Byzantine Resilient Distributed Clustering with Redundant Data Assignment

Saikiran Bulusu*, Venkata Gandikota*, Arya Mazumdar†, Ankit Singh Rawat‡ and Pramod K. Varshney*
*Electrical Engineering & Computer Science Department, Syracuse University, Syracuse, NY 13202, sabulusu@syr.edu, 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 paper, we present robust variants of distributed clustering algorithms for large datasets distributed across multiple machines in the presence of Byzantines. We propose a redundant data assignment scheme that enables us to obtain global information about the entire dataset for clustering purposes even when some machines are adversarial in nature. 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 is to group a given set of data points such that similarity within a group is maximized and similarity across the groups is minimized. In this work, we aim to find a subset of data points, called cluster centers, that provide a good representation of the given dataset. The quality of the clusters is measured using a cost function. Commonly used cost functions for clustering are k-medians and k-means, where the goal is to find k centers that minimize the sum of the distances (or sum of the squared distances) of the individual points to their closest cluster center.

Due to the large sizes of datasets, the centralized clustering algorithms where the operations are performed on a single machine are no longer feasible for real world applications. Hence, clustering algorithms adapted for a distributed setup have gained popularity. In the 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 $\{P_1, \ldots, P_m\} \subset P$ and assign each of these subsets to a different machine. The setup involves the machines performing computation on the locally available data points and transmitting the obtained results to the FC. Then, the FC aggregates these results to obtain the final clustering result. Recent works have provided clustering algorithms in the distributed setup where the cost of clustering is bounded by a constant multiple of the cost of clustering obtained by the centralized algorithms [1]–[3].

The distributed nature of the system makes it vulnerable to adversarial attacks where some machines can potentially be Byzantines [4]. Each honest machine sends a set of k-centers to the FC. However, a Byzantine may transmit arbitrary values to the FC instead of the correct set of k-centers. The goal of the Byzantine machines is to gain the ability to influence the k-centers in one (or more) of the clusters. This may lead to poor quality solutions given by the distributed clustering algorithm at the FC. In this work, we assume the static Byzantine attack model where the nature of the machines does not change as the algorithm progresses, i.e., Byzantine machines remain adversarial for the entire process. Typically, a faulty machine in the setup or an adversary corrupting the machines may lead to adversarial behavior of the machines. A naive approach for dealing with the Byzantines is to ignore their presence and rely on vanilla clustering techniques. This may lead to clustering results of extremely poor quality. Another approach is to provide filters to identify and remove the Byzantines in the setup as proposed in the Byzantine machine learning literature [5]–[9].

Alternately, data can be distributed to the machines in a redundant manner such that the information obtained from a subset of machines is sufficient to compute the desired function on the entire dataset. Many redundant data distribution schemes have been recently proposed [10]–[16] to mitigate the effect of non-responsive or slow machines known as *stragglers*. These coding based approaches were further used to handle Byzantines as well [17]–[20], however these works mainly focus on linear computations and first-order methods for distributed optimization.

In [15], the authors present a data distribution scheme that enables the computation of a good-quality clustering solution in the presence of stragglers. In this work, we study the distributed clustering algorithms using coded computations in the presence of Byzantines. The formulation deals with a more general scenario where a subset of the machines are adversarial and can send arbitrary information. The identity of these adversarial machines is not known to the FC which constitutes the main bottleneck in obtaining Byzantine resilient clustering algorithms. In particular, we show that a modification of data distribution scheme of [15] allows us to compute provably good-quality cluster centers even in the presence of a relatively large number of Byzantines.

A. Our Results

In this work, we provide a clustering approach that generates a solution with a cost at most $c\cdot$OPT, for a small approximation factor $c \geq 1$ for the underlying dataset in the presence of at

most t Byzantines, where OPT denotes the cost of the best clustering solution. The following are our major contributions.

- We propose a Byzantine-resilient data assignment scheme that enables us to filter out Byzantines and compute good-quality clusters.
- We design a robust k-medians and k-means clustering approach that generates a constant factor approximate solution given a dataset P that is distributed across m machines where at most t machines are Byzantines.
- We propose various constructions of the assignment scheme that can withstand large number of random or adversarial Byzantines with little redundancy.
- Simulation results illustrate the excellent performance of our algorithm.

