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 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[1]:

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

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:

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

This research was partly sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

[1]The norms in (1) and (2) refer to the ℓ-2 norm.

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,[2] (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-

[2]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.

dinality, d: dimension), residing at a data source (or sources) at the network edge. An obvious solution of solving k-means at the data source and sending the centers to the server will incur a high complexity at the data source, while another obvious solution of sending the data to the server and solving k-means there will incur a high communication cost. We seek to combine the best of the two by letting the data source send a small data summary generated by efficient DR/CR methods and leaving the k-means computation to the server.

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 ϵ-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[3] $\tilde{O}(kd\epsilon^{-4})$. The best known solution is the one in [7, Theorem 36], which showed that the cardinality of an ϵ-coreset can be reduced to $\tilde{O}(k^3\epsilon^{-4})$. 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 ϵ-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.

[3]We use $\tilde{O}(x)$ to denote a value that is at most linear in x times a factor that is polylogarithmic in x.

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 ℓ-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 $P_i := \{p \in P : \|p-x_i\| \le \|p-x_j\|, \forall x_j \in X \setminus \{x_i\}\}$ (ties broken arbitrarily). Given scalars x, y, and ϵ ($\epsilon > 0$), we will use $x \approx_{1+\epsilon} y$ to denote $\frac{1}{1+\epsilon}x \le y \le (1+\epsilon)x$. 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 $\pi : \mathbb{R}^d \to \mathbb{R}^{d'}$ ($d' < d$), we use $\pi(P) := \{\pi(p) : p \in P\}$ to denote the output dataset for an input dataset P, and $\pi(\mathcal{P}) := \{\pi(P_1), \ldots, \pi(P_k)\}$ to denote the partition of $\pi(P)$ corresponding to a partition $\mathcal{P} = \{P_1, \ldots, P_k\}$ of P. Moreover, given a partition $\mathcal{P}' = \pi(\mathcal{P})$, we use $\pi^{-1}(\mathcal{P}')$ 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

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 π. Note that there is no guarantee that $\pi^{-1}(P') = P$. However, solutions to $\pi(\tilde{P}) = P'$ must exist (P is a feasible solution) and $\pi^{-1}(P')$ denotes an arbitrary solution. If π 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* Π^+ [20] of Π 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 $\pi : \mathbb{R}^d \to \mathbb{R}^{d'}$ ($d' < d$) is an* ϵ-projection *if it preserves the cost of any partition up to a factor of $1+\epsilon$, i.e., cost$(\mathcal{P}) \approx_{1+\epsilon}$ cost$(\pi(\mathcal{P}))$ for every partition $\mathcal{P} = \{P_1, \ldots, P_k\}$ of a finite set $P \subset \mathbb{R}^d$.*

One commonly used method to construct ϵ-projection is random projection, where the cornerstone result is the JL Lemma:

Lemma II.1 ([21])**.** *There exists a family of random linear maps $\pi : \mathbb{R}^d \to \mathbb{R}^{d'}$ with the following properties: for every $\epsilon, \delta \in (0, 1/2)$, there exists $d' = O(\frac{\log(1/\delta)}{\epsilon^2})$ such that for every $d \geq 1$ and all $x \in \mathbb{R}^d$, we have $\Pr\{\|\pi(x)\| \approx_{1+\epsilon} \|x\|\} \geq 1 - \delta$.*

Based on this lemma, the best known result achieved by random projection is the following:

Theorem II.1 ([6])**.** *Consider any family of random linear maps $\pi : \mathbb{R}^d \to \mathbb{R}^{d'}$ that (i) satisfies Lemma II.1, and (ii) is sub-Gaussian-tailed (i.e., the probability for the norm after mapping to be larger than the norm before mapping by a factor of at least $1 + t$ is bounded by $e^{-\Omega(d't^2)}$). Then for every $\epsilon, \delta \in (0, 1/4)$, there exists $d' = O(\frac{1}{\epsilon^2}\log\frac{k}{\epsilon\delta})$, such that π is an ϵ-projection with probability at least $1 - \delta$.*

There are many known methods to construct a random linear map that satisfies the conditions (i–ii) in Theorem II.1, e.g., maps defined by matrices with i.i.d. Gaussian and sub-Gaussian entries [22]–[24]. We will refer to such a random projection as a *JL projection*.

