Toward Understanding the Improved Robustness to Initialization in Distributed Clustering

1st Aleksandar Armacki
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA, USA
aarmacki@andrew.cmu.edu

2nd Soummya Kar
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA, USA
soummyak@andrew.cmu.edu

***Abstract*—It is well-known that the performance of many clustering algorithms is strongly affected by center initialization. Recently, a number of papers empirically showed that distributed clustering algorithms exhibit improved robustness to center initialization compared to their centralized counterparts. In this paper we provide a theoretical justification for this phenomena, by studying the statistical guarantees of a distributed center initialization method, inspired by the celebrated K-means++ approach. In the presence of K distinct data populations and the client-server setup, where each user contains a chunk of the data and communicates directly with the server, we establish mean-squared error (MSE) guarantees of the proposed initialization scheme, in terms of the gap from the population means. We show under mild assumptions that the probability of a population not being represented in the center initialization decays exponentially in the number of users, implying that the MSE approaches the *optimal error* exponentially fast as the number of users grows. Our result provides the first theoretical explanation for the improved robustness to center initialization exhibited by distributed clustering algorithms in practice.**

***Index Terms*—clustering, distributed, client-server, center initialization, K-means++, generalization, mean-squared error**

I. Introduction

Clustering is a well-studied unsupervised learning problem, where the goal is to group the data into $K \geq 2$ groups based on some similarity criteria, without having knowledge of the true data distributions or the number of groups present [1], [2]. There are many approaches to clustering, including center-based [3], density [4] or spectral clustering [5]. In particular, the center-based clustering algorithms, which include the celebrated K-means algorithm [6], Bregman [7] and gradient based clustering [8], are known to be extremely sensitive to center initialization, with arbitrarily bad performance possible under bad center initialization [9]. To mitigate this issue, many center initialization algorithms have been proposed, among which K-means++ [10] is arguably the most renowned. The idea of K-means++ is to choose centers that are far from each other, thereby increasing the probability of initializing a center from each of the underlying clusters. Formally, for a given dataset $\mathcal{D}$, K-means++ provides a set $\mathcal{C}$ of K initial centers, by first selecting a center from $\mathcal{D}$ uniformly at random, after which the successive centers are selected with probability $\frac{d(x,\mathcal{C})^2}{\sum_{y\in\mathcal{D}} d(y,\mathcal{C})^2}$, for all $x \in \mathcal{D}$, where $d(x,\mathcal{C}) = \min_{c\in\mathcal{C}} \|x-y\|$ is the distance of the point x from the set $\mathcal{C}$. The procedure is summarized in Algorithm 1. Theoretical guarantees of K-means++ have been studied extensively, e.g., [10], [11], with the expected cost of the center initialization produced by K-means++ guaranteed to approximate the optimal clustering cost up to an additive $\log(K)$ factor.

The work of Aleksandar Armacki and Soummya Kar was supported in part by the National Science Foundation, under grant ECCS 2330195.

Distributed learning is a paradigm where multiple users collaborate to train a shared model on the global data, with each user holding a chunk of the data locally. Distributed learning has been widely studied in both client-server [12]–[14] and fully decentralized settings [15]–[17]. While the primary advantage of distributed learning is the reduced computation cost and enhanced privacy guarantees due to each user only having access to a chunk of the data, it was observed recently in the context of clustering that distributed methods show less sensitivity to center initialization [18]–[20]. The authors in [20] hypothesized that the improved resilience stems from the fact that, while centralized algorithms initialize exactly K centers, in distributed algorithms each user initializes their own centers, effectively initializing a total of mK centers, where $m > 2$ is the number of users, decreasing the likelihood of a cluster not being represented in the initialization. However, a theoretical explanation for this phenomena is lacking.

