Communication-efficient $k$-Means for Edge-based Machine Learning

Hanlin Lu*, Ting He*, Shiqiang Wang†, Changchang Liu‡, Mehrdad Mahdavi*, Vijaykrishnan Narayanan*, Kevin S. Chan†, and Stephen Pasteris§

*Pennsylvania State University, University Park, PA, USA. Email: {hzl263, tzh58, mzm616, vxn9}@psu.edu
†IBM T. J. Watson Research Center, Yorktown, NY, USA. Email: {wangshiq@us., Changchang.Liu33}@ibm.com
‡Army Research Laboratory, Adelphi, MD, USA. Email: kevin.s.chan.civ@mail.mil
§University College London, London, UK. Email: s.pasteris@cs.ucl.ac.uk

Abstract—We consider the problem of computing the $k$-means centers for a large high-dimensional dataset in the context of edge-based machine learning, where data sources offload machine learning computation to nearby edge servers. $k$-Means computation is fundamental to many data analytics, and the capability of computing provably accurate $k$-means centers by leveraging the computation power of the edge servers, at a low communication and computation cost to the data sources, will greatly improve the performance of these analytics. We propose to let the data sources send small summaries, generated by joint dimensionality reduction (DR) and cardinality reduction (CR), to support approximate $k$-means computation at reduced complexity and communication cost. By analyzing the complexity, the communication cost, and the approximation error of $k$-means algorithms based on state-of-the-art DR/CR methods, we show that: (i) in the single-source case, it is possible to achieve a near-optimal approximation at a near-linear complexity and a constant communication cost, (ii) in the multiple-source case, it is possible to achieve similar performance at a logarithmic complexity and a constant communication cost, and (iii) the order of applying DR and CR significantly affects the complexity and the communication cost. Our findings are validated through experiments based on real datasets.

Index Terms—Coreset, dimensionality reduction, random projection, $k$-means, edge-based machine learning.

I. INTRODUCTION

Given a dataset $P \subset \mathbb{R}^d$ with cardinality $n$, where both $n \gg 1$ and $d \gg 1$, consider the problem of finding $k$ points $X = \{x_i\}_{i=1}^k$ to minimize the following cost function:

$$\text{cost}(P, X) := \sum_{p \in P} \min_{x_i \in X} \|p - x_i\|^2. \quad (1)$$

This is the $k$-means clustering problem, and the points in $X$ are called centers. Equivalently, the $k$-means clustering problem can be considered as the problem of finding the partition $\mathcal{P} = \{P_1, \ldots, P_k\}$ of $P$ into $k$ clusters that minimizes the following cost function:

$$\text{cost}(\mathcal{P}) := \sum_{i=1}^k \min_{x_i \in \mathbb{R}^d} \sum_{p \in P_i} \|p - x_i\|^2. \quad (2)$$

$k$-Means clustering is one of the most widely-used machine learning techniques. Algorithms for $k$-means are used in many areas of data science, e.g., for data compression, quantization, hashing; see the survey in [1] for more details. Recently, it was shown in [2], [3] that the centers of $k$-means can be used as a proxy (called a coreset) of the original dataset in computing a broader set of machine learning models with sufficiently continuous cost functions. Thus, efficient and accurate computation of $k$-means can bring broad benefits to machine learning applications. However, solving $k$-means is non-trivial. The problem is known to be NP-hard, even for two centers [4] or in the plane [5]. Due to its fundamental importance, how to speed up the $k$-means computation for large datasets has received significant attention. Existing solutions can be classified into two approaches: dimensionality reduction (DR) techniques that aim at running $k$-means on a “thinner” dataset with a reduced dimension [6], and cardinality reduction (CR) techniques that aim at running $k$-means on a “smaller” dataset with a reduced number of points [7]. However, these solutions only considered the computation cost, while implicitly assuming that the data is at the same location where the $k$-means computation is performed.

To our knowledge, we are the first to explicitly analyze the communication cost in computing $k$-means over remote (and possibly distributed) data. The need of communications arises in the emerging application scenario of edge-based machine learning [8], where mobile/wireless devices collect the raw data and transmit them to nearby edge servers for processing. Compared to alternative approaches, e.g., transmitting the locally learned model parameters as in federated learning [9], transmitting data (summaries) has the advantage that: (i) only one round of communications is required, (ii) the transmitted data can potentially be used to compute other machine learning models [2], [3], and (iii) the edge server can solve the machine learning problem closer to the optimality more easily than the data-collecting devices (within the same time).

In this work, we consider the problem of solving $k$-means for a large high-dimensional dataset, i.e., $n, d \gg 1$ ($n$: car-

\[\text{In cases that the raw data are spread over multiple nodes, another round of communications is needed to decide the sizes of data summaries to collect from each node [10]. However, each node only sends one scalar in this round and hence the communication cost is negligible.}\]
A. Related Work

Our work belongs to the studies on data reduction for approximate $k$-means. Existing solutions are classified into (i) dimensionality reduction and (ii) cardinality reduction.

Dimensionality reduction (DR) for $k$-means, initiated by [11], aims at speeding up $k$-means by reducing the number of features (i.e., the dimension). Two approaches have been proposed: 1) feature selection that selects a subset of the original features, and 2) feature extraction that constructs a smaller set of new features. For feature selection, the best known algorithm in [12] achieves a $(1+\epsilon)$-approximation using a random sampling algorithm. For feature extraction, there are two methods with guaranteed approximation, based on singular value decomposition (SVD) [12] and random projections [6].

Cardinality reduction (CR) for $k$-means, initiated by [13], aims at using a small weighted set of points in the same space, referred to as a coreset, to replace the original dataset. A coreset is called an $\epsilon$-coreset (for $k$-means) if it can approximate the $k$-means cost of the original dataset for every candidate set of centers up to a factor of $1 \pm \epsilon$. Many coreset construction algorithms have been proposed for $k$-means. Most state-of-the-art coreset construction algorithms are based on the sensitivity sampling framework [14], which needs a coreset cardinality of $O(kd\epsilon^{-4})$. The best known solution is the one in [7, Theorem 36], which showed that the cardinality of an $\epsilon$-coreset can be reduced to $O(kd^2\epsilon^{-4})$. Note that [7] is the full version of [15], and thus we will refer to [7] instead of [15].

In the distributed setting, [10] proposed a distributed version of sensitivity sampling to construct an $\epsilon$-coreset over a distributed dataset, and [16] further combined this algorithm with a distributed PCA algorithm from [7]. Besides these theoretical results, there are also works on adapting centralized $k$-means algorithms for distributed settings, e.g., MapReduce [17], sensor networks [18], and Peer-to-Peer networks [19]. However, these algorithms are only heuristics, while we focus on algorithms with guaranteed approximation errors.

Only [7], [16] considered joint DR and CR for $k$-means. However, they blindly assumed that DR should be applied before CR, leaving open several important questions: 1) Is it possible to achieve the same approximation error at a lower complexity or communication cost? 2) Does the order of applying DR and CR matter? 3) Will repeated DR and CR help? We will address all these questions.

