tag{14}$$

From the definition of cluster centers, we know that for any two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and for any set of k-centers C, $d(\mathbf{x}, C(\mathbf{x})) \leq d(\mathbf{x}, C(\mathbf{y}))$. Applying this observation in (14), we get

$$\begin{aligned}\text{cost}(P, C) &\leq \sum_{i\in\mathcal{R}} b_i \sum_{\mathbf{x}\in P_i} d(\mathbf{x}, C(\mathbf{x})) \\ &\leq \sum_{i\in\mathcal{R}} b_i \sum_{\mathbf{x}\in P_i} d(\mathbf{x}, C(Y_i(\mathbf{x}))), \end{aligned} \tag{15}$$

where $Y_i(\mathbf{x})$ is the cluster center in Y_i closest to $\mathbf{x} \in P_i$. Using triangular inequality, we obtain

$$\begin{aligned}\text{cost}(P, C) &\leq \sum_{i\in\mathcal{R}} b_i \sum_{\mathbf{x}\in P_i} (d(\mathbf{x}, Y_i(\mathbf{x})) + d(Y_i(\mathbf{x}), C(Y_i(\mathbf{x})))) \\ &= \sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, Y_i) \\ &\quad + \sum_{i\in\mathcal{R}} b_i \sum_{\mathbf{x}\in P_i} d(Y_i(\mathbf{x}), C(Y_i(\mathbf{x}))) \\ &= \sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, Y_i) \\ &\quad + \sum_{i\in\mathcal{R}} b_i \sum_{\mathbf{x}\in Y_i} |\text{cluster}(\mathbf{y}, P_i)| d(\mathbf{y}, C(\mathbf{y})) \\ &= \sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, Y_i) + \sum_{i\in\mathcal{R}} b_i \text{cost}(Y_i, g_i, C) \\ &= \sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, Y_i) + \sum_{i\in\mathcal{R}} \text{cost}(Y_i, b_i \cdot g_i, C) \\ &= \sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, Y_i) + \text{cost}(Y, g, C). \end{aligned} \tag{16}$$

Combining the upper and the lower bounds, we obtain the final result. □

We now prove Theorem A.1.

Proof of Theorem A.1. Utilizing the lower bound from Lemma A.2 with $C = \hat{C}$, we have

$$\begin{aligned}\text{cost}(P, \hat{C}) &\leq \text{cost}(Y, g, \hat{C}) + \sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, Y_i) \\ &\overset{(a)}{\leq} \alpha\ \text{cost}(Y, g, C^*) + \alpha \sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, C^*) \\ &\overset{(b)}{\leq} \alpha(1+\alpha)(1+\delta)\text{cost}(P, C^*) \\ &\quad + \alpha(1+\delta)\text{cost}(P, C^*) \\ &= \alpha(1+\delta)(2+\alpha)\text{cost}(P, C^*), \end{aligned} \tag{17}$$

where (a) follows from the fact that $\hat{C}$ and Y_i are the α-approximate set of centers for the weighted dataset (Y, g) and the partial dataset P_i, respectively. For (b), we utilize the upper bound in Lemma A.2 and Lemma III.1 with $C = C^*$. □

B. Straggler-Resilient Distributed k-means

In this section, we prove the guarantees of Algorithm 2. First, we prove the following intermediate lemma that will establish bounds on the quality of the accumulated summaries from the machines.

Lemma A.3. *For the k-means clustering, for any set of k-centers $C \subset \mathbb{R}^d$, we have*

$$\begin{aligned}\frac{1}{2} cost(P, C) - \sum_{i\in\mathcal{R}} b_i cost(P_i, Y_i) &\leq cost(Y, g, C) \\ &\leq (2+2\alpha)(1+\delta) cost(P, C). \end{aligned}$$

Proof of Lemma A.3. We split the proof into two parts. The first part involves the upper bound and in the second part, we prove the lower bound.

Upper Bound: We first show that for any set of k-centers $C \subset \mathbb{R}^d$, for any $i \in [m]$, $\text{cost}(Y_i, g_i, C) \leq (2+2\alpha)\text{cost}(P_i, C)$ which ensures that the weighted k-centers (Y_i, g_i) are a good representation of the partial dataset P_i. Consider the following:

$$\begin{aligned}\text{cost}(Y_i, g_i, C) &= \sum_{\mathbf{y}\in Y_i} g_i(\mathbf{y}) d^2(\mathbf{y}, C) \\ &= \sum_{\mathbf{y}\in Y_i} |\text{cluster}(\mathbf{y}, P_i)| d^2(\mathbf{y}, C) \\ &= \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in \text{cluster}(\mathbf{y}, P_i)} d^2(\mathbf{y}, C). \end{aligned} \tag{18}$$

For any $\mathbf{x} \in \mathbb{R}^d$, recall that $C(\mathbf{x})$ denotes its closest center in C. From the above equality, we have

$$\begin{aligned}\text{cost}(Y_i, g_i, C) &= \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in \text{cluster}(\mathbf{y}, P_i)} d^2(\mathbf{y}, C(\mathbf{y})) \\ &\overset{(a)}{\leq} \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in \text{cluster}(\mathbf{y}, P_i)} d^2(\mathbf{y}, C(\mathbf{x})) \\ &\overset{(b)}{\leq} \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in \text{cluster}(\mathbf{y}, P_i)} (2d^2(\mathbf{x}, \mathbf{y}) + 2d^2(\mathbf{x}, C(\mathbf{x}))) \\ &= \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in \text{cluster}(\mathbf{y}, P_i)} 2d^2(\mathbf{x}, \mathbf{y}) + \sum_{\mathbf{x}\in P_i} 2d^2(\mathbf{x}, C(\mathbf{x})) \\ &= 2\text{cost}(P_i, Y_i) + 2\text{cost}(P_i, C) \\ &\overset{(c)}{\leq} (2+2\alpha)\text{cost}(P_i, C), \end{aligned} \tag{19}$$

where (a) follows from the definition of $C(\mathbf{x})$ and (b) follows from scaled triangular inequality. (c) follows from the fact that Y_i is a set of α-approximate k centers on the partial dataset P_i, $\text{cost}(P_i, Y_i) \leq \alpha\ \text{cost}(P_i, C)$. Next, we have

$$\begin{aligned}\text{cost}(Y, C, g) &= \sum_{i\in\mathcal{R}} \text{cost}(Y_i, C, b_i \cdot g_i) \\ &= \sum_{i\in\mathcal{R}} b_i \text{cost}(Y_i, C, g_i) \\ &\leq (2+2\alpha) \sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, C) \\ &\leq (2+2\alpha)(1+\delta)\text{cost}(P, C), \end{aligned} \tag{20}$$

where the first inequality follows from (19) and the second inequality follows from Lemma III.1.