Literature review. We next review the related literature on distributed clustering and guarantees of initialization methods.

a) Distributed clustering: In the context of distributed learning, clustering has been studied in both client-server, e.g., [21]–[24] and decentralized settings, e.g., [18]–[20], [25]. The work [21] designs both client-server and decentralized versions of K-means, providing theoretical guarantees in the client-server setup, in terms of the deviation of the resulting clustering from a clustering produced by running K-means on the full, centralized data. The works [22], [23] study clustering in the presence of outliers, while [24] propose a client-server algorithm based on the spectral K-means clustering method [26], with provable cluster recovery guarantees. The authors in [18] propose decentralized versions of K-means and Expectation Maximization algorithms in the decentralized settings, based on the Alternating Direction Method of Multipliers framework, showing that the centers provably converge to the centers of their centralized counterparts, obtained on

the joint data. The work [20] proposes a general gradient based framework for decentralized clustering, which subsumes popular clustering methods such as K-means and Huber clustering, with the centers provably converging to the set of fixed points of the clustering cost and achieving consensus asymptotically. This is followed up by [19], who consider the same gradient based clustering framework and propose a decentralized version of the K-means++ method, which is shown to improve the performance empirically. The authors in [25] propose decentralized versions of K-means and fuzzy C-means methods, as well as a decentralized version of the K-means++ initialization. The authors in [18] first empirically observed that distributed clustering algorithms show more resilience to center initialization. This was further observed in [19], [20], where the authors hypothesized that the reason for the improved resilience stems from the fact that in the distributed setting, each user initializes their own centers, thus increasing the likelihood of selecting at least one representative from all the underlying clusters.

b) Initialization guarantees: Perhaps the most widely used algorithm for center initialization, K-means++, was originally proposed in [10]. For a finite dataset the authors show that the ratio of the expected cost of the produced center initialization and the optimal clustering cost over the dataset is of the order $\mathcal{O}(1 + \log(K))$. This initial work sparked many extensions, e.g., the authors in [27] proposed a scalable version of K-means++ which draws multiple points in a single pass. The authors in [28] proposed a generalized version, where a total of βK points is sampled, for $\beta \geq 1$, with each point $x \in \mathcal{D}$ sampled with probability $\frac{d(x,\mathcal{C})^l}{\sum_{y\in\mathcal{D}} d(x,\mathcal{C})^l}$, for $l \geq 1$, showing that the ratio of the expected cost of the initialization with βK centers and the optimal cost is of the order $\mathcal{O}(1 + \log(K)/m)$. Finally, the authors in [11] improve the guarantees from [10], [27], showing that performing $\Theta(\log(n))$ passes over the data, where n is the size of $\mathcal{D}$, while drawing $\Theta(K)$ centers in each pass, results in the ratio of the expected and optimal cost of the order $\mathcal{O}(1)$.

Contributions. The aim of this work is to provide a theoretical justification of the improved robustness to center initialization in distributed clustering. To that end, we first propose a distributed center initialization method in the client-server setup, wherein each user initializes K centers locally, by running K-means++ on their dataset and sends them to the server, which produces the final K centers by clustering the initial centers and averaging across clusters. Next, we study the generalization guarantees of the proposed method, in terms of the MSE from the true population means, showing that the resulting MSE approaches the optimal MSE exponentially fast as the number of users increases. This is achieved by showing that the probability of a cluster not being represented by one of the centers decays exponential as the number of users grows, confirming the hypothesis made in [20] on the reasons for improved robustness to center initialization in distributed clustering. While the idea of sampling more than K centers was explored previously in [11], [27], [28] in the context of improving scalability and parallelizing centralized K-means++, our work differs in that the center sampling step is performed independently at each user, without updating their sampling probabilities in relation to other users. While [11], [27], [28] propose to perform a clustering step to obtain the final K centers, their analysis is focused on the guarantees from the first, sampling part of the procedure, whereas we show explicit conditions under which the second, clustering step is guaranteed to perform the correct clustering.

Paper organization. The rest of the paper is organized as follows. Section II defines the problem considered in the paper, Section III outlines the proposed method, Section IV presents the main results, while Section V concludes the paper. The remainder of this section introduces some notation.