B. Summary of Contributions

Our contribution is three-fold:

1) If the data reside at a single data source, we show that (i) it is possible to solve $k$-means arbitrarily close to the optimal with constant communication cost and near-linear complexity at the data source, (ii) the order of applying DR and CR methods will not affect the approximation error, but will lead to different tradeoffs between communication cost and complexity, and (iii) repeating DR both before and after CR can further improve the performance.

2) If the data are distributed over multiple data sources, we show that it is possible to solve $k$-means arbitrarily close to the optimal with near-linear complexity at the data sources and a total communication cost that is logarithmic in the data size, achieved by combining a data-oblivious DR method with a state-of-the-art CR method in a particular order.

3) Through experiments on real datasets, we verify that (i) joint DR and CR can effectively reduce the communication cost without incurring a high complexity at the data sources or significantly degrading the solution quality, and (ii) the proposed algorithms can achieve a solution quality similar to state-of-the-art data reduction algorithms while significantly reducing the communication cost and the complexity.

Roadmap. Section II reviews the background on DR/CR methods. Section III presents our results for the single-source case. Section IV addresses extensions including multiple data sources and repeated DR/CR methods. Section V presents our experimental results. Section VI concludes this paper.

II. BACKGROUND AND FORMULATION

We start with a brief overview of the state-of-the-art results on DR and CR in support of $k$-means computation.

A. Notations

We will use $||x||$ to denote the $\ell$-2 norm if $x$ is a vector, or the Frobenius norm if $x$ is a matrix. We will use $A_P \in \mathbb{R}^{n \times d}$ to denote the matrix representation of a dataset $P \subset \mathbb{R}^d$, where each row corresponds to a data point. Let $\mu(P)$ denote the optimal 1-means center of $P$, which is well-known to be the sample mean, i.e., $\mu(P) = \frac{1}{|P|} \sum_{p \in P} p$. Let $\mathcal{P}_{P,X}$ denote the partition of dataset $P$ induced by centers $X$, i.e., $\mathcal{P}_{P,X} = \{P_1,\ldots,P_{|X|}\}$ for $P_i := \{p \in P : ||p-x_i|| \leq ||p-x_j||, \forall x_j \in X \setminus \{x_i\}\}$ (ties broken arbitrarily). Given scalars $x$, $y$, and $\epsilon$ ($\epsilon > 0$), we will use $x \approx_{1+\epsilon} y$ to denote $\frac{1}{1+\epsilon} x \leq y \leq (1+\epsilon)x$. In our analysis, we use $O(x)$ to denote a value that is at most linear in $x$, $\Omega(x)$ to denote a value that is at least linear in $x$, and $\tilde{O}(x)$ to denote a value that is at most linear in $x$ times a factor that is polylogarithmic in $x$.

Given a dimensionality reduction map $\pi : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ ($d' < d$), we use $\pi(P) := \{\pi(p) : p \in P\}$ to denote the output dataset for an input dataset $P$, and $\pi(P) := \{\pi(P_1),\ldots,\pi(P_k)\}$ to denote the partition of $\pi(P)$ corresponding to a partition $P = \{P_1,\ldots,P_k\}$ of $P$. Moreover, given a partition $P' = \pi(P)$, we use $\pi^{-1}(P')$ to denote the corresponding partition of $P$, which puts $p,q \in P$ into the same cluster if and only if $\pi(p),\pi(q) \in P'$ belong to the

3We use $\tilde{O}(x)$ to denote a value that is at most linear in $x$ times a factor that is polylogarithmic in $x$. 
same cluster under $\mathcal{P}'$. Finally, given $P' = \pi(P)$, we use $\pi^{-1}(P') := \{\pi^{-1}(p') : p' \in P'\}$ to denote a set of points in $\mathbb{R}^d$ that is mapped to $P'$ by $\pi$. Note that there is no guarantee that $\pi^{-1}(P') = P$. However, solutions to $\pi(P) = P$ must exist ($P$ is a feasible solution) and $\pi^{-1}(P')$ denotes an arbitrary solution. If $\pi$ is a linear map, i.e., $\pi(P) := A_P\Pi$ for a matrix $\Pi \in \mathbb{R}^{d \times d'}$, then the Moore-Penrose inverse $\Pi^+$ [20] of $\Pi$ gives a feasible solution $\pi^{-1}(P') := A_{P'}\Pi^+$. 

B. Dimensionality Reduction for $k$-Means

**Definition II.1.** We say that a DR map $\pi : \mathbb{R}^d \rightarrow \mathbb{R}^{d'} (d' < d)$ is an $\epsilon$-projection if it preserves the cost of any partition up to a factor of $1 + \epsilon$, i.e., $\text{cost}(P) \approx_{1+\epsilon} \text{cost}(\pi(P))$ for every partition $P = \{P_1, \ldots, P_k\}$ of a finite set $P \subset \mathbb{R}^d$.

One commonly used method to construct $\epsilon$-projection is random projection, where the cornerstone result is the JL theorem II.2.

**Theorem II.2** ([7]). For any $\epsilon, \delta \in (0, 1)$, with probability at least $1 - \delta$, an $\epsilon$-coreset $(S, \Delta, w)$ of size $|S| = O\left(\frac{k^{3/2} \log^2 k}{\epsilon^4 \log \frac{1}{\delta}}\right)$ can be computed in time $O\left(\min(n(d^2, n^2d) + nke^{-2}(d + k\log(1/\delta)))\right)$.

However, [7] only focused on minimizing the cardinality of coreset, ignoring the cost of transmitting the coreset. As is shown later (Section III-C), its proposed algorithm can be severely suboptimal in the communication cost.

D. Problem Statement

The motivation of most existing DR/CR methods designed for $k$-means is to speed up $k$-means computation in a setting where the node holding the data is also the node computing $k$-means. In contrast, we want to develop efficient $k$-means algorithms in scenarios where the data generation and the $k$-means computation occur at different locations, such as in the case of edge-based learning. We will refer to the node(s) holding the original data as the data source(s), and the node running $k$-means computation as the server.

We will perform a holistic performance analysis for all the considered algorithms by the following metrics:

- **Approximation error:** We say that a set of $k$-means centers $X$ is an $\alpha$-approximation ($\alpha > 1$) for $k$-means clustering of $P$ if $\text{cost}(P, X) \leq \alpha \cdot \text{cost}(P, X^*)$, where $X^*$ is the optimal set of $k$-means centers for $P$.

- **Communication cost:** We say that an algorithm incurs a communication cost of $y$ if a data source employing the algorithm needs to send $y$ scalars to the server.

- **Complexity:** We say that an algorithm incurs a complexity of $z$ at the data source if a data source employing the algorithm needs to perform $z$ elementary operations.

III. Joint DR and CR for $k$-Means

We will first focus on the scenario where all the data are at a single data source (the centralized setting), and leave the scenario where the data are split among multiple data sources (the distributed setting) to Section IV-A.