B. Outline and Notation

The problem statement and the system model are presented in Section II. The proposed algorithm is given in Section III-A. The k-medians clustering problem is considered and extended for k-means clustering in Section III-B and Section III-C, respectively. Constructions of assignment matrix are presented in Section IV. Simulation results are provided in Section V, followed by our conclusions in Section VI. Please refer to the full version for all the missing proofs.

Notation: Let $d(\mathbf{x}, \mathbf{y})$ denote the Euclidean distance between two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. Let $[n] = \{1, \ldots, n\}$, and let $\mathbf{1}_n$ denote a vector of all 1's of length n.

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 centers is usually measured by a cost function $\text{cost}(P, C)$. For k-medians, the cost function is defined as $\text{cost}(P, C) = \sum_{x \in P} d(\mathbf{x}, C)$, 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 $\text{cost}(P, C) = \sum_{x \in P} d^2(\mathbf{x}, C)$. For any data point $\mathbf{x} \in P$, and any set of centers C, we denote the cluster center by $C(\mathbf{x}) := \arg\min_{\mathbf{c} \in C} d(\mathbf{x}, \mathbf{c})$.

We consider the 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. Note that we require 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 Algorithms 1 and 2). This assumption is reasonable as in distributed optimization the server has access to entire dataset and distributes it among the clients. Moreover, one can possibly eliminate this requirement using sophisticated data structures like locality sensitive hash maps at the FC to approximately estimate the costs while incurring a slightly larger approximation factors in the global clustering solution. We denote the cluster of P_i associated with the center $\mathbf{y} \in C$ by $\text{cluster}(\mathbf{y}, P_i) := \{\mathbf{x} \in P_i | C(\mathbf{x}) = \mathbf{y}\}$.

Definition II.1 (α-approximate solution)**.** For any $\alpha > 1$, the set of cluster centers C, is an α-approximate solution to the 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 (minimum) set k-centers, $\text{cost}(P, C) \leq \alpha \cdot \text{OPT}$.

If the dataset P is weighted with an associated non-negative weight function $g : P \to \mathbb{R}$, the k-medians 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, C) is defined analogously.

In the distributed setup, each machine W_i has access to a partial dataset $P_i \subset P$. In distributed clustering, each machine transmits a summary of its local data to the fusion center (FC). Hence, an approximate solution to the clustering problem can be computed by aggregating the summaries received at the FC. To mitigate the effect of Byzantines, we assume that the FC can compute on the local summaries to evaluate the quality of the data sent by each local machine.

Problem Statement: In this paper, the main goal is to design the distributed clustering approach that is robust to the presence of Byzantines. Given a dataset P and distributed setup with m machines where at most t machines are Byzantines, we design a clustering approach that generates a solution with the cost at most $c \cdot$OPT, for a small approximation factor $c \geq 1$ for the k-medians and the k-means clustering problems.

III. Byzantine-resilient Clustering

We propose a modification of the initial data assignment to the machines to mitigate the effect of Byzantines. In particular, the assignment process incorporates redundancy so that every data point in the dataset P is mapped to multiple machines. This ensures that each data point affects the local computations performed at multiple machines. Therefore, the final clusters at the FC can be obtained by taking into account the contribution of most of the data points in P even though some of the machines are Byzantines. The assignment scheme with Byzantine-resilient property is introduced below. This property enables the aggregation of local computations from honest machines at the FC and preserves the relevant information present in the dataset P for clustering. This assignment scheme is used to obtain good-quality solutions to the k-medians and k-means clustering problems.

A. Byzantine-resilient Data Assignment

We denote by $A \in \{0, 1\}^{m \times n}$ 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 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.1 ((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 $\exists \rho > 0$ such that for any subset of $m - t$ rows $\mathcal{R} \subseteq [m]$,

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

where $\leq$ indicates coordinate-wise inequality.

We remark that the (t, δ)-Byzantine resilience property is significantly different from that in [10] where the property depends on the gradients that are related to each other across different machines. Furthermore, our resilience property is much stronger than the straggler resilience property introduced in [15]. 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. Furthermore, we also need this reconstruction factor to be the same across all subsets of Byzantines.

B. Byzantine-Resilient Distributed k-medians Clustering

The dataset P is distributed among the m machines using the assignment matrix A which satisfies Property III.1. The basic idea of the proposed algorithm is that each honest machine sends a set of k-medians centers of their respective data subsets. Then, the FC combines the set of k-medians centers from all the machines and computes on them to obtain additional information about the quality of the centers sent by each machine. Using this information, and the aggregated summary, the FC then computes a good-quality clustering solution for the entire dataset. We present the aforementioned steps in detail in Algorithm 1.