Notation. The sets of real numbers and p-dimensional vectors are denoted by $\mathbb{R}$ and $\mathbb{R}^p$, respectively, with the Euclidean norm denoted by $\|\cdot\|$. For a positive integer $m \geq 1$, the set of positive integers up to and including m is denoted by $[m] = \{1, \ldots, m\}$. For a given probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and event $B \in \mathcal{F}$, we use $\mathbb{E}[\cdot|B]$ to denote the expectation conditioned on the σ-algebra induced by B. Unless stated otherwise, we will use subscripts to denote clusters and superscripts to denote users, e.g., c_k^i denotes the center of the k-th cluster at the i-th user.

II. Problem Setup

We consider the problem of clustering data coming from $K \geq 2$ different populations P_k, each with mean $\mu_k \in \mathbb{R}^p$, $k \in [K]$. In a slight abuse of terminology, we will use the terms *population* and *cluster* interchangeably in the reminder of the paper. The data is split among $m \geq 2$ users, communicating in the client-server setup, where users exchanges messages directly with the central server. To facilitate our analysis, we make the following assumptions on the data.

Assumption 1. *Each population has almost surely bounded radius, namely, for all $k \in [K]$ there exist constants $R_k > 0$, such that $\|x - \mu_k\| \leq R_k$, for all $x \sim P_k$, almost surely. Additionally, each population has bounded variance, i.e., $\mathbb{E}_{x\sim P_k}\|x - \mu_k\|^2 \leq \sigma_k^2$, for some $0 < \sigma_k \ll R_k$, $k \in [K]$.*

Assumption 1 requires each population to have an almost surely bounded radius, as well as bounded variance, with the variance much smaller than the cluster radius. This corresponds to setting where most samples are concentrated around the population mean, with some potentially large outliers. Denote the maximum radius by $R = \max_{k\in[K]} R_k$. Next, we make the following assumption on population separation.

Assumption 2. *The population means are sufficiently separated, i.e., $\min_{k\neq l} \|\mu_k - \mu_l\| \geq r$, for some $r > R\sqrt{6}$.*

Assumption 2 requires the population means to be at least $R\sqrt{6}$ apart. Using the triangle inequality, it follows that

$$\min_{x\sim P_k, y\sim P_l, k\neq l} \|x - y\| \geq r - 2R > 0,$$

i.e., the populations do not overlap, almost surely. Denote the maximum population mean distance by $D =$

Algorithm 1 K-means++

Require: Number of centers K;
1: Choose a sample $x \in \mathcal{D}$ uniformly at random;
2: Initialize the set of centers $\mathcal{C} = \{c_1\}$, where $c_1 = x$;
3: **for** t = 2,...,K iterations **do**
4: Choose sample $x \in \mathcal{D}$ with probability $\frac{d(x,\mathcal{C})^2}{\sum_{y\in\mathcal{D}} d(y,\mathcal{C})^2}$;
5: Update the set of centers $\mathcal{C} = \mathcal{C} \cup \{c_t\}$, where $c_t = x$;
6: **end for**
7: **return** Center initialization $\mathcal{C} = \{c_1, \ldots, c_K\}$;

$\max_{k,l\in[K]} \|\mu_k - \mu_l\|$. The following assumption specifies the distribution of data across users.

Assumption 3. *Each user has access to a total of $n \geq K$ independent, identically distributed (IID) samples from each population, for a total of nK samples per user.*

Assumption 3 states that the data across users is independent and distributed in a homogeneous manner, in the sense that each user has access to data from all of the K underlying populations. Data independence is widely used in statistical learning and signal processing literature, e.g., [29]–[31].

III. Proposed Method

In this section we describe the proposed initalization method. We consider a method built on the idea of K-means++ [10], consisting of a two-step procedure. In the first step, each user initializes K centers by running K-means++ on the local data in isolation and sends the centers to the server. In the second step, the sever runs a clustering algorithm Ξ on the mK centers received from the users, after which the final K centers are obtained as the cluster averages and sent back to the users. The method is summarized in Algorithm 2. We now define the class of *admissible* clustering algorithms used in the second step of Algorithm 2.