In this section, we will show that: 1) treating given DR/CR methods as black boxes, fixed qualities of these methods will yield a fixed quality of the $k$-means solution computed from the reduced dataset, regardless of the order of applying these
methods; 2) using suitably selected DR/CR methods and a sufficiently powerful server, it is possible to solve $k$-means arbitrarily close to the optimal, while incurring a constant communication cost and a near-linear complexity at the data source; 3) applying CR first and applying DR first lead to a tradeoff between the communication cost and the complexity.

### A. DR+CR

We first consider the approach of applying DR and then CR.

1) **A Black-box Approach**: Given DR/CR methods as black boxes, our analysis needs the following lemma to show the relationship between Equation (1) and Equation (2).

#### Lemma III.1

The two cost functions defined in (1) and (2) are related by:

1. For $\mathcal{P} = \{P_i\}_{i=1}^k$, cost($\mathcal{P}$) $\geq$ cost($P, X$) if $X = \{\mu(P_i)\}_{i=1}^k$;
2. cost($P, X$) $\geq$ cost($\mathcal{P}_{P, X}$).

**Proof.** For 1), since $x_i := \mu(P_i)$ is the optimal center of $P_i$,

$\text{cost}(\mathcal{P}) = \sum_{i=1}^k \sum_{p \in P_i} \|p - x_i\|^2 = \sum_{p \in P} \|p - x_i\|^2 \geq \sum_{p \in P} \min_{x_i \in X} \|p - x_i\|^2 = \text{cost}(P, X),$

where $i_p$ in (5) is the index of the cluster $P_i$ such that $p \in P_i$.

For 2), by definition of $\mathcal{P}_{P, X} = (P_1, \ldots, P_k)$ ($k := |X|$),

$\text{cost}(P, X) = \sum_{i=1}^k \sum_{p \in P_i} \|p - x_i\|^2 \geq \sum_{i=1}^k \min_{p \in P} \sum_{p \in P} \|p - q\|^2 = \text{cost}(\mathcal{P}_{P, X}),$

where (6) is by the definition in (2).

Next we show the approximation error of first applying DR method $\pi_1$, followed by CR method $\pi_2$ is bounded as follows.

#### Theorem III.1

In Algorithm 1, let $X'$ be the optimal $k$-means centers of $S' := \pi_2(\pi_1(P))$, where $\pi_1$ is an $\epsilon$-projection and $\pi_2$ generates an $\epsilon$-creset for some $\epsilon \in (0, 1)$. Then $X := \{\mu(P_i)\}_{i=1}^k$, where $\{P_1, \ldots, P_k\} = \pi_1^{-1}(\pi_1(P, X^*))$, is a $(1 + \epsilon)^3/(1 - \epsilon)$-approximation for $k$-means clustering of $P$.

**Proof.** Let $X^*$ denote the optimal set of $k$-means centers for $P$ and $\hat{X}^*$ denote the set of means for each cluster under the partition $\pi_1(\mathcal{P}_{P, X^*})$ $(\hat{X}^* = \pi_1(X^*)$ if $\pi_1$ is a linear map). Let $P''$ denote $\pi_1(P)$ and $S' := (S', \Delta, w)$ denote $\pi_2(P'')$. Then

\[
(1 + \epsilon)\text{cost}(P, X^*) = (1 + \epsilon)\text{cost}(P_{P', X'^*}) \geq \text{cost}(P', \hat{X}^*) \geq \text{cost}(S', \hat{X}^*) \geq \frac{1}{1 + \epsilon}\text{cost}(S', X^*)
\]

where (7) is due to the optimality of $X^*$, (8) is because $\pi_1$ is an $\epsilon$-projection, (9) is by Lemma III.1.1), (10) is because $S'$ is an $\epsilon$-creset of $P'$, (11) is because $X'$ is optimal for $S'$, (12) is because $S'$ is an $\epsilon$-creset of $P'$, (13) is by Lemma III.1.2), (14) is because $\pi_1$ is an $\epsilon$-projection, and (15) is by Lemma III.1.1). The last inequality (15) gives the desired upper bound on cost($P, X$).

**Discussion:** If $P$ resides on a data source and the $k$-means computation is performed by a server, then we have to transmit the entire dataset $P$ in order to implement Algorithm 1, as its solution is an explicit function of a partition of $P$. Thus, Algorithm 1 is only useful for reducing the complexity in solving $k$-means, but is not useful for reducing the communication cost. In contrast, for certain DR/CR methods, it is possible to achieve a similar approximation error at a reduced communication cost, as shown below.

2) **An Existing DR+CR Algorithm:** The state-of-the-art joint DR and CR algorithm, referred to as FSS following the authors’ last names, was implicitly presented in Theorem 36 [7]. The approximation error and the communication cost of FSS (for transmitting its output) were not given in [7]. Thus, we provide them (proved in [25]) to facilitate later comparison.

#### Corollary III.1.1

Suppose that the data source reports the creset $S := (S, \Delta, w)$ computed by FSS [7] and the server computes the optimal $k$-means centers $X$ of $S$. Then:

1. $X$ is a $(1 + \epsilon)/(1 - \epsilon)$-approximation for $k$-means clustering of $P$ with probability at least $1 - \delta$;
2. the communication cost is $O(kd/\epsilon^2)$, assuming $min(n, d) \gg k, 1/\epsilon, 1/\delta$.

3) **Communication-efficient DR+CR:** Now the question is: can we further improve the communication cost without hurting approximation and complexity?

---

Algorithm 1: $k$-Means under Generic DR+CR

**Input:** Original dataset $P$, number of centers $k$, DR method $\pi_1$, CR method $\pi_2$

**Output:** Centers for $k$-means clustering of $P$

1. $P'' \leftarrow \pi_1(P)$;
2. $(S', \Delta, w) \leftarrow \pi_2(P'')$;
3. $X' \leftarrow \text{kmeans}(S', w, k)$;
4. $P' \leftarrow \mathcal{P}_{P', X'}$;
5. $P \leftarrow \pi_1^{-1}(P')$ ($\mathcal{P} = \{P_1, \ldots, P_k\}$);
6. foreach $i = 1, \ldots, k$ do
7. $x_i \leftarrow \mu(P_i)$;
8. return $\{x_i\}_{i=1}^k$.
Our key observation is that, for FSS, the linear communication cost in $d$ is due to the transmission of the projected subspace. In contrast, JL projections are data-oblivious. Thus, we can circumvent the linear communication cost by employing a JL projection as the DR method. The following is directly implied by the JL Lemma (Lemma II.1); see the proof in [25].