Lower Bound: From Lemma III.1 for any set of k-centers C, we have

$$\begin{aligned}\text{cost}(P, C) &\leq \sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, C) \\ &= \sum_{i\in\mathcal{R}} b_i \sum_{\mathbf{x}\in P_i} d^2(\mathbf{x}, C(\mathbf{x})). \end{aligned} \tag{21}$$

From the definition of cluster centers, we know that for any two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and for any set of k-centers, $d^2(\mathbf{x}, C(\mathbf{x})) \leq d^2(\mathbf{x}, C(\mathbf{y}))$. Applying this observation in (21), we get

$$\begin{aligned}\text{cost}(P, C) &\leq \sum_{i\in\mathcal{R}} b_i \sum_{\mathbf{x}\in P_i} d^2(\mathbf{x}, C(\mathbf{x})) \\ &\leq \sum_{i\in\mathcal{R}} b_i \sum_{\mathbf{x}\in P_i} d^2(\mathbf{x}, C(Y_i(\mathbf{x}))), \end{aligned} \tag{22}$$

where $Y_i(\mathbf{x})$ is the cluster center in Y_i closest to $\mathbf{x} \in P_i$. Using scaled triangular inequality, we obtain

$$\begin{aligned}\text{cost}(P, C) &\leq \sum_{i\in\mathcal{R}} b_i \sum_{\mathbf{x}\in P_i} (2d^2(\mathbf{x}, Y_i(\mathbf{x})) \\ &\quad + 2d^2(Y_i(\mathbf{x}), C(Y_i(\mathbf{x})))) \\ &= \sum_{i\in\mathcal{R}} 2b_i \text{cost}(P_i, Y_i) \\ &\quad + \sum_{i\in\mathcal{R}} b_i \sum_{\mathbf{x}\in P_i} 2d^2(Y_i(\mathbf{x}), C(Y_i(\mathbf{x}))) \\ &= \sum_{i\in\mathcal{R}} 2b_i \text{cost}(P_i, Y_i) \\ &\quad + \sum_{i\in\mathcal{R}} b_i \sum_{\mathbf{x}\in Y_i} |\text{cluster}(\mathbf{y}, P_i)| 2d^2(\mathbf{y}, C(\mathbf{y})) \\ &= \sum_{i\in\mathcal{R}} 2b_i \text{cost}(P_i, Y_i) + \sum_{i\in\mathcal{R}} 2b_i \text{cost}(Y_i, C, g_i) \\ &= \sum_{i\in\mathcal{R}} 2b_i \text{cost}(P_i, Y_i) + \sum_{i\in\mathcal{R}} 2\text{cost}(Y_i, C, b_i \cdot g_i) \\ &= \sum_{i\in\mathcal{R}} 2b_i \text{cost}(P_i, Y_i) + 2\text{cost}(Y, C, g). \end{aligned} \tag{23}$$

Combining the upper and the lower bounds, we obtain the final result. □

Theorem IV.3. *Let C^* be the optimal set of k-means centers for dataset P. Then, Algorithm 2 on dataset P returns a set of centers $\hat{C}$ such that cost$(P, \hat{C}) \leq 2\alpha(3+2\alpha)(1+\delta)$cost$(P, C^*)$.*

Proof of Theorem IV.3. Utilizing the lower bound from Lemma A.3 with $C = \hat{C}$, we have

$$\begin{aligned}\text{cost}(P, \hat{C}) &\leq 2\text{cost}(Y, \hat{C}, g) + \sum_{i\in\mathcal{R}} 2b_i \text{cost}(P_i, Y_i) \\ &\overset{(a)}{\leq} 2\alpha\text{cost}(Y, C^*, g) + \ \alpha \sum_{i\in\mathcal{R}} 2b_i \text{cost}(P_i, C^*) \\ &\overset{(b)}{\leq} 2\alpha(2+2\alpha)(1+\delta)\text{cost}(P, C^*) \\ &\quad + 2\alpha(1+\delta)\text{cost}(P, C^*) \\ &= 2\alpha(3+2\alpha)(1+\delta)\text{cost}(P, C^*), \end{aligned} \tag{24}$$

where (a) follows from the fact that $\hat{C}$ and Y_i are the α-approximate set of centers for the weighted dataset (Y, g) and the partial dataset P_i, respectively. For (b), we utilize the upper bound in Lemma A.3 and Lemma III.1 with $C = C^*$. □

C. Straggler-Resilient Distributed (r,k)-Subspace Clustering

In this section, we present the missing proof of Lemma IV.4.

Lemma IV.4. *Let $\delta \in (0, 1)$. For any set of k-centers $C \subset \mathbb{R}^d$, we have*

$$(1-\delta)\text{cost}(P, C) \leq \text{cost}(Y, g, C) \leq (1+3\delta)\text{cost}(P, C).$$

Proof of Lemma IV.4. For any $i \in \mathcal{R}$, note that the weighted point set (Y_i, g_i) is an δ-coreset of the partial dataset P_i.

Hence, from the Definition II.4, we have that for any set of k-centers $C \subset \mathbb{R}^d$,

$$(1-\delta)\text{cost}(P_i, C) \leq \text{cost}(Y_i, g_i, C) \leq (1+\delta)\text{cost}(P_i, C). \tag{25}$$

For $Y = \cup_{i\in\mathcal{R}} Y_i$ and any set of k-centers C, we have

$$\begin{aligned}\text{cost}(Y, g, C) &= \sum_{\mathbf{y}\in Y} g(\mathbf{y}) d^2(\mathbf{y}, C) \\ &= \sum_{i\in\mathcal{R}} b_i \sum_{\mathbf{y}\in Y_i} g_i(\mathbf{y}) d^2(\mathbf{y}, C) \\ &= \sum_{i\in\mathcal{R}} b_i \text{cost}(Y_i, g_i, C). \end{aligned} \tag{26}$$

Combining (26) and (25), we get

$$\begin{aligned}(1-\delta)\sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, C) &\leq \text{cost}(Y, g, C) \\ &\leq (1+\delta)\sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, C). \end{aligned} \tag{27}$$

Now using the above inequality and Lemma III.1, we have

$$\begin{aligned}\text{cost}(Y, C, g) &\geq (1-\delta)\sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, C) \\ &\geq (1-\delta)\text{cost}(P, C), \end{aligned} \tag{28}$$

and

$$\begin{aligned}\text{cost}(Y, C, g) &\leq (1+\delta)\sum_{i\in\mathcal{R}} b_i \text{cost}(P_i, C) \\ &\leq (1+\delta)(1+\delta)\text{cost}(P, C) \\ &\leq (1+3\delta)\text{cost}(P, C) \qquad \text{for any } \delta \leq 1. \end{aligned} \tag{29}$$

Combining the upper and the lower bounds, we obtain the final result. □

BYZANTINE RESILIENT CLUSTERING

D. Byzantine Resilient Distributed k-Median

In this section, we present the proof of Lemma V.1. We restate the lemma here for the benefit of the reader.