Definition 1. *We say that a dataset $\{a_i\}_{i\in[n]}$ is* α-separable *with respect to a clustering $\{A_k\}_{k\in[K]}$, if for some $\alpha > 1$*

$$\alpha \max_{i\in A_k, k\in[K]} \|\mu_k - a_i\| < \min_{l\neq k} \|\mu_k - \mu_l\|,$$

where $\mu_k = \frac{1}{|A_k|}\sum_{i\in A_k} a_i$ is the mean of cluster A_k, $k \in [K]$.

Definition 2. *A clustering algorithm Ξ is* α-admissible*, if there exists an $\alpha > 1$, such that for any dataset $\{a_i\}_{i\in[n]}$ which is α'-separable with respect to a clustering $\{A_k\}_{k\in[K]}$, for any $\alpha' \geq \alpha$, there exist hyperparameters η, for which $\Xi(\eta)$ exactly recovers the clustering $\{A_k\}_{k\in[K]}$.*

Definition 1 provides a separability condition with respect to a clustering of the data, requiring the largest cluster radius to be smaller than the smallest distance between two cluster means, by a factor of $\alpha > 1$. Definition 2 states that an algorithm is α-admissible if, given a α-separable dataset, the algorithm is guaranteed to recover the associated clustering. Definitions 1 and 2 were originally introduced in [29], where it was shown that algorithms such as spectral K-means [26], [32] and convex clustering [33] are α-admissible, for α sufficiently large. From Definition 2, the parameter α can be seen as a measure of efficiency of a clustering algorithm, with better algorithms requiring smaller α to guarantee exact clustering.

Algorithm 2 Distributed K-Means++ with clustering

Require: Number of centers K, clustering algorithm Ξ;
1: **for** all users $i \in [m]$ in parallel **do**
2: Produce initial centers $\mathcal{C}^i = \{c_k^i\}_{k\in[K]}$, by running Algorithm 1 on the local dataset $\mathcal{D}^i$;
3: Send the centers $\mathcal{C}^i = \{c_k^i\}_{k\in[K]}$ to the server;
4: **end for**
5: **Server:** (i) Choose hyperparameters η and run the clustering algorithm $\Xi(\eta)$ to cluster the centers $\cup_{i\in[m]}\mathcal{C}^i$ into K clusters $\{C_k\}_{k\in[K]}$;
(ii) Produce the final centers $\widetilde{c}_k = \frac{1}{|C_k|}\sum_{y\in C_k} y$, for all $k \in [K]$ and send them to each user $i \in [m]$;

IV. Theoretical Analysis

In this section we analyze the performance of Algorithm 2. We start by defining the notion of a *covered* cluster.

Definition 3. *A cluster $k \in [K]$ is* covered *by center initialization $\mathcal{C}$, if $\mathcal{C}$ contains a sample belonging to population k.*

Next, we provide an important intermediate lemma on the guarantees of K-means++ initialization.

Lemma 1. *Let Assumptions 1 and 2 hold and let centers be initialized by running Algorithm 1 on a dataset $\mathcal{D}$, containing n IID samples from each population. Then, for any $k \in [K]$*

$$\mathbb{P}(\textit{cluster } k \textit{ not covered}) \leq \left(1 - \frac{r^2 - 6R^2}{4K(D+2R)^2}\right)^K$$

Proof. Let $\mathcal{D} = \cup_{k\in[K]}\mathcal{D}_k$, where $\mathcal{D}_k = \{x \in \mathcal{D} : x \text{ is drawn from the } k\text{-th population}\}$. For ease of notation, let $B_k = \{\omega : \text{cluster } k \text{ not covered}\}$. Noting that Algorithm 1 draws K samples and using Bayes' rule, it follows that

$$\mathbb{P}(B_k) = \mathbb{P}(B_{k,1})\mathbb{P}(B_{k,2}|B_{k,1})\ldots\mathbb{P}(B_{k,K}|B_{k,1}\ldots B_{k,K-1}),$$