Algorithm 1 Byzantine-resilient distributed k-medians

1: **Initialize:** A collection of n vectors $P \subset \mathbb{R}^d$
2: Allocate P to m machines according to A with Property III.1.
3: Assign the set of points $P_i \subset P$ to worker W_i
4: Each honest worker W_i computes k-medians 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 k-medians solution on (Y, g).

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

Lemma III.1. *Let $\mathcal{T} \subseteq [m]$ 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 cost$(P, C) \leq \sum_{i \in \mathcal{T}} \rho$ cost$(P_i, C) \leq (1 + \delta)$cost$(P, C)$.*

Proof. Proof follows based on the combinatorial characterization for the assignment scheme enforced by Property III.1. □

We present the following intermediate results which show that the cost incurred by the weighted data subset Y_i at machine W_i is close to the cost incurred by the local data subset P_i for any set of k centers C.

Lemma III.2. *For any $i \in [m]$, the weighted point set (Y_i, g_i) satisfies cost$(Y_i, g_i, C) \leq$ cost$(P_i, C) +$ cost(P_i, Y_i).*

Lemma III.3. *Let C be any set of k-centers, then for any $i \in [m]$, we have cost$(Y_i, g_i, C) \geq$ cost$(P_i, C) -$ cost(P_i, Y_i).*

The above two lemmas show that the cost of clustering the weighted data subset (Y_i, g_i) obtained from W_i with any set of centers C, $\text{cost}(Y_i, g_i, C)$ is tightly bounded by $\text{cost}(P_i, C)$ and $\text{cost}(P_i, Y_i)$. Since the latter term can be computed by the FC, we can use this information to filter out any bad summaries. From these observations, we get our main result that evaluates the quality of the clustering solution, $\hat{C}$, obtained by Algorithm 1 on the entire dataset P.

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

C. Byzantine-Resilient Distributed k-means Clustering

In this section, we extend the results obtained for k-medians clustering to the k-means clustering problem. We use Algorithm 2, which is similar to Algorithm 1, to compute the k-means clustering. Here instead of returning the k-medians solution to the partial data set P_i, each machine returns the k-means centers.

Similar to the analysis of Algorithm 1, we show that cost of clustering weighted data subset sent by any machine with any set of centers $\text{cost}(Y_i, g_i, C)$ is tightly bounded by $\text{cost}(P_i, C)$ and $\text{cost}(P_i, Y_i)$. Using these observations, Algorithm 2 guarantees the following.

Theorem III.5. *Let C^* be the optimal solution to the k-means problem on point set P. Then, Algorithm 1 returns a set of k-centers $\hat{C}$ such that cost$(P, \hat{C}) \leq 10(1 + \delta)$cost$(P, C^*)$, even in the presence of t Byzantines.*

We remark that no redundancy is a baseline to compare our proposed algorithms with. We note that no redundancy might lead to complete loss of contribution from the subset of points that are assigned to the Byzantines. This phenomenon is evident in the simulation results provided in Section V.

IV. Construction of Data Assignment Matrix

In this section, we present various constructions of the data assignment matrices that satisfy Property III.1. We consider

Algorithm 2 Byzantine-resilient distributed k-means

1: **Initialize:** A collection of n vectors $P \subset \mathbb{R}^d$
2: Allocate P to m machines according to A with Property III.1.
3: Assign the set of points $P_i \subset P$ to worker 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 cost(P_i, Y_i).
8: Without loss of generality, assume cost$(P_1, Y_1) \leq$ cost$(P_2, Y_2) \leq \ldots \leq$ 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).

both the random and the adversarial Byzantine models for the construction of assignment matrices. For each of these constructions, we analyse the tradeoffs between the load per machine (ℓ) and the fraction of Byzantines that can be tolerated.

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 $n = m$. We now present the construction of various assignment matrices $A \in \{0,1\}^{m \times m}$ that satisfy Property III.1.

In the random Byzantine model, each machine W_i, for $i \in [m]$ behaves as a Byzantine independently with some fixed probability p_t. The Bernoulli matrix based randomized construction of straggler-resilient assignment matrix presented [15] was shown to be resilient to a constant fraction of stragglers with $\ell = O(\log m)$ load per machine. We note that their construction satisfies the Byzantine-resilience property (Property III.1) as well. We now provide an explicit construction of an assignment matrix based on Fractional Repetition Codes that is resilient to a constant fraction of Byzantines with $\ell = O(\log m)$ load per machine.