Lemma V.1. *For any $i \in [m]$, the weighted point set (Y_i, g_i) satisfies*

$$\begin{aligned}\textit{cost}(P_i, C) - \textit{cost}(P_i, Y_i) &\leq \textit{cost}(Y_i, g_i, C) \\ &\leq \textit{cost}(P_i, C) + \textit{cost}(P_i, Y_i). \end{aligned} \tag{7}$$

Proof of Lemma V.1. We prove both sides of the inequality separately.

Upper Bound: Using the definitions of $\text{cost}(Y_i, g_i, C)$, and $g_i(\mathbf{y}) = |\text{cluster}(\mathbf{y}, P_i)|$, we get

$$\text{cost}(Y_i, g_i, C) = \sum_{\mathbf{y}\in Y_i} \sum_{x\in\text{cluster}(\mathbf{y}, P_i)} d(\mathbf{y}, C(\mathbf{y})) \tag{30}$$

$$\leq \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in\text{cluster}(\mathbf{y}, P_i)} d(\mathbf{y}, C(\mathbf{x})). \tag{31}$$

Applying triangular inequality, we obtain

$$\text{cost}(Y_i, g_i, C) \leq \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in\text{cluster}(\mathbf{y}, P_i)} (d(\mathbf{x}, \mathbf{y}) + d(\mathbf{x}, C(\mathbf{x}))). \tag{32}$$

Splitting the summation into two terms, simplifying further, and utilizing the definition of $\text{cost}(\cdot, \cdot)$ yields the final result as the following.

$$\begin{aligned}\text{cost}(Y_i, g_i, C) &\leq \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in\text{cluster}(\mathbf{y}, P_i)} d(\mathbf{x}, \mathbf{y}) + \sum_{\mathbf{x}\in P_i} d(\mathbf{x}, C(\mathbf{x})) \\ &= \text{cost}(P_i, Y_i) + \text{cost}(P_i, C). \end{aligned} \tag{33}$$

Lower Bound: For any machine $i \in [m]$, we have $\text{cost}(P_i, C) = \sum_{\mathbf{x}\in P_i} d(\mathbf{x}, C(\mathbf{x}))$. Let $Y_i(x)$ be the cluster center in Y_i that is closest to $\mathbf{x} \in P_i$. Then, we get $\text{cost}(P_i, C) \leq \sum_{\mathbf{x}\in P_i} d(\mathbf{x}, C(Y_i(\mathbf{x})))$, applying triangular inequality, we have

$$\text{cost}(P_i, C) \leq \sum_{\mathbf{x}\in P_i} d(\mathbf{x}, Y_i(\mathbf{x})) + \sum_{\mathbf{x}\in P_i} d(Y_i(\mathbf{x}), C(Y_i(\mathbf{x}))).$$

simplifying further, and utilizing the definitions of $\text{cost}(P_i, Y_i)$ and $\text{cost}(Y_i, g_i, C)$, we obtain the final result.

$$\begin{aligned}\text{cost}(P_i, C) &\leq \text{cost}(P_i, Y_i) + \sum_{\mathbf{y}\in Y_i} |\text{cluster}(\mathbf{y}, P_i)| d(\mathbf{y}, C(\mathbf{y})) \\ &= \text{cost}(P_i, Y_i) + \text{cost}(Y_i, g_i, C), \end{aligned}$$

□

E. Improved Byzantine Resilient Distributed k-Median

Lemma V.4. *Let $\gamma \geq \frac{1}{k}$. For any $i \in [m]$, and $\mathbf{y} \in Y_i$,*

$$\Pr[|\tilde{g}_i(\mathbf{y}) - g_i(\mathbf{y})| \geq \gamma\ g_i(\mathbf{y})] \leq \frac{1}{k}$$

Proof of Lemma V.4. The weight $\tilde{g}_i(\mathbf{y})$ can be written as

$$\begin{aligned}\tilde{g}_i(\mathbf{y}) &= \sum_{p\in\text{cluster}(\mathbf{y}, \tilde{P}_i)} w_i(p) \\ &= \sum_{p\in\text{cluster}(\mathbf{y}, P_i)} w_i(p)\mathbb{1}(p \in \tilde{P}_i). \end{aligned} \tag{34}$$

Applying expectation on both sides where the randomness is due to the sampling while constructing the coreset [31], we obtain $\mathbb{E}[\tilde{g}_i(\mathbf{y})] = \sum_{p\in\text{cluster}(\mathbf{y}, P_i)} w_i(p)\mathbb{P}(p \in \tilde{P}_i)$. From [31], we know that $w_i(p)\mathbb{P}(p \in \tilde{P}_i) = 1$. Therefore, we have

$$\mathbb{E}[\tilde{g}_i(\mathbf{y})] = \sum_{p\in\text{cluster}(\mathbf{y}, P_i)} 1 = g_i(\mathbf{y}). \tag{35}$$

From Chernoff's inequality, we have $\mathbb{P}(|g_i(\mathbf{y}) - \tilde{g}_i(\mathbf{y})| \leq \gamma g_i(\mathbf{y})) \geq 1 - e^{-2\gamma^2 g_i(\mathbf{y})^2 |P_i|}$, for a given $i \in [m-t]$ and $\mathbf{y} \in Y_i$.

Taking union bound over all $i \in [m-t]$ and $\mathbf{y} \in Y_i$, and setting $\gamma^2 \geq \frac{\log k}{(\min_{i,y} g_i(\mathbf{y})^2 |P_i|)\log(k(m-t))}$, the above inequality holds with probability at least $1 - \frac{1}{k}$. We assume that a cluster includes itself, thus ensuring that $g_i(\mathbf{y}) \geq 1$. Note that an upper bound for $g_i(\mathbf{y})$ and $|P_i|$ is n. Therefore, $\gamma^2 \geq \frac{\log k}{n^3 \log(k(m-t))}$. Thus, we choose $\gamma = 1/k$ which satifies the inequality. □

Lemma V.5. *Let $\delta, \gamma \in (0, 1)$. For any $i \in [m]$, the weighted point set $(Y_i, \tilde{g}_i)$ satisfies*

$$\begin{aligned}&(1-\gamma)\textit{cost}(P_i, C) - \frac{1-\gamma}{1-\delta}\textit{cost}(\tilde{P}_i, w_i, Y_i) \leq \textit{cost}(Y_i, \tilde{g}_i, C) \\ &\leq (1+\delta)\textit{cost}(P_i, C) + \textit{cost}(\tilde{P}_i, w_i, Y_i). \end{aligned}$$

Proof of Lemma V.5. We prove the upper and the lower bound separately.