where $B_{k,t} = \{\omega : \text{cluster } k \text{ not covered in } t\text{-th draw}\}$, $t \in [K]$. From Algorithm 1, it is easy to see that $\mathbb{P}(B_{k,1}) = 1 - \frac{n}{nK} = 1 - \frac{1}{K}$. Next, let $\mathcal{C}_t$ denote the set of centers chosen after t iterations, for any $t \geq 2$. The K-means++ weighted selection rule then tells us that, for any $t \geq 2$

$$\mathbb{P}(B_{k,t}^c|B_{k,1},\ldots,B_{k,t-1}) = \sum_{x\in\mathcal{D}_k} \frac{d(x,\mathcal{C}_{t-1})^2}{\sum_{y\in\mathcal{D}} d(y,\mathcal{C}_{t-1})^2},$$

where $B_{k,t}^c = \{\omega : \text{cluster } k \text{ covered in the } t\text{-th draw}\}$ is the complement of $B_{k,t}$. Notice that, conditioned on $\cap_{s=1}^{t-1}B_{k,s}$, the set $\mathcal{C}_{t-1}$ contains no points from $\mathcal{D}_k$. Denote by $\mu_{(y)}$ the population mean corresponding to the true population of any point $y \in \mathcal{D}$, i.e., if $y \in \mathcal{D}_k$, then $\mu_{(y)} = \mu_k$. Using the triangle

inequality and $(a+b)^2 \leq 2a^2 + 2b^2$, we then have, for any $c \in \mathcal{C}_{t-1}$

$$\begin{aligned}\|x-c\|^2 &\geq \frac{1}{4}\left(\|\mu_k - \mu_{(c)}\|^2 - 2(\|\mu_k - x\|^2 + 2\|\mu_{(c)} - c\|^2)\right) \\ &\geq \frac{1}{4}(r^2 - 6R^2) > 0,\end{aligned}$$

where the last inequality follows from Assumption 2. Next, we want to quantify $d(y, \mathcal{C}_{t-1})$, for any $y \in \mathcal{D}$. To that end, consider two cases. If y belongs to one of the clusters that is covered by $\mathcal{C}_{t-1}$, from Assumptions 1 and 2 it follows that $d(y, \mathcal{C}_{t-1}) \leq 2R$. On the other hand, if y belongs to a cluster that is not covered by $\mathcal{C}_{t-1}$, we then have, for any $c \in \mathcal{C}_{t-1}$

$$\|y - c\| \leq \|y - \mu_{(y)}\| + \|\mu_{(y)} - \mu_{(c)}\| + \|\mu_{(c)} - c\| \leq 2R + D.$$

Combining the two cases, we get $d(y, C_{t-1})^2 \leq (D+2R)^2$, for any $y \in \mathcal{D}$. Therefore, we have, for any $t \geq 2$

$$\begin{aligned}\mathbb{P}(B^c_{k,t} | B_{k,1}, \ldots, B_{k,t-1}) &\geq \sum_{x \in \mathcal{D}_k} \frac{r^2 - 6R^2}{4\sum_{y \in \mathcal{D}}(D+2R)^2} \\ &= \frac{r^2 - 6R^2}{4K(D+2R)^2},\end{aligned}$$

where the last equality follows from the fact that $|\mathcal{D}_k| = n$, for all $k \in [K]$. Equivalently, for any $t \geq 2$, we have

$$\mathbb{P}(B_{k,t} | B_{k,1}, \ldots, B_{k,t-1}) \leq 1 - \frac{r^2 - 6R^2}{4K(D+2R)^2},$$

which, combined with the fact that $\frac{r^2-6R^2}{4(D+2R)^2} < 1$, from definitions of r and D, implies that

$$\begin{aligned}\mathbb{P}(B_k) &= \mathbb{P}(B_{k,1}) \prod_{t=2}^{K} \mathbb{P}(B_{k,t} | B_{k,1} \ldots B_{k,t-1}) \\ &\leq \left(1 - \frac{r^2 - 6R^2}{4K(D+2R)^2}\right)^K.\end{aligned}$$

□

Lemma 1 provides an upper bound on the probability of a cluster not being covered by Algorithm 1, in terms of the number of clusters K, maximum cluster radius R and minimum and maximum mean separation r and R. Note that the resulting probability can be arbitrarily close to 1, as the number of underlying clusters K increases. As we show next, this is mitigated in the distributed setting.