**Lemma III.2.** Let $\pi : \mathbb{R}^d \to \mathbb{R}^d$ be a JL projection. Then there exists $d' = O(\epsilon^{-2} \log (nk/\delta))$ such that for any $P \subseteq \mathbb{R}^d$ with $|P| = n$ and $X, X^* \subseteq \mathbb{R}^d$ with $|X| = |X^*| = k$, the following holds with probability at least $1 - \delta$:

$$\text{cost}(P, X) \approx (1 + \epsilon)^2 \text{cost}(\pi(P), \pi(X)), \quad (16)$$

$$\text{cost}(P, X^*) \approx (1 + \epsilon^2) \text{cost}(\pi(P), \pi(X^*)). \quad (17)$$

Using a JL projection for DR and FSS for CR, we propose Algorithm 2, which differs from Algorithm 1 in that it directly computes the centers in the original space from the optimal centers in the low-dimensional space (line 7). Thus the communication cost is reduced by only transmitting $(S', \Delta, w)$. The following is the performance analysis of Algorithm 2.

**Theorem III.2.** For any $\epsilon, \delta \in (0, 1)$, if in Algorithm 2, $\pi_1$ satisfies Lemma III.2, $\pi_2$ generates an $\epsilon$-coreset with probability at least $1 - \delta$, and kmeans$(S', w, k)$ returns the optimal $k$-means centers of the dataset $S'$ with weights $w$, then

1) the output $X$ is a $(1 + \epsilon^5)/(1 - \epsilon)$-approximation for $k$-means clustering of $P$ with probability at least $(1 - \delta)^2$,

2) the communication cost is $O(\epsilon^{-4} \log n)$, and

3) the complexity at the data source is $O((ndc^{-2}) \log^2 n)$.

Proof. For 1), let $X^*$ be the optimal $k$-means centers of $P$ and $S' := (S', \Delta, w)$ be generated in line 3 of Algorithm 3. With probability at least $(1 - \delta)^2$, $\pi_1$ satisfies (16), (17) and $\pi_2$ generates an $\epsilon$-coreset. Thus, with probability at least $(1 - \delta)^2$,

$$\text{cost}(P, X) \leq (1 + \epsilon)^2 \text{cost}(\pi_1(P), X^*) \leq (1 + \epsilon^2) \text{cost}(S', X^*), \quad (18)$$

$$\text{cost}(P, X^*) \leq (1 + \epsilon^2) \text{cost}(\pi_2(P), \pi_1(X^*)). \quad (19)$$

For 2), the communication cost is dominated by transmitting $S'$. By Lemma III.2, the dimension of $P'$ is $d' = O(\epsilon^{-2} \log (nk/\delta)) = O(\epsilon^{-2} \log n)$. By Theorem II.2, the cardinality of $S'$ is $|S'| = O(k^3 \epsilon^{-4} \log^2 (k) (1/\delta))$. Moreover, points in $S'$ lie in a $d$-dimensional subspace for $d = O(k/\epsilon^2)$. Thus, it suffices to transmit the coordinates of points in $S'$ in the $d$-dimensional subspace and a basis of the subspace. Thus, the total communication cost is

$$O(|S'| + d')d = O\left(\frac{k^4}{\epsilon^6} \log^2 (k) \log (\frac{1}{\delta}) \right. + \frac{k}{\epsilon^2} \log n \right)$$

$$= O\left(\frac{k^4}{\epsilon^6} \log^2 (k) \log (\frac{1}{\delta}) \right. + \frac{k}{\epsilon^2} \log n \right). \quad (20)$$

Thus, the total complexity at the data source is:

$$O\left(\frac{nk^2}{\epsilon^2} \log^2 n + \log^2 (\frac{1}{\delta}) \right) + \log n \right). \quad (21)$$

**Remark:** We only focus on the complexity at the data source as the server is usually much more powerful. Theorem III.2 shows that Algorithm 2 can solve $k$-means arbitrarily close to the optimal with an arbitrarily high probability, while incurring a complexity at the data source that is roughly linear in the data size (i.e., $nd$) and a communication cost that is roughly logarithmic in the data cardinality (i.e., $n$).

**B. CR+DR**

While Algorithm 2 can drastically reduce the communication cost without incurring much computation, it remains unclear whether its order of applying DR and CR is optimal. To this end, we consider the approach of applying CR first.

1) A Black-box Approach: Now suppose that we first apply a generic CR method $\pi_2$ and then apply a generic DR method $\pi_1$. It turns out that the quality of the resulting $k$-means solution is the same as that in Theorem III.1, as stated below; the proof is similar to that of Theorem III.1 and hence left to [25].

**Theorem III.3.** Let $X'$ be the optimal $k$-means centers of $S' := (\pi_1(S), \Delta, w)$ for $(S, \Delta, w) := \pi_2(P)$, where $\pi_1$ is
Lemma III.3. Let \( \pi : \mathbb{R}^\mathcal{d} \rightarrow \mathbb{R}^{\mathcal{d}'} \) be a JL projection. Then there exists \( \mathcal{d}' = O\left(e^{-2}\log(n'k/\delta)\right) \) such that for any coreset \( \mathcal{S} := (S, \Delta, w) \), where \( S \subseteq \mathbb{R}^d \) with \( |S| = n' \), \( w : S \rightarrow \mathbb{R} \), and \( \Delta \in \mathbb{R} \), and any \( X, X^* \subseteq \mathbb{R}^d \) with \( |X| = |X^*| = k \), the following holds with probability at least \( 1 - \delta \):

\[
\begin{align*}
\text{cost}(\mathcal{S},X) & \approx \left(1 + \epsilon^2\right) \text{cost}\left(\pi(S),\Delta,w\right), \\
\text{cost}(\mathcal{S},X^*) & \approx \left(1 + \epsilon^2\right) \text{cost}\left(\pi(S),\Delta,w\right), \\
\end{align*}
\]

Below, we will show that Algorithm 4 achieves the same approximation as Algorithm 2, but at a different communication cost and a different complexity.

**Theorem III.4.** For any \( \epsilon, \delta \in (0,1) \), if in Algorithm 4, \( \pi_1 \) satisfies Lemma III.3, \( \pi_2 \) generates an \( \epsilon \)-coreset with probability at least \( 1 - \delta \), and \( \text{kmeans}(S', w, k) \) returns the optimal \( k \)-means centers of the dataset \( S' \) with weights \( w \), then

1. the output \( X \) is an \( (1 + \epsilon)^2/(1 - \epsilon) \)-approximation for \( k \)-means clustering of \( P \) with probability \( \geq (1 - \delta)^2 \),
2. the communication cost is \( O(\mathcal{d}'/(\epsilon^2)) \), and
3. the complexity at the data source is \( O(nd \cdot \min(n,d)) \), assuming \( \min(n,d) \gg k, 1/\epsilon, 1/\delta \).