Upper bound: Expanding the definition of $\text{cost}(Y_i, \tilde{g}_i, C)$ and $\tilde{g}_i(\mathbf{y})$ for any $\mathbf{y} \in Y_i$, we have

$$\begin{aligned}
\text{cost}(Y_i, \tilde{g}_i, C) &= \sum_{\mathbf{y}\in Y_i} \tilde{g}_i(\mathbf{y}) d(\mathbf{y}, C(\mathbf{y})) \\
&\leq \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in \text{cluster}(\mathbf{y},\tilde{P}_i)} w_i(\mathbf{x}) d(\mathbf{y}, C(\mathbf{x})) \\
&\overset{(a)}{\leq} \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in \text{cluster}(\mathbf{y},\tilde{P}_i)} w_i(\mathbf{x}) d(\mathbf{y}, \mathbf{x}) \\
&\quad + \sum_{\mathbf{y}\in Y_i} \sum_{\mathbf{x}\in \text{cluster}(\mathbf{y},\tilde{P}_i)} w_i(\mathbf{x}) d(\mathbf{x}, C(\mathbf{x})) \\
&= \text{cost}(\tilde{P}_i, w_i, Y_i) + \text{cost}(\tilde{P}_i, w_i, C) \\
&\overset{(b)}{\leq} \text{cost}(\tilde{P}_i, w_i, Y_i) + (1+\delta)\text{cost}(P_i, C).
\end{aligned} \tag{36}$$

The inequality (a) follows from triangular inequality, and (b) follows from the fact that $(\tilde{P}_i, w_i)$ is a δ-coreset of P_i as mentioned in Observation V.1.

Lower bound: For any machine $i \in [m]$ with any set of centers C, we have $\text{cost}(P_i, C) = \sum_{\mathbf{x}\in P_i} d(\mathbf{x}, C(\mathbf{x}))$. Let $Y_i(x)$ be the cluster center in Y_i that is closest to $x \in P_i$. Then, we get $\text{cost}(P_i, C) \leq \sum_{\mathbf{x}\in P_i} d(\mathbf{x}, C(Y_i(\mathbf{x})))$, applying triangular inequality, we have

$$\text{cost}(P_i, C) \leq \sum_{\mathbf{x}\in P_i} d(\mathbf{x}, Y_i(\mathbf{x})) + \sum_{\mathbf{x}\in P_i} d(Y_i(\mathbf{x}), C(Y_i(\mathbf{x}))).$$

For any $\mathbf{y} \in Y_i$, define $g_i(\mathbf{y}) := |\text{cluster}(\mathbf{y}, P_i)|$. Simplifying further, and utilizing the definitions of $\text{cost}(P_i, Y_i)$ and $\text{cost}(Y_i, g_i, C)$, we obtain

$$\begin{aligned}
\text{cost}(P_i, C) &\leq \text{cost}(P_i, Y_i) + \sum_{\mathbf{y}\in Y_i} |\text{cluster}(\mathbf{y}, P_i)| d(\mathbf{y}, C(\mathbf{y})) \\
&= \text{cost}(P_i, Y_i) + \text{cost}(Y_i, g_i, C).
\end{aligned} \tag{37}$$

Now, using Observation V.2, we know that with probability at least $1 - 1/k$, $\text{cost}(Y_i, g_i, C) \leq \frac{1}{1-\gamma}\text{cost}(Y_i, \tilde{g}_i, C)$. Plugging this back in Equation 37, and rearranging the terms, we get that

$$\begin{aligned}
\text{cost}(Y_i, \tilde{g}_i, C) &\geq (1-\gamma)\text{cost}(P_i, C) - (1-\gamma)\text{cost}(P_i, Y_i) \\
&\geq (1-\gamma)\text{cost}(P_i, C) - \frac{(1-\gamma)}{1-\delta}\text{cost}(\tilde{P}_i, w_i, Y_i),
\end{aligned}$$

where the last inequality follows from the fact that $(\tilde{P}_i, w_i)$ is a δ-coreset of P_i (Observation V.1). □

Theorem V.3. *Let $\delta \in (0,1)$. Let C^* be the optimal solution to the k-median problem on point set P. Then, Algorithm 5 returns a set of k-centers $\hat{C}$ such that cost$(P, \hat{C}) \leq \left(\frac{2}{1-1/k} + \frac{1}{1-\delta}\right)(1+3\delta)$cost$(P, C^*)$, even in the presence of t Byzantines with probability $1 - \frac{1}{k}$.*

Proof of Theorem V.3. We need to show that $\text{cost}(P, \hat{C}) \leq \alpha\text{cost}(P, C^*)$, for some $\alpha \geq 1$. Starting from the LHS, using the lower bound from Lemma V.5, we get

$$\begin{aligned}
\text{cost}(P, \hat{C}) &\leq \sum_{i=1}^{m-t} \rho\ \text{cost}(P_i, \hat{C}) \\
&\leq \underbrace{\frac{\rho}{1-\delta} \sum_{i=1}^{m-t} \text{cost}(\tilde{P}_i, w_i, Y_i)}_{(Term1)} \\
&\quad + \underbrace{\frac{\rho}{1-\gamma} \sum_{i=1}^{m-t} \text{cost}(Y_i, \tilde{g}_i, \hat{C})}_{(Term2)}.
\end{aligned} \tag{38}$$

We now bound each of the terms in Equation 38 separately.

a) Term 1:

$$\begin{aligned}
\frac{\rho}{1-\delta} \sum_{i=1}^{m-t} \text{cost}(\tilde{P}_i, w_i, Y_i) &\overset{(a)}{\leq} \frac{\rho}{1-\delta} \sum_{i\in\mathcal{R}} \text{cost}(\tilde{P}_i, w_i, Y_i) \\
&\overset{(b)}{\leq} \frac{\rho(1+\delta)}{1-\delta} \sum_{i\in\mathcal{R}} \text{cost}(P_i, Y_i) \\
&\overset{(c)}{\leq} \frac{\rho(1+\delta)}{1-\delta} \sum_{i\in\mathcal{R}} \text{cost}(P_i, C^*) \\
&\overset{(d)}{\leq} \frac{(1+\delta)^2}{1-\delta} \text{cost}(P, C^*).
\end{aligned} \tag{39}$$

Since for every Byzantine in the first $[m-t]$ machines, there will exist an honest machine with higher cost, (a) follows. (b) follows from the fact that $(\tilde{P}_i, w_i)$ is a δ-coreset of P_i. The optimality of the centers Y_i on P_i computed at the honest machines implies (c). Finally, (d) follows from the property of the assignment matrix shown in Lemma III.2.

b) Term 2:

$$\begin{aligned}
\frac{\rho}{1-\gamma} \sum_{i=1}^{m-t} \text{cost}(Y_i, \tilde{g}_i, \hat{C}) &\overset{(a)}{=} \frac{1}{1-\gamma} \text{cost}(Y, \tilde{g}, \hat{C}) \\
&\overset{(b)}{\leq} \frac{1}{1-\gamma} \text{cost}(Y, \tilde{g}, C^*) \\
&\overset{(c)}{\leq} \frac{\rho}{1-\gamma} \sum_{i=1}^{m-t} \text{cost}(Y_i, \tilde{g}_i, C^*)
\end{aligned} \tag{40}$$

(a) and (c) follow from the definitions of Y and $\tilde{g}$, and the optimality of the k-centers $\hat{C}$ on (Y, g) implies (b).