A. Explicit Construction for Random Byzantines

Fractional Repetition Codes (FRC) have been well-studied in [21] for straggler resilient gradient computations. In this section, we show that the FRC scheme also satisfies Property III.1 for random Byzantines with high probability, and hence provides redundant data assignment for Byzantine-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

$$A = \begin{pmatrix} \mathbf{1_{s\times s}} & \mathbf{0_{s\times s}} & \mathbf{0_{s\times s}} & \ldots & \mathbf{0_{s\times s}} \\ \mathbf{0_{s\times s}} & \mathbf{1_{s\times s}} & \mathbf{0_{s\times s}} & \ldots & \mathbf{0_{s\times s}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0_{s\times s}} & \mathbf{0_{s\times s}} & \mathbf{0_{s\times s}} & \ldots & \mathbf{1_{s\times s}} \end{pmatrix}, \tag{2}$$

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 IV.1. *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 Byzantine model, and provides resilience against $t = O(m)$ Byzantines.*

Theorem IV.1 provides good tradeoffs between the load per machine ℓ, and the number of Byzantines tolerated, t. However, the guarantees hold in the random Byzantine model. In the adversarial Byzantine model, any subset $\mathcal{B} \subset [m]$ can be Byzantines. This is a much stronger yet practical model for Byzantines. We now give two constructions - one randomized and one explicit construction that provide decent tradeoffs in the adversarial Byzantine model.

B. Random Construction for Adversarial Byzantines

In this section we show that a random Bernoulli assignment matrix satisfies Property III.1 under the adversarial Byzantine model 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.

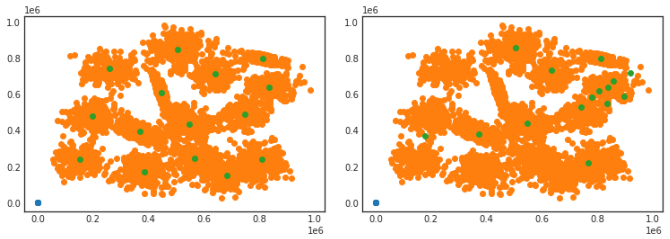
Theorem IV.2. *For any $\delta > 0$, the Bernoulli assignment matrix A with $p = O(\frac{1}{\log m})$, satisfies Property III.1 with probability at least $1 - O(\frac{1}{m})$ under the adversarial Byzantine model, and is resilient to $t = O(\frac{m}{\log^2 m})$ Byzantines.*

Alternatively, Theorem IV.2 can be stated as a randomized construction that is resilient to t arbitrary Byzantines with expected load of $O(\frac{mt}{m-t} \log m)$. Note that Theorem IV.2 provides lesser redundancy in the regime when $t = o(m)$ compared to the naïve solution of distributing all the points to all the machines.

C. Explicit Construction for Adversarial Byzantines

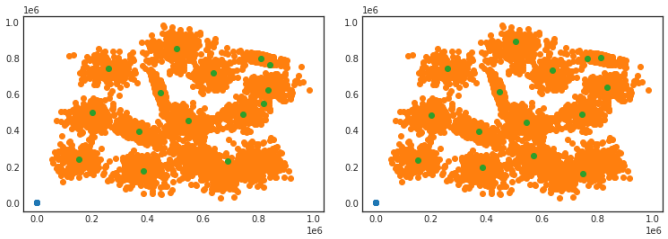
We now present an explicit construction of assignment matrix that satisfies Property III.1 in the adversarial Byzantine model. The construction is based on expander graphs which were recently used to construct explicit data assignment schemes for gradient coding [12], [13].

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 $\lambda = \max\{|\lambda_2|, |\lambda_m|\}$.



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

Fig. 1: Performance of the proposed Byzantine-resilient k-medians algorithm with no redundancy.



(a) $p = 0.1$. **(b)** $p = 0.2$.

Fig. 2: Performance of the proposed Byzantine-resilient k-medians algorithm.

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 (n, d, λ)-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 will satisfy Property III.1 for any set of t Byzantines.

Theorem IV.3. *For any $\delta > 0$, the assignment matrix A satisfies Property III.1 under adversarial Byzantine model with $t = \sqrt{\log m / \log\log m}$, and $\ell = O(\log m)$.*

The proof follows from the fact that if $\tilde{G}$ is an expander, then G satisfies the expander Mixing Lemma [22].