**Proof.** For 1), let \( X^* \) be the optimal \( k \)-means centers of \( P \) and \( \mathcal{S} := (S, \Delta, w) \) generated in line 2 of Algorithm 4. With probability at least \( (1 - \delta)^2 \), \( \mathcal{S} \) is an \( \epsilon \)-coreset of \( P \), and \( \pi_1 \) satisfies (26, 27). Thus, with this probability,

\[
\begin{align*}
\text{cost}(P, X) & \leq \frac{1}{1 - \epsilon} \text{cost}(\mathcal{S}, X) \\
& \leq (1 + \epsilon)^2 \text{cost}(\pi(S'), \Delta, X') \\
& \leq (1 + \epsilon)^2 \text{cost}(\pi(S'), \Delta, \pi_1(X^*)) \\
& \leq (1 + \epsilon)^4 \text{cost}(\pi(S), \pi_1(X^*)) \\
& \leq (1 + \epsilon)^5 \text{cost}(\mathcal{S}, X^*) \\
& \leq (1 + \epsilon)^5 \frac{\text{cost}(P, X^*)}{1 - \epsilon} \\
\end{align*}
\]

where (28) is because \( \mathcal{S} \) is an \( \epsilon \)-coreset of \( P \), (29) is due to (26) (note that \( \pi_1(S') = S' \) and \( \pi_1(X^*) = X^* \)), (30) is because \( X' \) minimizes cost((\( S', \Delta, w \)), ), (31) is due to (27), and (32) is again because \( \mathcal{S} \) is an \( \epsilon \)-coreset of \( P \).

For 2), note that by Theorem II.2, the cardinality of \( S \) needs to be \( n' = O\left(k\mathcal{d}'\log(1/\delta)\right) \). By Lemma III.3, the dimension of \( S' \) needs to be \( \mathcal{d}' = O\left(e^{-2}\log(n'k/\delta)\right) \). Thus, the cost of transmitting \((S', \Delta, w)\) is dominated by the cost of transmitting \( S' \),

\[
O(n'd') = O\left(\frac{k^3 \log^2 k}{e^6} \log\left(\frac{1}{\delta}\right) \left(\log k + \log\left(\frac{1}{\epsilon}\right) + \log\left(\frac{1}{\delta}\right)\right)\right) = O\left(\frac{k^3}{e^6}\right).
\]

For 3), we know from Theorem II.2 that line 2 of Algorithm 4 takes time \( O\left(\min(nd^2, n'd^2) + nk\mathcal{d}' - 2(d + k \log(1/\delta))\right) \).

Given a projection matrix \( \Pi \in \mathbb{R}^{\mathcal{d} \times \mathcal{d}'} \) such that \( \pi_1(S) = \mathcal{A}_S \Pi \), line 3 takes time \( O(n'd'd') \). Thus, the total complexity at the data source is

\[
O\left(\min(nd^2n'd^2) + \frac{k^2 \log k}{e^2} n + \frac{k^3 \log^3 k (\log k + \log(1/\delta))}{e^6} d\right) = O\left(nd \cdot \min(n,d)\right).
\]
C. Comparison

We now summarize the above results via a comparison between the approach of applying DR first (i.e., DR+CR) and the approach of applying CR first (i.e., CR+DR).

Viewing the given DR/CR methods as black boxes, we have shown in Theorems III.1 and III.3 that the two approaches lead to the same approximation error. However, the approach of CR+DR has the advantage that the data source only needs to report coreset, resulting in a lower communication cost.

Utilizing the unique properties of carefully selected DR/CR methods, i.e., JL projection and FSS, we have shown in Theorems III.2 and III.4 that we can achieve an arbitrarily small approximation error with an arbitrarily high probability, while drastically reducing the communication cost and the complexity at the data source. Specifically, to achieve the same approximation error with the same probability, DR+CR (Algorithm 2) incurs a communication cost of \( O(ke^{-4} \log n) \) and a complexity of \( O(e^{-2}nd) \), while CR+DR (Algorithm 4) incurs a communication cost of \( O(k^3e^{-6}) \) and a complexity of \( O(nd \cdot \min(n, d)) \). This shows a communication-computation tradeoff: the approach of DR+CR, incurs a linear complexity and a logarithmic communication cost, whereas the approach of CR+DR incurs a super-linear complexity (which is still less than quadratic) and a constant communication cost.

Meanwhile, we note that by Theorem II.2 and Corollary III.1.1, the best existing solution FSS [7] requires a communication cost \( O(\min(n, d)) \), a complexity of \( O(nd \cdot \min(n, d)) \), worse than either of the proposed algorithms.

IV. EXTENSIONS

So far we have assumed that all the data reside on a single data source and each DR/CR method is only applied once. In this section, we will investigate more general scenarios without these limiting assumptions.

A. Multiple Data Sources

Consider the scenario where the entire dataset \( P \) is split across \( m \) data sources (\( m \geq 2 \)). Let \( P_i \) denote the dataset at data source \( i \) and \( n_i \) be its cardinality. As shown below, the previous algorithms can be adapted to the distributed setting.

1) Distributed Version of FSS: It turns out that the state-of-the-art distributed DR and CR algorithm, proposed in [16] (Algorithm 1), is exactly a distributed version of FSS, referred to as BKLW following the authors’ last names. As in FSS, BKLW first uses PCA to reduce the intrinsic dimension of the dataset and then applies sensitivity sampling. However, it uses two distributed algorithms to perform these steps.

For distributed PCA, BKLW applies an algorithm \( \text{disPCA} \) from [7] (formalized in Algorithm 1 in [26]), where:

1) each data source \( i (i = 1, \ldots, m) \) computes local SVD \( A_{P_i} = U_i\Sigma_iV_i^T \), and sends \( \Sigma_i^{(t)} \) and \( V_i^{(t)} \) to the server (\( \Sigma_i^{(t)} \) and \( V_i^{(t)} \) contain the first \( t_1 \) columns of \( \Sigma_i \) and \( V_i \), respectively);
2) the server constructs \( Y^T = [Y_1^T, \ldots, Y_m^T] \), with \( Y_i = \Sigma_i^{(t)}(V_i^{(t)})^T \), computes a global SVD \( Y = U\Sigma^TV \);
3) the first \( t_2 \) columns of \( V \) are returned as an approximate solution to the PCA of \( \bigcup_{i=1}^m P_i \).

For distributed sensitivity sampling, BKLW applies an algorithm \( \text{disSS} \) from [10] (Algorithm 1), where:

1) each data source \( i (i = 1, \ldots, m) \) computes a bicriteria approximation \( X_i \) for \( P_i \) and reports cost \( (P_i, X_i) \);
2) the server allocates a global sample size \( s \) to each data source proportionally to its cost, i.e., \( s_i = s \cdot \text{cost}(P_i, X_i)/(\sum_{j=1}^m \text{cost}(P_j, X_j)) \);
3) each data source \( i \) draws \( s_i \) i.i.d. samples \( S_i \) from \( P_i \) with probability proportional to cost \( \{\{P_i, X_i\} \), and reports \( S_i \cup X_i \) with their weights \( w : S_i \cup X_i \rightarrow \mathbb{R} \), that are set to match the number of points per cluster;
4) the union of the reported sets \( \bigcup_{i=1}^m (S_i \cup X_i), 0, w \) is returned as a coreset of \( \bigcup_{i=1}^m P_i \).