Now using the upper bound from Lemma V.5, continuing from Equation 40, we get

$$\begin{aligned}
\frac{\rho}{1-\gamma} \sum_{i=1}^{m-t} \text{cost}(Y_i, \tilde{g}_i, \hat{C}) &\leq \frac{\rho}{1-\gamma} \sum_{i=1}^{m-t} \text{cost}(Y_i, \tilde{g}_i, C^*) \\
&\leq \underbrace{\frac{\rho(1+\delta)}{1-\gamma} \sum_{i=1}^{m-t} \text{cost}(P_i, C^*)}_{\text{Term 21}} \\
&\quad + \underbrace{\frac{\rho}{1-\gamma} \sum_{i=1}^{m-t} \text{cost}(\tilde{P}_i, w_i, Y_i)}_{\text{Term 22}}
\end{aligned} \tag{41}$$

Term 21, by the property of the assignment matrix is equivalent to

$$\frac{\rho(1+\delta)}{1-\gamma}\sum_{i=1}^{m-t}\text{cost}(P_i,C^*)=\frac{(1+\delta)^2}{1-\gamma}\ \text{cost}(P,C^*),$$

(from Lemma III.2)

Also, observe that Term 22 is just a scaled version of Term 1 simplified above in Equation 39. Therefore,

$$\frac{\rho}{1-\gamma}\sum_{i=1}^{m-t}\text{cost}(\tilde{P}_i,w_i,Y_i)\le\frac{(1+\delta)^2}{1-\gamma}\ \text{cost}(P,C^*)$$

Plugging these two inequalities back in Equation 41, we get that Term 2 is bounded by

$$\frac{\rho}{1-\gamma}\sum_{i=1}^{m-t}\text{cost}(Y_i,\tilde{g}_i,\hat{C})\le\frac{2(1+\delta)^2}{1-\gamma}\ \text{cost}(P,C^*)\quad(42)$$

Finally, combining Equation 39 and Equation 42 in Equation 38 we get

$$\begin{aligned}\text{cost}(P,\hat{C})&\le(1+\delta)^2\left(\frac{1}{1-\delta}+\frac{2}{1-\gamma}\right)\ \text{cost}(P,C^*)\\&\le(1+3\delta)\left(\frac{1}{1-\delta}+\frac{2}{1-\gamma}\right)\ \text{cost}(P,C^*)\\&\text{(for any }\delta\in(0,1]).\end{aligned}$$

□

Construction of Data Assignment Matrix

F. Random Construction for Adversarial Byzantines

Theorem VI.1. *For any $\delta>0$, the Bernoulli assignment matrix A with $p=O(\frac{1}{\log m})$, satisfies Property III.2 with probability at least $1-O(\frac{1}{m})$, and is resilient to $t=O(\frac{m}{\log^2 m})$ Byzantines.*

Proof of Theorem VI.1. The proof follows from the observation that on deleting any set of t rows, the column weights in $A_{\mathcal{R}}$ are almost preserved with high probability.

Let $\mathcal{B}\subset[m]$ denote a fixed set of t Byzantines. Note that on deleting a fixed set of t the rows of A indexed by $\mathcal{B}\subset[m]$, the expected weight of a fixed column j is $p(m-t)$. Therefore, from standard Chernoff bounds it follows that

$$\Pr[|\text{wt}(A'_j)-p(m-t)|\ge\gamma p(m-t)]\le e^{-\frac{\gamma^2}{3}p(m-t)},$$

where $\text{wt}(A'_j)$ denotes the number of non-zero entries in the j-th column of $A_{\mathcal{R}}$ - the submatrix of A obtained from deleting the rows in $\mathcal{B}$.

By a union bound over all $\binom{m}{t}$ subsets of rows and all $n\ (=m)$ columns of A, we get that with probability at least $1-n\cdot m^t\cdot e^{-\frac{\gamma^2}{3}p(m-t)}$, all columns of A will have weight in the range $[(1-\gamma)p(m-t),(1+\gamma)p(m-t)]$. Therefore, setting $\rho=(1-\gamma)p(m-t)$, we get that for any set $\mathcal{B}$ of t rows, $\mathbf{1}_n^T\le\rho\ \sum_{i\in[m]\setminus\mathcal{B}}\mathbf{a}_i\le(1+\delta)\mathbf{1}_n^T$, for $\delta=\frac{2\gamma}{1-\gamma}$.

Setting $p=O\left(\frac{t\log m(2+\delta)^2}{(m-t)\delta^2}\right)$, A satisfies Property III.2 with probability at least $1-1/m$. In particular, for $p=O(1/\log m)$, the result follows for any $t=O(m/\log^2 m)$, with probability at least $1-1/m$. □

G. Explicit Construction for Adversarial Byzantines

Theorem VI.2. *For any $\delta>0$, the assignment matrix A satisfies Property III.2 with $t=\sqrt{\log m/\log\log m}$, and $\ell=O(\log m)$.*

Proof of Theorem VI.2. Let $G=(L\cup R,E)$ be the double cover of a c-regular expander graph on m vertices with expansion $\lambda=\max\{|\lambda_2|,|\lambda_m|\}$

We construct the $m\times m$ assignment matrix A from G by setting $A_{u,v}=1$ if there is an edge between $(u,v)\in G$ for any $u\in R$ and, $v\in L$. Note that each column of A has weight exactly c. Also, any set of t Byzantines will now correspond to a set of t vertices in R. We show that removing any set of t vertices from R does not reduce the individual degrees of any vertex $v\in L$ by a lot. This implies that the column weight in $A_{\mathcal{R}}$ is almost preserved.

Using Expander Mixing Lemma, we get that for any vertex $v\in L$, and any set of t vertices $B\subset R$,

$$|E(\{v\},B)|\le\frac{c}{m}t+\lambda\sqrt{t}=c\left(\frac{t}{m}+\frac{\lambda}{c}\sqrt{t}\right).$$

Therefore, for $\left(\frac{t}{m}+\frac{\lambda}{c}\sqrt{t}\right)=\gamma$, all vertices $v\in L$ are connected to at most $c\gamma$ machines in any set of t machines in R. So on deleting any set of t vertices in R all the vertices $v\in L$ will have degree $\deg(v)\in[(1-\gamma)c,c]$.