Remark: Compared with PCA-based DR methods, JL projection has the advantage that the projection matrix is *data-oblivious*, and can hence be pre-generated and distributed, or generated independently by different nodes using a shared random number generation seed, both incurring negligible communication cost at runtime. As is shown later, this can lead to significant savings in the communication cost.

C. Cardinality Reduction for k-Means

CR methods, also known as *coreset construction algorithms*, aim at constructing a smaller weighted dataset (*coreset*) with a bounded approximation error as follows.

Definition II.2 ([7])**.** *We say that a tuple (S, Δ, w), where $S \subset \mathbb{R}^d$, $w : S \to \mathbb{R}$, and $\Delta \in \mathbb{R}$, is an ϵ-coreset of $P \subset \mathbb{R}^d$ if it preserves the cost for every set of k centers up to a factor of $1 \pm \epsilon$, i.e.,*

$$(1-\epsilon)\textit{cost}(P, X) \leq \textit{cost}(\mathbf{S}, X) \leq (1+\epsilon)\textit{cost}(P, X) \quad (3)$$

for any $X \subset \mathbb{R}^d$ with $|X| = k$, where

$$\textit{cost}(\mathbf{S}, X) := \sum_{q\in S} w(q) \cdot \min_{x_i \in X} \|q - x_i\|^2 + \Delta \quad (4)$$

denotes the k-means cost for a coreset $\mathbf{S} := (S, \Delta, w)$ and a set of centers X.

We note that the above definition generalizes most of the existing definitions of ϵ-coreset, which typically ignore Δ.

The best known coreset construction algorithm for k-means was given in [7], which first reduces the intrinsic dimension of the dataset by PCA, and then applies sensitivity sampling to the dimension-reduced dataset to obtain an ϵ-coreset of the original dataset with a size that is constant in n and d.

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

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 α-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,

$$\begin{aligned}\text{cost}(\mathcal{P}) &= \sum_{i=1}^{k}\sum_{p\in P_i}\|p-x_i\|^2 = \sum_{p\in P}\|p-x_{i_p}\|^2 \qquad (5)\\ &\geq \sum_{p\in P}\min_{x_i\in X}\|p-x_i\|^2 = \text{cost}(P,X).\end{aligned}$$

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

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

$$\begin{aligned}\text{cost}(P,X) &= \sum_{i=1}^{k}\sum_{p\in P_i}\|p-x_i\|^2\\ &\geq \sum_{i=1}^{k}\min_{q\in\mathbb{R}^d}\sum_{p\in P_i}\|p-q\|^2 = \text{cost}(\mathcal{P}_{P,X}), \qquad (6)\end{aligned}$$

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

Next we show the approximation error of first applying DR method π_1, followed by CR method π_2 is bounded as follows.

Theorem III.1. *In Algorithm 1, let* X' *be the optimal* k*-means centers*[4] *of* $\mathbf{S}' := \pi_2(\pi_1(P))$*, where* π_1 *is an* ϵ*-projection and* π_2 *generates an* ϵ*-coreset for some* $\epsilon \in (0,1)$*. Then* $X := \{\mu(P_i)\}_{i=1}^k$*, where* $\{P_1, \ldots, P_k\} = \pi_1^{-1}(\mathcal{P}_{\pi_1(P),X'})$*, 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 $\tilde{X}^*$ denote the set of means for each cluster under the partition $\pi_1(\mathcal{P}_{P,X^*})$ ($\tilde{X}^* = \pi_1(X^*)$ if π_1 is a linear map). Let P' denote $\pi_1(P)$ and $\mathbf{S}' := (S', \Delta, w)$ denote $\pi_2(P')$. Then

$$\begin{aligned}(1+\epsilon)\text{cost}(P,X^*) &= (1+\epsilon)\text{cost}(\mathcal{P}_{P,X^*}) &(7)\\ &\geq \text{cost}(\pi_1(\mathcal{P}_{P,X^*})) &(8)\\ &\geq \text{cost}(P',\tilde{X}^*) &(9)\\ &\geq \frac{1}{1+\epsilon}\text{cost}(\mathbf{S}',\tilde{X}^*) &(10)\\ &\geq \frac{1}{1+\epsilon}\text{cost}(\mathbf{S}',X') &(11)\\ &\geq \frac{1-\epsilon}{1+\epsilon}\text{cost}(P',X') &(12)\\ &\geq \frac{1-\epsilon}{1+\epsilon}\text{cost}(\mathcal{P}_{P',X'}) &(13)\\ &\geq \frac{1-\epsilon}{(1+\epsilon)^2}\text{cost}(\pi_1^{-1}(\mathcal{P}_{P',X'})) &(14)\\ &\geq \frac{1-\epsilon}{(1+\epsilon)^2}\text{cost}(P,X), &(15)\end{aligned}$$