Theorem 1. *Let Assumptions 1-3 hold and let $\{\widetilde{c}_k\}_{k \in [K]}$ be the centers obtained by running Algorithm 2 using an α-admissible clustering algorithm Ξ. If $r \geq 2(\alpha+1)R$, then we have, for all $k \in [K]$*

$$\begin{aligned}\mathbb{E}[\min_{j \in [K]} \|\widetilde{c}_j - \mu_k\|^2] \leq \sigma_k^2 + 2K(R^2 + D^2) \times \\ \times \left(1 - \frac{r^2 - 6R^2}{4K(D+2R)^2}\right)^{mK}.\end{aligned}$$

Prior to proving Theorem 1, we provide some remarks.

Remark 1. *Theorem 1 states that for each population $k \in [K]$, Algorithm 2 will produce a center initialization which will be at most $\mathcal{O}(\sigma_k^2 + [1 - {}^{(r^2-6R^2)}/_{4K(D+2R)^2}]^{mK})$ apart from the population mean μ_k, in the MSE sense. Noting that σ_k^2 is the* optimal MSE error *for any single point, it follows that the center initialization approaches the optimal error exponentially fast, as the number of users $m \to \infty$.*

Remark 2. *Note that the dependence on σ_k^2 is worst case, in the sense that some of the clusters produced in the second step in Algorithm 2 will contain more than a single point. As such, a variance reduced effect can be achieved, as the dependence on σ_k^2 can be reduced to $\frac{\sigma_k^2}{|C_k|}$. However, in the analysis used in the proof of Theorem 1, we do not know for which $k \in [K]$ we have $|C_k| > 1$, therefore, we present the worst case dependence on σ_k^2 in Theorem 1.*

Remark 3. *As was discussed in Section III, the value of α is an inherent property of the clustering algorithm used in the second step in Algorithm 2. The authors in [29] show that, if the spectral K-means method [26] is used, then $\alpha = \Theta(\sqrt{mK})$, while if the convex clustering algorithm [33] is used, then $\alpha = \Theta(mK)$. While on first inspection this might seem like the second term on the right-hand side in Theorem 1 becomes vacuous, this is not the case, as $D \geq r$, by definition. If $D = \Theta(\sqrt{m})$ and the spectral K-means algorithm is used as a sub-routine in Algorithm 2, this implies that the second term on the right-hand side in Theorem 1 is of the order $\mathcal{O}\left(m\left(1 - \frac{m}{m+C}\right)^m\right)$, which still decays to zero as $m \to \infty$. As discussed, this dependence stems from the clustering algorithm used in the second part of Algorithm 2.*

Proof of Theorem 1. Let $\widetilde{\mathcal{C}} = \{\widetilde{c}_1, \ldots, \widetilde{c}_K\}$ denote the center initialization produced by Algorithm 2 and define $B = \{\omega : \text{all clusters are covered}\}$. Noting that $\mathbb{E}[\min_{j \in [K]} \|\widetilde{c}_j - \mu_k\|^2] = \mathbb{E}[d(\mu_k, \widetilde{\mathcal{C}})^2]$, we then have

$$\begin{aligned}\mathbb{E}[d(\mu_k, \widetilde{\mathcal{C}})^2] &= \mathbb{E}[d(\mu_k, \widetilde{\mathcal{C}})^2 | B] + \mathbb{E}[d(\mu_k, \widetilde{\mathcal{C}})^2 | B^c] \\ &\leq \mathbb{E}[d(\mu_k, \widetilde{\mathcal{C}})^2 | B] + 2(R^2 + D^2)\mathbb{P}(B^c),\end{aligned}$$