Therefore, setting $\rho=\frac{1}{c(1-\gamma)}$, we satisfy $\sum_{i\in\mathcal{R}}\mathbf{a}_i\le\frac{1}{1-\gamma}\mathbf{1}_n^T=(1+\delta)\mathbf{1}_n^T$, for $\gamma=\frac{\delta}{1+\delta}$.

Using the expander constructions in [49], we get an assignment scheme that is resilient to any set of $t=O(\sqrt{\log m/\log\log m})$ Byzantines with an overhead of $O(\log m)$ points per machine. □

Construction for Random Straggler Model

H. Randomized Construction for Random Byzantines

Theorem VII.1. *For any $\delta>0$, the randomized assignment matrix in (9) with $\ell=\frac{6(2+\delta)^2}{\delta^2}\cdot\frac{\log(\sqrt{2}m)}{1-p_t}$ satisfies Property III.1 with probability at least $1-\frac{1}{m}$ under the random straggler model.*

Proof of Theorem VII.1. Recall that $\mathcal{R}\subseteq[m]$ denotes the set of non-stragglers. Then, for any $i\in[m]$, we have

$$\mathbb{P}(i\in\mathcal{R})=1-p_t.\quad(43)$$

Next, we argue that for any $\delta>0$, we can choose $p_a=\frac{\ell}{m}$ large enough to ensure Property III.1 with high probability. First, we analyze the weight of each of the column in the random matrix. For $i\in[m]$ and $j\in[n]$, define an event $E_{i,j}$ as follows

$$E_{i,j}=\begin{cases}1 & \text{if } i\in\mathcal{R}\text{ and }A_{i,j}=1\\0 & \text{otherwise.}\end{cases}\quad(44)$$

Note that for any fixed $j\in[n]$, $\{E_{i,j}\}_{i\in[m]}$ is a collection of m independent events. Further, it follows from (9) and (43) that $\mathbb{P}(E_{i,j}=1)=p_a(1-p_t)$. Note that $\mathbb{E}\left[\sum_{i=1}^m E_{i,j}\right]=mp_a(1-p_t)=\ell(1-p_t)$. It then follows from standard Chernoff bound that for any $\gamma\in(0,1)$, we have

$$\mathbb{P}\left(\left|\sum_{i=1}^m E_{i,j}-\ell(1-p_t)\right|\ge\gamma(1-p_t)\right)\le2e^{-\frac{\gamma^2\ell(1-p_t)}{3}}.\quad(45)$$

Specifically, if we choose $\gamma = \frac{\delta}{2+\delta}$ and $\ell = \frac{6\log(n\sqrt{2})}{\gamma^2(1-p_t)}$, then with probability at least $1-\frac{1}{n^2}$ the following holds for a given $j \in [n]$

$$1 \leq \frac{1}{(1-\gamma)\ell(1-p_t)} \sum_{i=1}^{m} E_{i,j} \leq 1+\delta.$$

Now, taking a union bound over all $j \in [n]$, we have with probability at least $1-\frac{1}{n}$,

$$1 \leq \frac{1}{(1-\gamma)\ell(1-p_t)} \sum_{i=1}^{m} E_{i,j} \leq 1+\delta, \forall j \in [n]. \quad (46)$$

Recall that to establish Property III.1, we need to show that there exists a non-negative vector $\mathbf{b} \in \mathbb{R}^{|\mathcal{R}|}$ such that

$$\mathbf{1}_n^T \leq \mathbf{b}^T A_{\mathcal{R}} = (a_1, \ldots, a_n) \leq (1+\delta)\mathbf{1}_n^T.$$

Setting $\mathbf{b} = \frac{1}{(1-\gamma)\ell(1-p_t)} \cdot (1, \ldots, 1)$ as a candidate, we have

$$\mathbf{b}^T A_{\mathcal{R}} = \frac{1}{(1-\gamma)\ell(1-p_t)} \cdot \left(\sum_{i=1}^{m} E_{i,1}, \ldots, \sum_{i=1}^{m} E_{i,n} \right).$$

It follows from (46) that with probability at least $1-\frac{1}{n}$, each of the coordinates of $\mathbf{b}^T B_{\mathcal{R}}$ falls in the interval $[1, 1+\delta]$. This completes the proof. □

I. Explicit Construction for Random Byzantines

Theorem VII.2. *For any $\delta > 0$, the FRC based assignment matrix A with $\ell = s = O(\log m)$, satisfies Property III.1 with probability at least $1 - O(\frac{1}{m})$ under the random straggler model, and provides resilience against $t = O(m)$ stragglers.*

Proof of Theorem VII.2. Recall that $\mathcal{R} \subseteq [m]$ indicates the set of honest machines. Then, for any $i \in [m]$, we have

$$\Pr\{i \in \mathcal{R}\} = 1 - p_t. \quad (47)$$

Next, we show that the proposed construction satisfies Property III.1 with high probability.

Consider the block of $\mathbf{B}_i = \mathbf{1}_{s \times s}$, of A for any $i \in [m/s]$. First we show that for any block and a random set $\mathcal{R}$ of non-straggler machines, the weights of every column concentrates around it expected values.

For any block $i \in [m/s]$ and row in block $j \in [s]$, we define an event $F_{i,j}$ as follows:

$$F_{i,j} = \begin{cases} 1 & \text{if row } j \text{ in block } i \in \mathcal{R} \\ 0 & \text{otherwise.} \end{cases} \quad (48)$$

From (47), we know that $\Pr\{F_{i,j} = 1\} = 1 - p_t$. Therefore, for any fixed block i of s rows, we have $\mathbb{E}\left[\sum_{j=1}^{s} F_{i,j}\right] = s(1-p_t)$.

Utilizing Chernoff bound, for any $\gamma \in (0,1)$, we have

$$\Pr\left\{ \left| \sum_{j=1}^{s} F_{i,j} - s(1-p_t) \right| \geq \gamma s(1-p_t) \right\} \leq 2e^{-\frac{\gamma^2 s(1-p_t)}{3}}. \quad (49)$$

So, with high probability, the random set of Byzantines leave about $s(1-p_t)(1 \pm \gamma)$ rows unaffected in each block. So summing over the rows in block i of $A_{\mathcal{R}}$, we get that with probability at least $1 - 2e^{-\frac{\gamma^2}{3}(s(1-p_t))}$,

$$s(1-p_t)(1-\gamma)\mathbf{1}_s^T \leq \sum_{j \in [s]} F_{i,j}\mathbf{B}_{i,j} \leq s(1-p_t)(1+\gamma)\mathbf{1}_s^T.$$

where, $\mathbf{B}_{i,j}$ denotes the j-th row in the i-th block $\mathbf{B}_i$.