Theorem IV.4 (Expander Mixing Lemma)**.** *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 [23], 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 IV.5 ([23])**.** *There exists a polynomial time algorithm to construct $(n, d, \lambda) = (2^\ell, \ell - 1, \sqrt{\ell \log^3 \ell})$.*

V. Simulation Results

In this section, we illustrate the performance of our Byzantine-resilient distributed k-medians algorithm and benchmark it with the non-redundant data assignment scheme. We consider the synthetic Gaussian dataset [24] with $n = 5000$ two-dimensional points that are distributed among $m = 10$ machines with $t = 3$ randomly chosen Byzantines. We present the results in Figures 1a, 1b, 2a, and 2b.

We plot the ground truth using the centroids provided in the dataset in Fig. 1a with k-medians clustering, for $k = 15$. In Fig. 1b, we present the results by ignoring the local computations from the Byzantines, i.e., Algorithm 1 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-medians centers to the FC. Then, the FC runs a k-medians algorithm on the $(m - t)$ centers obtained from the honest machines. From Fig. 1b, the set of poor quality k-centers obtained from this scheme is noticeable.

In Fig. 2a, 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. 2b, 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. 1a.

VI. Conclusion

In this paper, we provided $O(1)$-approximate solutions for the distributed k-medians and k-means clustering problems in the presence of Byzantines. Note that the approach for k-means (Algorithm 2) used in this work can be generalized to obtain Byzantine-resilient algorithms for a larger class of ℓ_2 fitting problems such as (r, k)-subspace clustering solutions.

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 1 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 cost between the machines and the FC resulting in communication efficient clustering algorithms in the presence of Byzantines.

REFERENCES

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] M. Glasgow and M. Wootters, "Approximate gradient coding with optimal decoding," *arXiv preprint arXiv:2006.09638*, 2020.

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] D. Data and S. Diggavi, "Byzantine-resilient high-dimensional federated learning," *arXiv e-prints*, pp. arXiv–2006, 2020.

[20] 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.

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

[22] 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.

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

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

VII. APPENDIX

A. Missing Proofs from Section III-B

Proof of Lemma III.2. From the definition of $\text{cost}(Y_i, g_i, C)$, noting that $g_i(\mathbf{y}) = |\text{cluster}(\mathbf{y}, P_i)|$, and rewriting $|\text{cluster}(\mathbf{y}, P_i)|$ as $\sum_{x \in \text{cluster}(\mathbf{y}, P_i)}$, we get

$$\text{cost}(Y_i, g_i, C) = \sum_{\mathbf{y} \in Y_i} \sum_{x \in \text{cluster}(\mathbf{y}, P_i)} d(\mathbf{y}, C(\mathbf{y})). \tag{3}$$

For any $\mathbf{x} \in \mathbb{R}^d$, let $C(\mathbf{x})$ denote its closest center in C. It follows from (3) that

$$\text{cost}(Y_i, g_i, C) \leq \sum_{\mathbf{y} \in Y_i} \sum_{\mathbf{x} \in \text{cluster}(\mathbf{y}, P_i)} d(\mathbf{y}, C(\mathbf{x})). \tag{4}$$

Applying triangular inequality, we obtain

$$\text{cost}(Y_i, g_i, C) \leq \sum_{\mathbf{y} \in Y_i} \sum_{\mathbf{x} \in \text{cluster}(\mathbf{y}, P_i)} (d(\mathbf{x}, \mathbf{y}) + d(\mathbf{x}, C(\mathbf{x}))). \tag{5}$$

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

$$\begin{aligned} \text{cost}(Y_i, g_i, C) &\leq \sum_{\mathbf{y} \in Y_i} \sum_{\mathbf{x} \in \text{cluster}(\mathbf{y}, P_i)} d(\mathbf{x}, \mathbf{y}) + \sum_{\mathbf{x} \in P_i} d(\mathbf{x}, C(\mathbf{x})) \\ &= \text{cost}(P_i, Y_i) + \text{cost}(P_i, C). \end{aligned} \tag{6}$$

□

Proof of Lemma III.3. 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 $x \in P_i$. 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.

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

□

Proof of Theorem III.4. Let $\hat{C}$ be the set of k-centers returned by Algorithm 1. From Lemma III.1, 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 III.3 with $C = \hat{C}$, we get

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

Next, we note that for every Byzantine in $[m-t]$, there is an honest machine $i \in \mathcal{R}$ with a higher cost which yields the following.