for any $k \in [K]$, where the inequality follows from Assumption 1 and the fact that, if cluster k is not covered, then $d(\mu_k, \widetilde{\mathcal{C}})^2 \leq 2(R^2 + D^2)$. By definition of B, to guarantee that $\mathbb{E}[d(\mu_k, \widetilde{\mathcal{C}})^2 | B] \leq \sigma_k^2$, we need to show that the clustering step finds the true clusters. Let $\mathcal{C}^i = \{c_k^i\}_{k \in [K]}$ denote the center initialization of user $i \in [m]$ and $A_k = \{c_k^i, k \in [K], i \in [m] : c_k^i \text{ comes from the } k\text{-th population}\}$, $b_k = \frac{1}{|A_k|}\sum_{a \in A_k} a$, $k \in [K]$, be the desired clusters and centers, respectively. Then, for any $k \neq l$

$$\begin{aligned}\|b_k - b_l\| &\geq \|\mu_k - \mu_l\| - \|\mu_k - b_k\| - \|\mu_l - b_l\| \\ &\geq r - 2R \geq 2\alpha R,\end{aligned} \tag{1}$$

almost surely, where the last inequality follows from $r \geq 2(\alpha+1)R$. Similarly, conditioned on B, for any $a \in A_k$

$$\|a - b_k\| \leq \|a - \mu_k\| + \|b_k - \mu_k\| \leq 2R, \tag{2}$$

almost surely, since all the clusters are covered on the event B, i.e., $|A_k| \geq 1$, conditioned on B. Combining (1) and (2), it readily follows that $\cup_{i\in[m]}\mathcal{C}^i$ is α-admissible with respect to $\{A_k\}_{k\in[K]}$, ensuring that the true clustering will be recovered by Ξ on B, i.e., that $\widetilde{c}_k = b_k = \frac{1}{|A_k|}\sum_{a\in A_k} a$, which in turns implies $\mathbb{E}[d(\mu_k, \widetilde{C})^2|B] \leq \sigma_k^2$, for all $k \in [K]$, as desired. Next, noting that $B^c = \cup_{k\in[K]} \cap_{i\in[m]} B_k^i$, where $B_k^i = \{\omega : k\text{-th cluster not covered at user } i\}$, it follows that

$$\mathbb{P}(B^c) \leq \sum_{k\in[K]} \mathbb{P}(\cap_{i\in[m]} B_k^i) = \sum_{k\in[K]} \mathbb{P}(B_k^1)^m,$$

where the last equality stems from the fact that the data across users is IID and users run K-means++ on the local data independent from one another. The claim now follows by applying Lemma 1 on $\mathbb{P}(B_k^1)$, for all $k \in [K]$. □

V. Conclusion

In this paper we provide a theoretical explanation for the recently observed phenomena of improved robustness to center initialization in distributed clustering. To that end, we proposed and analyzed the guarantees of a distributed version of the K-means++ initialization scheme in the client-server setup, assuming users have access to IID data drawn from K different populations. We show under mild assumptions that the center initialization produced by our method is guaranteed to achieve a MSE with respect to true population means which approaches the optimal error exponentially fast as the number of users grows. This is achieved by showing that the probability of a population not being covered by our initialization scheme decays exponentially as the number of users grows, aligning well with prior hypotheses and helping to explain why distributed clustering algorithms show less sensitivity to center initialization than their centralized counterparts.

References

[1] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, pp. 651–666, 2010, award winning papers from the 19th International Conference on Pattern Recognition (ICPR).

[2] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[3] P. Awasthi and M. F. Balcan, "Foundations for center-based clustering: worst-case approximations and modern developments," in *Handbook of cluster analysis*, 1st ed. Chapman and Hall/CRC, 2016, pp. 67–100.

[4] A. Beer, A. Draganov, E. Hohma, P. Jahn, C. M. Frey, and I. Assent, "Connecting the dots - density-connectivity distance unifies dbscan, k-center and spectral clustering," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2023, p. 80–92.

[5] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[6] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[7] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with bregman divergences," *JMLR*, vol. 6, no. 58, pp. 1705–1749, 2005.

[8] A. Armacki, D. Bajovic, D. Jakovetic, and S. Kar, "Gradient based clustering," in *39th International Conference on Machine Learning*, vol. 162. PMLR, 2022, pp. 929–947.