Setting $\gamma = \frac{\delta}{2+\delta}$, then with high probability the following holds for a given $i \in [m/s]$.

$$\mathbf{1}_s^T \leq \frac{1}{(1-\gamma)s(1-p_t)} \sum_{j \in [s]} F_{i,j}\mathbf{B}_{i,j} \leq (1+\delta)\mathbf{1}_s^T. \quad (50)$$

Taking union bound over all blocks $i \in [m/s]$, we have with the probability at least $1 - \frac{2m}{s}e^{-\frac{\gamma^2}{3}(s(1-p_t))}$,

$$\mathbf{1}_s^T \leq \frac{1}{(1-\gamma)s(1-p_t)} \sum_{j \in [s]} F_{i,j}\mathbf{B}_{i,j} \leq (1+\delta)\mathbf{1}_s^T, \ \forall i \in [m/s]. \quad (51)$$

The result then follows from the fact that all the blocks are in mutually exclusive rows of A. Setting $s = O(\log m)$ for a constant p_t, we see that the assignment scheme satisfies Property III.2 with probability at least $1 - O(1/m)$ and $\rho = \frac{1}{(1-\gamma)s(1-p_t)}$, where $\gamma = \frac{\delta}{2+\delta}$. □

REFERENCES

[1] V. Gandikota, A. Mazumdar, and A. S. Rawat, "Reliable distributed clustering with redundant data assignment," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 2556–2561.

[2] S. Bulusu, V. Gandikota, A. Mazumdar, A. S. Rawat, and P. K. Varshney, "Byzantine resilient distributed clustering with redundant data assignment," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 2143–2148.

[3] S. Dasgupta, "The hardness of k-means clustering," *' Dept. Comput. Sci. Eng., Univ. California, San Diego, CA, USA, Tech. Rep. CS2008-0916*, 2008.

[4] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical computer science*, vol. 38, pp. 293–306, 1985.

[5] M.-F. F. Balcan, S. Ehrlich, and Y. Liang, "Distributed k-means and k-median clustering on general topologies," *Advances in neural information processing systems*, vol. 26, 2013.

[6] G. Malkomes, M. J. Kusner, W. Chen, K. Q. Weinberger, and B. Moseley, "Fast distributed k-center clustering with outliers on massive data," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15, 2015, p. 1063–1071.

[7] J. Chen, H. Sun, D. Woodruff, and Q. Zhang, "Communication-optimal distributed clustering," *Advances in Neural Information Processing Systems*, vol. 29, pp. 3727–3735, 2016.

[8] P. Awasthi, M. Balcan, and C. White, "General and robust communication-efficient algorithms for distributed clustering," *CoRR*, vol. abs/1703.00830, 2017.

[9] S. Guha, Y. Li, and Q. Zhang, "Distributed partial clustering," in *Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures*, ser. SPAA '17. Association for Computing Machinery, 2017, p. 143–152.

[10] A. Bhaskara and M. Wijewardena, "Distributed clustering via lsh based data partitioning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 570–579.

[11] C. Karakus, Y. Sun, S. Diggavi, and W. Yin, "Straggler mitigation in distributed optimization through data encoding," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[12] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 06–11 Aug 2017, pp. 3368–3376.

[13] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1514–1529, 2018.

[14] S. Dutta, G. Joshi, S. Ghosh, P. Dube, and P. Nagpurkar, "Slow and stale gradients can win the race: Error-runtime trade-offs in distributed sgd," in *International conference on artificial intelligence and statistics*. PMLR, 2018, pp. 803–812.

[15] B. Buyukates, E. Ozfatura, S. Ulukus, and D. Gündüz, "Gradient coding with dynamic clustering for straggler mitigation," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[16] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1920–1933, 2020.

[17] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, 1982.

[18] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, pp. 1–25, 2017.

[19] L. Su and J. Xu, "Securing distributed machine learning in high dimensions," *arXiv preprint arXiv:1804.10140*, 2018.

[20] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *Advances in neural information processing systems*, 2011, pp. 693–701.

[21] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 10–15 Jul 2018, pp. 5650–5659.

[22] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 119–129.

[23] N. Raviv, I. Tamo, R. Tandon, and A. G. Dimakis, "Gradient coding from cyclic mds codes and expander graphs," *IEEE Transactions on Information Theory*, vol. 66, no. 12, pp. 7475–7489, 2020.

[24] M. Glasgow and M. Wootters, "Approximate gradient coding with optimal decoding," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 3, pp. 855–866, 2021.

[25] S. Wang, J. Liu, and N. Shroff, "Fundamental limits of approximate gradient coding," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 3, no. 3, pp. 1–22, 2019.

[26] H. Wang, Z. Charles, and D. Papailiopoulos, "Erasurehead: Distributed gradient descent without delays using approximate gradient coding," *arXiv preprint arXiv:1901.09671*, 2019.

[27] D. Data, L. Song, and S. Diggavi, "Data encoding methods for byzantine-resilient distributed optimization," in *2019 IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 2719–2723.

[28] D. Data and S. Diggavi, "On byzantine-resilient high-dimensional stochastic gradient descent," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 2628–2633.

[29] D. Data and S. N. Diggavi, "Byzantine-resilient high-dimensional federated learning," *IEEE Transactions on Information Theory*, vol. 69, no. 10, pp. 6639–6670, 2023.

[30] A. Ghosh, R. K. Maity, S. Kadhe, A. Mazumdar, and K. Ramchandran, "Communication-efficient and byzantine-robust distributed learning," in *2020 Information Theory and Applications Workshop (ITA)*. IEEE, 2020, pp. 1–28.

[31] V. Braverman, D. Feldman, H. Lang, A. Statman, and S. Zhou, "Efficient coreset constructions via sensitivity sampling," in *Asian Conference on Machine Learning*. PMLR, 2021, pp. 948–963.

[32] S. Guha, Y. Li, and Q. Zhang, "Distributed partial clustering," *ACM Transactions on Parallel Computing (TOPC)*, vol. 6, no. 3, pp. 1–20, 2019.

[33] D. Feldman and L. J. Schulman, "Data reduction for weighted and outlier-resistant clustering," in *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 2012, pp. 1343–1354.

[34] D. Feldman, M. Schmidt, and C. Sohler, "Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering," in *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '13. USA: Society for Industrial and Applied Mathematics, 2013, p. 1434–1453.

[35] J. Byrka, K. Sornat, and J. Spoerhase, "Constant-factor approximation for ordered k-median," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 2018, pp. 620–631.

[36] A. Kumar, Y. Sabharwal, and S. Sen, "A simple linear time (1+/spl epsiv/)-approximation algorithm for k-means clustering in any dimensions," in *45th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 2004, pp. 454–462.