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

The optimality of the k-centers Y_i on the partial dataset P_i yields $\text{cost}(P_i, Y_i) \leq \text{cost}(P_i, \hat{C})$. Hence, we have

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

applying the result from Lemma III.1 to the first term. Utilizing the definition of the cost function on a weighted point set, $\text{cost}(Y, g, \hat{C})$ and the optimality of Y_i on the partial dataset P_i in the second term, we obtain

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

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

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

Next, applying the result from Lemma III.2 to the second term above, we have

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

For the second term above, we use the fact that for every Byzantine in $[m-t]$, there is an honest machine $i \in \mathcal{R}$ with a higher cost, and the optimality of the centers Y_i on P_i to obtain

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

applying Lemma III.1 to the second and third terms, we obtain

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

□

B. Missing Proofs from Section III-C

Lemma VII.1 (Scaled Triangular Inequality)**.** *Let $d(\boldsymbol{x}, \boldsymbol{y})$ denotes the Euclidean distance between two points $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$. Let $\boldsymbol{z} \in \mathbb{R}^d$ be a point such that*

$$d^2(\boldsymbol{x}, \boldsymbol{y}) \leq 2\ d^2(\boldsymbol{x}, \boldsymbol{z}) + 2\ d^2(\boldsymbol{z}, \boldsymbol{y}).$$

Proof of Lemma VII.1. From the definition of $d(\mathbf{x}, \mathbf{y})$, we have

$$\begin{aligned}d^2(\mathbf{x}, \mathbf{y}) &= \|\mathbf{x} - \mathbf{y}\|^2 \\ &= \|\mathbf{x} - \mathbf{z} + \mathbf{z} - \mathbf{y}\|^2 \\ &= \|(\mathbf{x} - \mathbf{z}) + (\mathbf{z} - \mathbf{y})\|^2 \\ &\leq (\|\mathbf{x} - \mathbf{z}\| + \|\mathbf{z} - \mathbf{y}\|)^2,\end{aligned}$$

where the last inequality is from triangular inequality. Applying Cauchy-Schwarz inequality, we get

$$\begin{aligned}d^2(\mathbf{x}, \mathbf{y}) &\leq 2(\|\mathbf{x} - \mathbf{z}\|^2 + \|\mathbf{z} - \mathbf{y}\|^2) \\ &= 2\ d^2(\mathbf{x}, \mathbf{z}) + 2\ d^2(\mathbf{z}, \mathbf{y}),\end{aligned} \tag{7}$$

□

Lemma VII.2. *For any machine W_i, the weighted point set (Y_i, g_i) satisfies*

$$\textit{cost}(Y_i, g_i, C) \leq 2\ \textit{cost}(P_i, Y_i) + 2\ \textit{cost}(P_i, C).$$

Proof of Lemma VII.2. From the definition of k-means cost, $\text{cost}(Y_i, g_i, C)$, noting that $g_i(\mathbf{y}) = |\text{cluster}(\mathbf{y}, P_i)|$, and rewriting $|\text{cluster}(\mathbf{y}, P_i)|$ as $\sum_{x \in \text{cluster}(\mathbf{y}, P_i)}$, we get

$$\text{cost}(Y_i, g_i, C) = \sum_{\mathbf{y} \in Y_i} \sum_{x \in \text{cluster}(\mathbf{y}, P_i)} d^2(\mathbf{y}, C(\mathbf{y})). \tag{8}$$

For any $\mathbf{x} \in \mathbb{R}^d$, let $C(\mathbf{x})$ denote its closest center in C. It follows from (8) that

$$\text{cost}(Y_i, g_i, C) \leq \sum_{\mathbf{y} \in Y_i} \sum_{\mathbf{x} \in \text{cluster}(\mathbf{y}, P_i)} d^2(\mathbf{y}, C(\mathbf{x})),$$

applying the result from Lemma VII.1, we have

$$\begin{aligned}\text{cost}(Y_i, g_i, C) &\leq \sum_{\mathbf{y} \in Y_i} \sum_{\mathbf{x} \in \text{cluster}(\mathbf{y}, P_i)} (2\ d^2(\mathbf{x}, \mathbf{y}) + 2\ d^2(\mathbf{x}, C(\mathbf{x}))) \\ &= \sum_{\mathbf{y} \in Y_i} \sum_{\mathbf{x} \in \text{cluster}(\mathbf{y}, P_i)} 2\ d^2(\mathbf{x}, \mathbf{y}) + \sum_{\mathbf{x} \in P_i} 2\ d^2(\mathbf{x}, C(\mathbf{x})) \\ &= 2\ \text{cost}(P_i, Y_i) + 2\ \text{cost}(P_i, C),\end{aligned} \tag{9}$$

where the first equality follows by splitting the summation into two terms, and utilizing the definition of $\text{cost}(\cdot, \cdot)$ yields the final result. □