BKLW first applies \( \text{disPCA} \), followed by \( \text{disSS} \) with \( s = O(e^{-4}(k^2/\epsilon^2 + \log(1/\delta)) + mk \log(mk/\delta)) \) to the dimension-reduced dataset \( \{A_{P_i}V_i^{(t)}(V_i^{(t)})^T\}_{i=1}^m \) to compute a coreset \( (S, 0, w) \) at the server. Finally, the server computes the optimal \( k \)-means centers \( X \) on \( (S, 0, w) \) and returns it as an approximation to the optimal \( k \)-means centers of \( \bigcup_{i=1}^m P_i \).

Although a theorem was given in [16] without proof on the performance of BKLW, the result is imprecise and incomplete. Hence, we provide the complete analysis below to facilitate later comparison. We will leverage the following results.

Theorem IV.1 ([26]). For any \( \epsilon \in (0, 1/3) \), let \( t_1 = t_2 \geq k + \lceil 4k/\epsilon^2 \rceil - 1 \) in \( \text{disPCA} \) and \( \hat{P}_i \) be the projected dataset at data source \( i \) (i.e., the set of rows of \( A_P V_i^{(t)}(V_i^{(t)})^T \)). Then there exists a constant \( \Delta \geq 0 \) such that for any set \( X \subset \mathbb{R}^d \) with \( |X| = k \),

\[
(1-\epsilon)\text{cost}(P, X) \leq \text{cost}(\hat{P}, X) + \Delta \leq (1+\epsilon)\text{cost}(P, X),
\]

where \( P := \bigcup_{i=1}^m P_i \) and \( \hat{P} := \bigcup_{i=1}^m \hat{P}_i \).

Theorem IV.2 ([10]). For a distributed dataset \( \{P_i\}_{i=1}^m \) with \( P_i \subset \mathbb{R}^d \) and any \( \epsilon, \delta \in (0, 1) \), with probability at least \( 1 - \delta \), the output \( (S, 0, w) \) of \( \text{disSS} \) is an \( \epsilon \)-coreset of \( \bigcup_{i=1}^m P_i \) of size \( |S| = \mathcal{O}\left(\frac{1}{\epsilon^4} \left( kd + \log\left(\frac{1}{\delta}\right) \right) + mk \log\left(\frac{mk}{\delta}\right) \right) \).

Theorems IV.1 and IV.2 bound the performance of \( \text{disPCA} \) and \( \text{disSS} \), respectively, based on which we have the following results for BKLW (see proof in [25]).

Theorem IV.3. For any \( \epsilon \in (0, 1/3) \) and \( \delta \in (0, 1) \), suppose that in BKLW, \( \text{disPCA} \) satisfies Theorem IV.1 for the input dataset \( \{P_i\}_{i=1}^m \) and \( \text{disSS} \) satisfies Theorem IV.2 for the input dataset \( \{\hat{P}_i\}_{i=1}^m \). Then

1) the output \( X \) is a \( (1+\epsilon)^2/(1-\epsilon)^2 \)-approximation for \( k \)-means clustering of \( \bigcup_{i=1}^m P_i \) with probability \( \geq 1-\delta \),

\[5\] To achieve the same approximation error with the same probability as the proposed algorithms, we need to plug in different values for \( \epsilon \) and \( \delta \) for FSS. However, this will not affect the order of the required complexity and communication cost wrt \( n, d, \) and \( k \).
Lemma IV.1. Let $P := \bigcup_{i=1}^m P_i$ be the union of the input datasets for the BKLW-based CR method $\pi_2$ and $S := (S, 0, w)$ be the resulting coreset reported to the server. For any $\epsilon \in (0, 1/3)$ and $\delta \in (0, 1)$, $\exists t_2 = t_2 = O(k/\epsilon^2)$, $s = O(e^{-4}(k^2/\epsilon^2) + \log(1/\delta)) + mk \log(\log(1/\delta))$, and $\Delta \geq 0$, such that with probability at least $1 - \delta$, $\pi_2$ with parameters $t_1, t_2$, and $s$ satisfies
\begin{equation}
(1 - \epsilon)^2 \text{cost}(P, X) \leq \text{cost}(S, X) + \Delta \leq (1+\epsilon)^2 \text{cost}(P, X)
\end{equation}
for any set $X$ of $k$ points in the same space as $P$.

Remark: Comparing Lemma IV.1 with Definition II.2, we see that $\pi_2$ does not construct an $O(\epsilon)$-coreset of its input dataset. Nevertheless, its output can approximate the $k$-means cost of the input dataset up to a constant shift, which is sufficient for computing an approximate $k$-means solution.

Theorem IV.4. For any $\epsilon \in (0, 1/3)$ and $\delta \in (0, 1)$, suppose that in Algorithm 5, $\pi_1$ satisfies Lemma III.2, $\pi_2$ satisfies Lemma IV.1, and $\text{kmeans}(\bigcup_{i=1}^m S_i^*, w, k)$ returns the optimal $k$-means centers of the dataset $\bigcup_{i=1}^m S_i$ with weights $w$. Then
1) the output $X$ is a $(1+\epsilon)^6/(1-\epsilon)^2$-approximation for $k$-means clustering of $\bigcup_{i=1}^m P_i$ with probability $\geq (1-\delta)^2$, 
2) the total communication cost over all the data sources is $O(nmk^2/\epsilon^2)$, assuming $\min(n, d) \gg m, k, 1/\epsilon$, and $1/\delta$.

Proof. For 1), let $S' := (\bigcup_{i=1}^m S_i^*, 0, w)$ be the overall coreset constructed by line 3 of Algorithm 5, and $\Delta$ is a constant satisfying Lemma IV.1 for the input dataset $\{P_i\}^m_{i=1}$ as in line 3 of Algorithm 5. Let $P := \bigcup_{i=1}^m P_i$, and $X^*$ be the optimal $k$-means centers for $P$. Then with probability $\geq (1 - \delta)^2$, we have
\begin{equation}
\text{cost}(P, X) \leq (1 + \epsilon)^2 \text{cost}(\pi_1(P), X^*) \leq \frac{(1 + \epsilon)^2}{(1 - \epsilon)^2} \text{cost}(S', X^*) \leq \frac{(1 + \epsilon)^4}{(1 - \epsilon)^2} \text{cost}(\pi_1(P), \pi_1(X^*)) \leq \frac{(1 + \epsilon)^6}{(1 - \epsilon)^2} \text{cost}(P, X^*)
\end{equation}
where (38) is by Lemma III.2 (note that $\pi_1(X) = X^*$), (39) is by Lemma IV.1 (note that $\text{cost}(S', X^*) = \text{cost}(\bigcup_{i=1}^m S_i^*, 0, w, X^*) + \Delta$), (40) is because $X^*$ is optimal in minimizing $\text{cost}(S_i^*, -)$, (41) is again by Lemma IV.1, and (42) is again by Lemma III.2.

For 2), only line 3 incurs communication cost. By Theorem IV.3, we know that applying BKLW to a distributed dataset $\{P_i\}^m_{i=1}$ with dimension $d'$ incurs a cost of $O(nkd'/\epsilon^2)$, and by Lemma III.2, we know that $d' = O(\log n/\epsilon^2)$, which yields the desired result.