[9] G. W. Milligan, "An examination of the effect of six types of error perturbation on fifteen clustering algorithms," *Psychometrika*, vol. 45, no. 3, pp. 325–342, Sep 1980.

[10] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *In Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*. New Orleans: SIAM, 2007, p. 1027–1035.

[11] K. Makarychev, A. Reddy, and L. Shan, "Improved guarantees for k-means++ and k-means++ parallel," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020.

[12] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv 1610.05492*, 2017.

[13] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *20th International Conference on Artificial Intelligence and Statistics*, vol. 54. PMLR, 2017, pp. 1273–1282.

[14] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[15] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, 2012.

[16] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014.

[17] S. Vlaski, S. Kar, A. H. Sayed, and J. M. Moura, "Networked signal and information processing: Learning by multiagent systems," *IEEE Signal Processing Magazine*, vol. 40, no. 5, pp. 92–105, 2023.

[18] P. A. Forero, A. Cano, and G. B. Giannakis, "Distributed clustering using wireless sensor networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 707–724, 2011.

[19] A. Armack, H. Sharma, D. Bajović, D. Jakovetić, M. Chakraborty, and S. Kar, "Distributed gradient clustering: Convergence and effects of initialization," in *58th Asilomar Conference on Signals, Systems, and Computers [To appear]*, 2024. [Online]. Available: https://github.com/aarmacki/aarmacki.github.io/blob/master/publications/Asilomar_2024.pdf

[20] A. Armacki, D. Bajović, D. Jakovetić, and S. Kar, "Distributed center-based clustering: A unified framework," *IEEE Transactions on Signal Processing*, vol. 73, pp. 903–918, 2025.

[21] S. Datta, C. Giannella, and H. Kargupta, "Approximate distributed k-means clustering over a peer-to-peer network," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 10, pp. 1372–1388, 2009.

[22] S. Li and X. Guo, "Distributed k-clustering for data with heavy noise," in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018.

[23] J. Huang, Q. Feng, Z. Huang, J. Xu, and J. Wang, "Fast algorithms for distributed k-clustering with outliers," in *40th International Conference on Machine Learning*, vol. 202. PMLR, 2023, pp. 13 845–13 868.

[24] D. K. Dennis, T. Li, and V. Smith, "Heterogeneity for the win: One-shot federated clustering," in *38th International Conference on Machine Learning*, vol. 139. PMLR, 2021, pp. 2611–2620.

[25] J. Qin, W. Fu, H. Gao, and W. X. Zheng, "Distributed k -means algorithm and fuzzy c -means algorithm for sensor networks based on multiagent consensus theory," *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 772–783, 2017.

[26] P. Awasthi and O. Sheffet, "Improved spectral-norm bounds for clustering," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Berlin: Springer, 2012, pp. 37–49.

[27] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii, "Scalable k-means++," *Proceedings of the VLDB Endowment*, 2012.

[28] D. Wei, "A constant-factor bi-criteria approximation guarantee for k-means++," in *Advances in Neural Information Processing Systems*, vol. 29. Curran Associates, Inc., 2016.

[29] A. Armacki, D. Bajović, D. Jakovetić, and S. Kar, "A one-shot framework for distributed clustered learning in heterogeneous environments," *IEEE Transactions on Signal Processing*, vol. 72, pp. 636–651, 2024.

[30] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *IEEE Transactions on Information Theory*, pp. 1–1, 2022.

[31] Y. Zhang, J. C. Duchi, and M. J. Wainwright, "Communication-efficient algorithms for statistical optimization," *Journal of Machine Learning Research*, vol. 14, no. 104, pp. 3321–3363, 2013.

[32] A. Kumar and R. Kannan, "Clustering with spectral norm and the k-means algorithm," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, 2010, pp. 299–308.

[33] D. Sun, K.-C. Toh, and Y. Yuan, "Convex clustering: Model, theoretical guarantee and efficient algorithm," *JMLR*, vol. 22, no. 1, 2021.