Lemma VII.3. *Let C be any set of k-centers, then for any machine W_i, we have*

$$\textit{cost}(P_i, C) \leq 2\ \textit{cost}(P_i, Y_i) + 2\ \textit{cost}(Y_i, g_i, C).$$

Proof of Lemma VII.3. For any machine $i \in [m]$, we have

$$\text{cost}(P_i, C) = \sum_{\mathbf{x} \in P_i} d^2(\mathbf{x}, C(\mathbf{x})).$$

Let $Y_i(x)$ be the cluster center in Y_i that is closest to $x \in P_i$. Then, we get

$$\text{cost}(P_i, C) \leq \sum_{\mathbf{x} \in P_i} d^2(\mathbf{x}, C(Y_i(\mathbf{x}))),$$

applying the result from Lemma VII.1, we have

$$\text{cost}(P_i, C) \leq \sum_{\mathbf{x} \in P_i} 2\ d^2(\mathbf{x}, Y_i(\mathbf{x})) + \sum_{\mathbf{x} \in P_i} 2\ d^2(Y_i(\mathbf{x}), C(Y_i(\mathbf{x}))),$$

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.

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

□

Proof of Theorem III.5. Let $\hat{C}$ be the set of k-centers returned by Algorithm 2. From Lemma III.1, 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 VII.3 with $C = \hat{C}$, we get

$$\text{cost}(P, \hat{C}) \leq \sum_{i=1}^{m-t} 2\rho\ \text{cost}(P_i, Y_i) + \sum_{i=1}^{m-t} 2\rho\ \text{cost}(Y_i, g_i, \hat{C}).$$

Using the fact that for every Byzantine in $[m-t]$, there is an honest machine $i \in \mathcal{R}$ with a higher cost to obtain the following.

$$\text{cost}(P, \hat{C}) \leq \sum_{i \in \mathcal{R}} 2\rho\ \text{cost}(P_i, Y_i) + \sum_{i=1}^{m-t} 2\rho\ \text{cost}(Y_i, g_i, \hat{C}).$$

The optimality of Y_i on the partial dataset P_i gives $\text{cost}(P_i, Y_i) \leq \text{cost}(P_i, \hat{C})$. Therefore, we have

$$\text{cost}(P, \hat{C}) \leq \sum_{i \in \mathcal{R}} 2\rho\ \text{cost}(P_i, C^*) + \sum_{i=1}^{m-t} 2\rho\ \text{cost}(Y_i, g_i, \hat{C})$$

applying the result from Lemma III.1 to the first term. Utilizing the definition of the cost function on a weighted point set, $\text{cost}(Y, g, \hat{C})$ and the optimality of Y_i on the partial dataset P_i in the second term, we obtain

$$\text{cost}(P, \hat{C}) \leq 2(1+\delta)\text{cost}(P, C^*) + 2\ \text{cost}(Y, g, C^*).$$

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

$$\begin{aligned}\text{cost}(P, \hat{C}) &\leq 2(1+\delta)\text{cost}(P, C^*) + 2 \sum_{i=1}^{m-t} \text{cost}(Y_i, \rho g_i, C^*) \\ &\leq 2(1+\delta)\text{cost}(P, C^*) + 2 \sum_{i=1}^{m-t} \rho\ \text{cost}(Y_i, g_i, C^*).\end{aligned}$$

Next, applying the result from Lemma VII.2 to the second term above, we have

$$\begin{aligned}\text{cost}(P, \hat{C}) \leq\ & 2(1+\delta)\text{cost}(P, C^*) + 4 \sum_{i=1}^{m-t} \rho\ \text{cost}(P_i, Y_i) \\ &+ 4 \sum_{i=1}^{m-t} \rho\ \text{cost}(P_i, C^*).\end{aligned}$$