where (7) is due to the optimality of X^*, (8) is because π_1 is an ϵ-projection, (9) is by Lemma III.1.1), (10) is because $\mathbf{S}'$ is an ϵ-coreset of P', (11) is because X' is optimal for $\mathbf{S}'$, (12) is because $\mathbf{S}'$ is an ϵ-coreset of P', (13) is by Lemma III.1.2), (14) is because π_1 is an ϵ-projection, and (15) is by Lemma III.1.1). The last inequality (15) gives the desired upper bound on cost(P, X). □

[4]Specifically, for $\mathbf{S}' = (S', \Delta, w)$, one can ignore Δ, and apply a weighted k-means algorithm to minimize $\sum_{q\in S'} w(q)\cdot\min_{x_i\in X}\|q-x_i\|^2$, or convert it into an unweighted dataset by duplicating each $q \in S'$ for $w(q)$ times and apply an unweighted k-means algorithm.

Algorithm 1: k-Means under Generic DR+CR

input : Original dataset P, number of centers k, DR method π_1, CR method π_2
output: Centers for k-means clustering of P

```
P' ← π1(P);
(S', Δ, w) ← π2(P');
X' ← kmeans(S', w, k);
P' ← P_{P',X'};
P ← π1^{-1}(P') (P = {P_1, ..., P_k});
foreach i = 1, ..., k do
    x_i ← μ(P_i);
return {x_i}_{i=1}^k;
```

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 coreset* $\mathbf{S} := (S, \Delta, w)$ *computed by FSS [7] and the server computes the optimal* k*-means centers* X *of* $\mathbf{S}$*. 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* $O\left(kd/\epsilon^2\right)$*,*

assuming $\min(n, d) \gg k, 1/\epsilon,$ *and* $1/\delta$*.*

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

Algorithm 2: k-Means under Communication-efficient DR+CR

input : Original dataset P, number of centers k, JL projection π_1, FSS-based CR method π_2
output: Centers for k-means clustering of P

1 **data source:**
2 $P' \leftarrow \pi_1(P)$;
3 $(S', \Delta, w) \leftarrow \pi_2(P')$;
4 report (S', Δ, w) to the server;
5 **server:**
6 $X' \leftarrow \text{kmeans}(S', w, k)$;
7 $X \leftarrow \pi_1^{-1}(X')$;
8 return X;

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 $P \subset \mathbb{R}^d$ with $|P| = n$ and $X, X^* \subset \mathbb{R}^d$ with $|X| = |X^*| = k$, the following holds with probability at least $1-\delta$:*

$$\textit{cost}(P, X) \approx_{(1+\epsilon)^2} \textit{cost}(\pi(P), \pi(X)), \tag{16}$$

$$\textit{cost}(P, X^*) \approx_{(1+\epsilon)^2} \textit{cost}(\pi(P), \pi(X^*)). \tag{17}$$

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', Δ, w). The following is the performance analysis of Algorithm 2.

Theorem III.2. *For any $\epsilon, \delta \in (0,1)$, if in Algorithm 2, π_1 satisfies Lemma III.2, π_2 generates an ϵ-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 $O\left(k\epsilon^{-4}\log n\right)$, and*
3) *the complexity at the data source is $\tilde{O}\left(nd\epsilon^{-2}\right)$,*

assuming $\min(n, d) \gg k, 1/\epsilon$, and $1/\delta$.

Proof. For 1), let X^* be the optimal k-means centers of P and $\mathbf{S}' := (S', \Delta, w)$ be generated in line 3 of Algorithm 2. With probability at least $(1-\delta)^2$, π_1 satisfies (16, 17) and π_2 generates an ϵ-coreset. Thus, with probability at least $(1-\delta)^2$,

$$\begin{aligned}
\text{cost}(P, X) &\le (1+\epsilon)^2 \text{cost}(\pi_1(P), X') && (18)\\
&\le \frac{(1+\epsilon)^2}{1-\epsilon}\text{cost}(\mathbf{S}', X') && (19)\\
&\le \frac{(1+\epsilon)^2}{1-\epsilon}\text{cost}(\mathbf{S}', \pi_1(X^*)) && (20)\\
&\le \frac{(1+\epsilon)^3}{1-\epsilon}\text{cost}(\pi_1(P), \pi_1(X^*)) && (21)\\
&\le \frac{(1+\epsilon)^5}{1-\epsilon}\text{cost}(P, X^*), && (22)
\end{aligned}$$

where (18) is by (16) and that $\pi_1(X) = X'$, (19) is because $\mathbf{S}'$ is an ϵ-coreset of $\pi_1(P)$, (20) is because X' minimizes $\text{cost}(\mathbf{S}', \cdot)$, (21) is again because $\mathbf{S}'$ is an ϵ-coreset of $\pi_1(P)$, and (22) is by (17).