[37] D. Feldman and M. Langberg, "A unified framework for approximating and clustering data," in *Proceedings of the forty-third annual ACM symposium on Theory of computing*, 2011, pp. 569–578.

[38] K. Varadarajan and X. Xiao, "A near-linear algorithm for projective clustering integer points," in *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 2012, pp. 1329–1342.

[39] D. Feldman, M. Monemizadeh, C. Sohler, and D. P. Woodruff, "Coresets and sketches for high dimensional subspace approximation problems," in *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '10. USA: Society for Industrial and Applied Mathematics, 2010, p. 630–649.

[40] K. Varadarajan and X. Xiao, "On the sensitivity of shape fitting problems," in *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 18, Dagstuhl, Germany, 2012, pp. 486–497.

[41] D. Feldman, M. Schmidt, and C. Sohler, "Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering," *SIAM Journal on Computing*, vol. 49, no. 3, pp. 601–657, 2020.

[42] C. Sohler and D. P. Woodruff, "Strong coresets for k-median and subspace approximation: Goodbye dimension," in *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2018, pp. 802–813.

[43] Z. Feng, P. Kacham, and D. Woodruff, "Dimensionality reduction for the sum-of-distances metric," in *International conference on machine learning*. PMLR, 2021, pp. 3220–3229.

[44] V. Cohen-Addad, D. Saulpic, and C. Schwiegelshohn, "A new coreset framework for clustering," in *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 2021, pp. 169–182.

[45] V. Cohen-Addad, K. G. Larsen, D. Saulpic, and C. Schwiegelshohn, "Towards optimal lower bounds for k-median and k-means coresets," in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2022. New York, NY, USA: Association for Computing Machinery, 2022, p. 1038–1051.

[46] V. Braverman, V. Cohen-Addad, H.-C. S. Jiang, R. Krauthgamer, C. Schwiegelshohn, M. B. Toftrup, and X. Wu, "The power of uniform sampling for coresets," in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2022, pp. 462–473.

[47] V. Braverman, D. Feldman, H. Lang, and D. Rus, "Streaming coreset constructions for m-estimators," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[48] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bulletin of the American Mathematical Society*, vol. 43, no. 4, pp. 439–561, 2006.

[49] Y. Bilu and N. Linial, "Lifts, discrepancy and nearly optimal spectral gap," *Combinatorica*, vol. 26, no. 5, pp. 495–519, 2006.

[50] Z. Charles, D. Papailiopoulos, and J. Ellenberg, "Approximate gradient coding via sparse random graphs," *arXiv preprint arXiv:1711.06771*, 2017.

[51] P. Fränti and O. Virmajoki, "Iterative shrinking method for clustering problems," *Pattern Recognition*, vol. 39, no. 5, pp. 761–775, 2006.

[52] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: A new data clustering algorithm and its applications," *Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 141–182, 1997.

BIOGRAPHY SECTION

Saikiran Bulusu (Member, IEEE) received the B.Tech. degree from the MGIT, Hyderabad, in 2009, the M.Tech. in communication engineering from the IIT Madras, in 2012, and Ph.D. degree in electrical and computer engineering from Syracuse University University, in 2023. Since September 2023, he has been a Postdoctoral Scholar at the AI-EDGE Institute, The Ohio State University. His research interests include robust machine learning, distributed optimization, and compressed sensing.

Venkata Gandikota (Senior Member, IEEE) received the B.E. degree in CS from BITS Pilani, Goa, in 2010, and the M.S./Ph.D. degrees in CS from Purdue University in 2017. Previously, he has held postdoctoral appointments at Johns Hopkins University and University of Massachusetts at Amherst. He is currently an Assistant Professor of Electrical Engineering and Computer Science at Syracuse University. His research interests include coding theory, point lattices, and their applications to foundational machine learning tasks.

Arya Mazumdar (Senior Member, IEEE) received the Ph.D. degree from the University of Maryland, College Park, in 2011. He is a Professor at the University of California, San Diego. From 2015 to 2021, he was an Assistant Professor followed by an Associate Professor with the College of Information and Computer Sciences, University of Massachusetts Amherst. Prior to that, he was a Faculty Member with the University of Minnesota, Twin Cities, from 2013 to 2015; and a Post-Doctoral Researcher with the Massachusetts Institute of Technology, from 2011 to 2012. His research interests include coding theory, information theory, statistical learning, and distributed optimization. He was a recipient of multiple awards, including the Distinguished Dissertation Award for Ph.D. Thesis in 2011, NSF CAREER Award in 2015, EURASIP JASP Best Paper Award in 2020, and IEEE ISIT Jack K. Wolf Student Paper Award in 2010. He was a Distinguished Lecturer of the Information Theory Society for 2023-24, and currently serves as an Associate Editor for IEEE TRANSACTIONS ON INFORMATION THEORY and an Area Editor for Now Publishers Foundation and Trends in Communication and Information Theory series.

Ankit Singh Rawat received the B.Tech. degree in electrical engineering from the IIT Kanpur, Kanpur, India, in 2010, and the M.S. and Ph.D. degrees in electrical and computer engineering from The University of Texas at Austin, in 2012 and 2015, respectively. Since October 2018, he has been a Research Scientist with Google Research, New York City. Previously, he has held postdoctoral appointments at the Research Laboratory of Electronics, Massachusetts Institute of Technology, the College of Information and Computer Sciences, University of Massachusetts Amherst, and the Computer Science Department, Carnegie Mellon University. His research interests include coding theory, information theory, and statistical machine learning. He is a recipient of the Microelectronics and Computer Development Fellowship from The University of Texas at Austin.

Pramod K. Varshney (S'72–M'77–SM'82–F'97) was born in Allahabad, India, in 1952. He received the B.S. degree (Hons.) in electrical engineering and computer science and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana–Champaign in 1972, 1974, and 1976, respectively. Since 1976, he has been with Syracuse University, Syracuse, NY, USA, where he is currently a Distinguished Professor of Electrical Engineering and Computer Science and the Director of the Center for Advanced Systems and Engineering. His current research interests are in distributed sensor networks and data fusion, detection and estimation theory, and wireless communications. He was a James Scholar, a Bronze Tablet Senior, and a fellow with the University of Illinois at Urbana–Champaign. He is a member of the Tau Beta Pi. In 2000, he received the Third Millennium Medal from the IEEE and Chancellors Citation for exceptional academic achievement at Syracuse University. He is a recipient of the 1981 ASEE Dow Outstanding Young Faculty Award. He is also a recipient of the IEEE 2012 Judith A. Resnik Award, an Honorary Doctor of Engineering degree from Drexel University in 2014, and the ECE Distinguished Alumni Award from UIUC in 2015. He was the President of the International Society of Information Fusion in 2001. He is on the Editorial Board of the Journal on Advances in Information Fusion and the IEEE Signal Processing Magazine.