For 3), the JL projection at each data source incurs a complexity of $O(nkd'/\epsilon^2)$, and the complexity at each data source is $O(nkd' \log n/\epsilon^2)$. By Theorem IV.3, applying BKLW incurs a complexity of $O(nkd' \log n/\epsilon^2)$ at each data source. Together, the complexity is $O(n^2 \log n + \frac{n^2}{\epsilon^2} \log^2 n) = O(n^2), \square$

Discussion: Comparing Theorems IV.4 and IV.3, we see that for $d > \log n$ (e.g., $d = \Theta(n)$), Algorithm 5 can significantly reduce the communication cost and the complexity at data sources, while achieving a similar $(1+O(\epsilon))$-approximation as BKLW. Note that although the possibility of applying another DR method before BKLW was mentioned in [16], no result was given there.

Meanwhile, although one could develop a distributed counterpart of Algorithm 4 that applies JL projection after BKLW, its performance will not be competitive. Specifically, this approach incurs the same order of communication cost and complexity as BKLW. Meanwhile, the JL projection introduces additional error, causing its overall approximation error to be larger. It is thus unnecessary to consider this algorithm.

One might wonder why in the distributed setting, applying JL projection after FSS (i.e., Algorithm 4) improves the performance, but the idea does not work in the distributed setting. Fundamentally, it is because in the centralized setting, applying JL projection after FSS avoids transmitting the principal components. However, since in the distributed setting,
disPCA already requires the local principal components to be transmitted, applying JL projection later only reduces a lower-order term in the communication cost that is dominated by the cost of transmitting the local principal components.

### B. Repeated DR/CR

We now consider the possibility of applying certain DR/CR methods repeatedly.

1) **Distributed Setting**: We claim that with suitable DR/CR methods, repeated application of DR/CR is unnecessary in the distributed setting. This is because after one round of BKLW (with or without applying JL projection beforehand), we already reduce the cardinality to \(O(\epsilon^{-4}(k^2/c^2 + \log(1/\delta)) + mk \log(nk/\delta))\) and the dimension to \(O(k/c^2)\), both constant in the size \((n,d)\) of the original dataset. Meanwhile, this round incurs a communication cost that scales with \((n,d)\) as \(O(\log n)\) (with JL projection) or \(O(d)\) (without JL projection), and a complexity that scales as \(O(nd)\) (with JL projection) or \(O(nd \cdot \min(n,d))\) (without JL projection). Therefore, any possible reduction in the cost or the complexity achieved by further reducing the cardinality or dimension will be dominated by the cost or the complexity in the first round. Hence, repeated application of DR/CR will not improve the order of the communication cost or the complexity.

2) **Centralized Setting**: In contrast, we will show that repeated application can be beneficial in the centralized setting. Recall from Section III-C that applying JL projection before FSS (Algorithm 2) results in a lower complexity, while applying JL projection after FSS (Algorithm 4) results in a lower communication cost. One may wonder whether it is possible to combine the strengths of both the algorithms. Below we give an affirmative answer by applying some of these methods repeatedly.

Specifically, we know from Theorem II.2 that applying FSS once already reduces the cardinality to a constant \((\delta)\), and hence there is no need to repeat FSS. The same theorem also implies that if we apply FSS first, we will incur a super-linear complexity, and hence we need to apply JL projection before FSS. Meanwhile, we see from Lemmas III.2 and III.3 that applying JL projection on a dataset of cardinality \(n'\) can reduce its dimension to \(O(\epsilon^{-2} \log(n'k/\delta))\) while achieving a \((1+O(\epsilon))\)-approximation with high probability. Thus, we can further reduce the dimension by applying JL projection again after reducing the cardinality by FSS. The above reasoning suggests a three-step procedure: JL → FSS → JL, presented in Algorithm 6, where \(\pi_1^{(1)}\) projects from \(\mathbb{R}^d\) to \(\mathbb{R}^{O(\log n/c^2)}\), and \(\pi_1^{(2)}\) projects from \(\mathbb{R}^{O(\log n/c^2)}\) to \(\mathbb{R}^{O(\log |S|/c^2)}\). Note that by convention, \(\pi_1^{(2)} \circ \pi_1^{(1)}(X)\) means \(\pi_1^{(2)}(\pi_1^{(1)}(X))\).

Below, we will show that this seemingly small change is able to combine the low communication cost of Algorithm 4 and the low complexity of Algorithm 2, at a small increase in the approximation error; see [25] for the proof.

**Theorem IV.5.** For any \(\epsilon, \delta \in (0,1)\), if in Algorithm 6, \(\pi_1^{(1)}\) satisfies Lemma III.2, \(\pi_1^{(2)}\) satisfies Lemma III.3, \(\pi_2\) generates an \(\epsilon\)-coreset of its input dataset with probability at least \(1 - \delta\), and \(\text{kmeans}(S', w, k)\) returns the optimal \(k\)-means centers of the dataset \(S'\) with weights \(w\), then

1. the output \(X\) is a \((1+\epsilon)^3/(1-\epsilon)\)-approximation for \(k\)-means clustering of \(P\) with probability \(\geq (1 - \delta)^3\);
2. the communication cost is \(O(k^3/\epsilon^6)\), and
3. the complexity at the data source is \(O(nd/\epsilon^2)\), assuming \(\min(n,d) \gg k, 1/\epsilon, 1/\delta\).

**Remark:** Theorem IV.5 implies that Algorithm 6 is essentially “optimal” in the sense that it achieves a \((1 + O(\epsilon))\)-approximation with an arbitrarily high probability, at a near-linear complexity and a constant communication cost at the data source. Thus, no qualitative improvement will be achieved by applying further DR/CR methods.

### C. Summary of Comparison

We are now ready to compare the performances of all the proposed algorithms and their best existing counterparts in both centralized (i.e., single data source) and distributed (i.e., multiple data sources) settings.

To ensure the same approximation error for all the algorithms, we set the error parameter \(\epsilon\) to \(\epsilon_1\) for Algorithms 2 and 4, \(\epsilon_3\) for FSS, \(\epsilon_3\) for Algorithm 6, \(\epsilon_4\) for BKLW, and \(\epsilon_5\) for Algorithm 5, where for any \(\epsilon \in (0,1)\), \(\epsilon_1\) satisfies \((1+\epsilon_1)^5/(1-\epsilon_1) = 1+\epsilon\), \(\epsilon_2\) satisfies \((1+\epsilon_2)/(1-\epsilon_2) = 1+\epsilon\), \(\epsilon_3\) satisfies \((1+\epsilon_3)^9/(1-\epsilon_3) = 1+\epsilon\), \(\epsilon_4\) satisfies \((1+\epsilon_4)^2/(1-\epsilon_4) = 1+\epsilon\), and \(\epsilon_5\) satisfies \((1+\epsilon_5)^9/(1-\epsilon_5)^2 = 1+\epsilon\).