We use the fact that for every Byzantine in $[m-t]$, there is an honest machine $i \in \mathcal{R}$ with a higher cost, and the optimality of the centers Y_i on P_i in the second term above, to obtain

$$\begin{aligned}\text{cost}(P, \hat{C}) \leq\ & 2(1+\delta)\text{cost}(P, C^*) + 4 \sum_{i \in \mathcal{R}} \rho\ \text{cost}(P_i, C^*) \\ &+ 4 \sum_{i=1}^{m-t} \rho\ \text{cost}(P_i, C^*),\end{aligned}$$

applying Lemma III.1 to the second and third terms, we obtain

$$\begin{aligned}\text{cost}(P, \hat{C}) \leq\ & 2(1+\delta)\text{cost}(P, C^*) + 4(1+\delta)\text{cost}(P, C^*) \\ &+ 4(1+\delta)\text{cost}(P, C^*) = 10(1+\delta)\text{cost}(P, C^*).\end{aligned}$$

□

C. Missing Proofs from Section IV

Proof of Theorem IV.1. Recall that $\mathcal{R} \subseteq [m]$ indicates the set of honest machines. Then, for any $i \in [m]$, we have

$$\Pr\{i \in \mathcal{R}\} = 1 - p_t. \tag{10}$$

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 honest 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:

$$F_{i,j} = \begin{cases} 1 & \text{if row } j \text{ in block } i \in \mathcal{R} \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

From (10), we know that

$$\Pr\{F_{i,j} = 1\} = 1 - p_t. \tag{12}$$

Therefore, for any block fixed block i of s rows, we have

$$\mathbb{E}\left[\sum_{j=1}^{s} F_{i,j}\right] = s(1 - p_t). \tag{13}$$

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

$$\Pr\left\{\left|\sum_{j=1}^{s} F_{i,j} - s(1-p_t)\right| \geq \gamma s(1-p_t)\right\} \leq 2e^{-\frac{\gamma^2 s(1-p_t)}{3}}. \tag{14}$$

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 $1-e^{-\Omega(s(1-p_t))}$,

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

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 $j \in [m]$.

$$\mathbf{1}_s^T \leq \frac{1}{(1-\gamma)s(1-p_t)} \sum_{j\in[s]} F_{i,j}\mathbf{B}_{i,j} \leq (1+\delta)\mathbf{1}_s^T. \quad (15)$$

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

$$\mathbf{1}_s^T \leq \frac{1}{(1-\gamma)s(1-p_t)} \sum_{j\in[s]} F_{i,j}\mathbf{B}_{i,j} \leq (1+\delta)\mathbf{1}_s^T,\ \forall i \in [m/s]. \quad (16)$$

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.1 with probabaility at least $1-O(1/m)$ and $\rho = \frac{1}{(1-\gamma)s(1-p_t)}$, where $\gamma = \frac{\delta}{2+\delta}$. □

Proof of Theorem IV.2. 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. the rows of A indexed by $B \subset [m]$, the expected weight of a fixed column j is $p(m-t)$. Therefore, from standard Chernoff bounds it follows that

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

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 B.

By a union bound over all $\binom{m}{t}$ subsets of rows and all n 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 of B of t rows,

$$\mathbf{1}_n^T \leq \rho \sum_{i\in[m]\setminus B} \mathbf{a}_i \leq (1+\delta)\mathbf{1}_n^T$$

for $\delta = \frac{2\gamma}{1-\gamma}$.

Setting $p = O(1/\log m)$, the result follows for any $t = O(m/\log^2 m)$, with probability at least $1-1/m$. □

Proof of Theorem IV.3. Let $G = (L \cup R, E)$ be the double cover of a c-regular expander graph on m vertices with expansion $\lambda = \max\{|\lambda_2|, |\lambda_n|\}$

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 $B \subset R$,

$$\begin{aligned}|E(\{v\}, B)| &\leq \frac{c}{m}t + \lambda\sqrt{t} \\ &= c\left(\frac{t}{m} + \frac{\lambda}{c}\sqrt{t}\right).\end{aligned}$$

Therefore, for $\left(\frac{t}{m} + \frac{\lambda}{c}\sqrt{t}\right) = \gamma$, all vertices $v \in L$ are connected to at most $c\gamma$ nodes in any set of t nodes 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 $\sum_{i\in\mathcal{R}} \mathbf{a}_i \leq \frac{1}{1-\gamma}\mathbf{1}_n^T = (1+\delta)\mathbf{1}_n^T$, for $\gamma = \frac{\delta}{1+\delta}$.

Using the expander constructions in [23], 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 $O(\log m)$ tasks per machine. □