The comparison, summarized in Table I, is in terms of the communication cost and the complexity at the data source(s) for achieving a \((1 + \epsilon)\)-approximation for \(k\)-means clustering of an input dataset of cardinality \(n\) and dimension \(d\), where the first four rows are for the centralized setting and the last two rows are for the distributed setting. Clearly, for high-dimensional datasets satisfying \(d \gg \log n\), the best proposed algorithm significantly outperforms the best existing algorithm in both the centralized and the distributed settings.

### V. Performance Evaluation

In this section, we use experiments on real datasets to validate our analysis about the proposed joint DR and CR algorithms in comparison with the existing algorithms in both the single-source and the multiple-source cases.
A. Datasets and Metrics

We use two datasets in our experiments: (1) MNIST training dataset [27], a handwritten digits dataset which has 60,000 images in 784-dimensional space; (2) NeurIPS Conference Papers 1987-2015 dataset [28], a word counts dataset of the NeurIPS conference papers published from 1987 to 2015, which has 11,463 instances (words) with 5,812 attributes (papers). Both of these two datasets are normalized to $[-1, 1]$ with zero mean. In the case of multiple data sources, we randomly partition each dataset among 10 data sources.

We measure the performance by (i) the approximation error, measured by the normalized $k$-means cost cost($P, X'$)/cost($P, X$), where $X$ is the set of centers returned by the evaluated algorithm and $X'$ is the set of centers computed from $P$, (ii) the communication cost, measured by the number of scalars transmitted by the data source(s), and (iii) the complexity at the data source(s), measured by the running time of the evaluated DR/CR algorithm. We set $k = 2$ in all the experiments. Because of the randomness of the algorithms, we repeat each test for 10 Monte Carlo runs.

B. Algorithms

In the case of single data source, we evaluate 4 algorithms:
1) “FSS”: the benchmark algorithm introduced in [7],
2) “JL+FSS”: Algorithm 2, where we use JL projection before applying FSS,
3) “FSS+JL”: Algorithm 4, where we use JL projection after applying FSS, and
4) “JL+FSS+JL”: Algorithm 6, where we apply JL projection both before and after applying FSS.

In the case of multiple data sources, we evaluate 2 algorithms:
1) “BKLW”: the benchmark algorithm from [16], and
2) “JL+BKLW”: Algorithm 5, where we apply JL projection before BKLW.

In both cases, we have tuned the parameters of both the benchmark and proposed algorithms to make all the algorithms achieve a similar empirical approximation error. As a baseline, we also include the naive method of “no reduction (NR)”, i.e., transmitting the raw data.

C. Results

1) Single data source: The results for the case of single data source (that computes and transmits a data summary to a remote server) are given in Figures 1–2 and Table II. Note that by definition, the baseline (NR) has a normalized cost of 1 and no computation at the data source. We observe the following: (i) Compared to the naive method of transmitting the raw data (NR), the proposed algorithms can dramatically reduce the communication cost (by 99%) with a moderate increase in the $k$-means cost (< 16%). (ii) Compared to the benchmark (FSS), the proposed algorithms can achieve a similar or smaller $k$-means cost while significantly reducing the communication cost (in the case of MNIST) or complexity (in the case of NeurIPS). (iii) Between the approaches of DR+CR (JL+FSS) and CR+DR (FSS+JL), we see that the DR+CR approach yields a better performance in these experiments, especially for the NeurIPS dataset, where JL+FSS has a substantially smaller running time than FSS+JL but similar approximation error and communication cost. This is because $\log n \ll \min(n, d)$ here, allowing JL+FSS to significantly reduce the complexity compared with FSS+JL without blowing up the communication cost. (iv) Repeated applications of DR/CR (JL+FSS) can further reduce the communication cost (for NeurIPS) while obtaining comparable performance in the other metrics.

2) Multiple data sources: Figures 3–4 and Table III show the results for the case of multiple data sources. We see that on both datasets, the proposed algorithm (JL+BKLW) achieves a $k$-means cost comparable to the benchmark (BKLW), while incurring substantially lower complexity and communication cost. This demonstrates the benefit of suitably combining the JL projection with other data reduction methods; recall that applying JL projection after BKLW will not reduce the communication cost or the complexity as explained after Theorem IV.4. Note that although JL+BKLW slightly underperforms BKLW in terms of $k$-means cost on the NeurIPS dataset, the difference is insignificant (< 2%), which may be because of the randomness of JL projection.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Communication cost</th>
<th>Computation complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSS [7]</td>
<td>$O(kd/\epsilon^2)$</td>
<td>$O(nd \cdot \min(n,d))$</td>
</tr>
<tr>
<td>JL + FSS (Alg. 2)</td>
<td>$O(k \log n/\epsilon^2)$</td>
<td>$O(nd/\epsilon^2)$</td>
</tr>
<tr>
<td>FSS + JL (Alg. 4)</td>
<td>$O(k^3/\epsilon^2)$</td>
<td>$O(nd \cdot \min(n,d))$</td>
</tr>
<tr>
<td>JL + FSS + JL (Alg. 6)</td>
<td>$O(k^3/\epsilon^2)$</td>
<td>$O(nd/\epsilon^2)$</td>
</tr>
<tr>
<td>BKLW [16]</td>
<td>$O(mkd/\epsilon^2)$</td>
<td>$O(nd \cdot \min(n,d))$</td>
</tr>
<tr>
<td>JL + BKLW (Alg. 5)</td>
<td>$O(mkd/\epsilon^2)$</td>
<td>$O(nd/\epsilon^2)$</td>
</tr>
</tbody>
</table>

A SUMMARY OF CDF

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NR</th>
<th>FSS</th>
<th>JL+FSS</th>
<th>FSS+JL</th>
<th>JL+FSS+JL</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>47,040,000</td>
<td>402,021</td>
<td>179,416</td>
<td>179,416</td>
<td>182,419</td>
</tr>
<tr>
<td>NeurIPS</td>
<td>66,622,956</td>
<td>756,741</td>
<td>724,072</td>
<td>724,836</td>
<td>743,373</td>
</tr>
</tbody>
</table>
D. Summary of Observations

Our experimental results verified the following:

- Solving $k$-means based on data summaries generated by DR/CR methods can effectively reduce the communication cost without incurring a high complexity at the data sources, while providing a solution of reasonable quality.
- Suitable combination of DR and CR methods will further improve the communication cost and the complexity while maintaining a similar solution quality.

VI. CONCLUSION

In this paper, we considered the problem of jointly applying DR and CR methods to efficiently compute the $k$-means centers for a large high-dimensional dataset located at remote data source(s). Through a comprehensive analysis of the approximation error, the communication cost, and the complexity of various combinations of state-of-the-art DR/CR methods, we proved that it is possible to achieve a near-optimal approximation of $k$-means at a near-linear complexity at the data source(s) and a very low (constant or logarithmic) communication cost. In the process, we also developed algorithms based on novel combinations of existing DR/CR methods that outperformed two state-of-the-art algorithms. Our findings were validated through experiments on two real datasets.

REFERENCES