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 $|S'| = O(k^3\epsilon^{-4}\log^2(k)\log(1/\delta))$. Moreover, points in S' lie in a $\tilde{d}$-dimensional subspace for $\tilde{d} = O(k/\epsilon^2)$. Thus, it suffices to transmit the coordinates of points in S' in the $\tilde{d}$-dimensional subspace and a basis of the subspace. Thus, the total communication cost is

$$\begin{aligned}
O((|S'| + d')\tilde{d}) &= O\left(\frac{k^4}{\epsilon^6}\log^2(k)\log(\frac{1}{\delta}) + \frac{k}{\epsilon^4}\log n\right)\\
&= O\left(\frac{k\log n}{\epsilon^4}\right). && (23)
\end{aligned}$$

For 3), note that for a given projection matrix $\Pi \in \mathbb{R}^{d\times d'}$ such that $\pi_1(P) := A_P\Pi$, line 2 takes $O(ndd') = O(nd\epsilon^{-2}\log n)$ time, where we have plugged in $d' = O(\epsilon^{-2}\log n)$. By Theorem II.2, line 3 takes time

$$\begin{aligned}
&O\left(\min(nd'^2, n^2d') + \frac{nk}{\epsilon^2}\left(d' + k\log(\frac{1}{\delta})\right)\right)\\
&= O\left(\frac{n}{\epsilon^2}\left(\frac{\log^2 n}{\epsilon^2} + \frac{k\log n}{\epsilon^2} + k^2\log(\frac{1}{\delta})\right)\right). && (24)
\end{aligned}$$

Thus, the total complexity at the data source is:

$$\begin{aligned}
&O\left(\frac{n}{\epsilon^2}\left(\frac{\log^2 n}{\epsilon^2} + \frac{k\log n}{\epsilon^2} + d\log n + k^2\log(\frac{1}{\delta})\right)\right)\\
&= O\left(\frac{nd}{\epsilon^2}\log^2 n\right) = \tilde{O}\left(\frac{nd}{\epsilon^2}\right). && (25)
\end{aligned}$$

□

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 π_2 and then apply a generic DR method π_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 $\mathbf{S}' := (\pi_1(S), \Delta, w)$ for $(S, \Delta, w) := \pi_2(P)$, where π_1 is*

Algorithm 3: k-Means under Generic CR+DR

input : Original dataset P, number of centers k, DR method π_1, CR method π_2
output: Centers for k-means clustering of P

1 $(S, \Delta, w) \leftarrow \pi_2(P)$;
2 $S' \leftarrow \pi_1(S)$;
3 $X' \leftarrow \text{kmeans}(S', w, k)$;
4 $\mathcal{S}' \leftarrow \mathcal{P}_{S',X'}$;
5 $\mathcal{S} \leftarrow \pi_1^{-1}(\mathcal{S}')$ ($\mathcal{S} = \{S_1, \ldots, S_k\}$);
6 **foreach** $i = 1, \ldots, k$ **do**
7 $\quad x_i \leftarrow \frac{1}{\sum_{p \in S_i} w(p)} \sum_{q \in S_i} w(q) q$;
8 return $\{x_i\}_{i=1}^k$;

Algorithm 4: k-Means under Communication-efficient CR+DR

input : Original dataset P, number of centers k, JL projection π_1, FSS-based CR method π_2
output: Centers for k-means clustering of P

1 **data source:**
2 $\quad (S, \Delta, w) \leftarrow \pi_2(P)$;
3 $\quad S' \leftarrow \pi_1(S)$;
4 $\quad$ report (S', Δ, w) to the server;
5 **server:**
6 $\quad X' \leftarrow \text{kmeans}(S', w, k)$;
7 $\quad X \leftarrow \pi_1^{-1}(X')$;
8 $\quad$ return X;

an ϵ-projection and π_2 generates an ϵ-coreset for $\epsilon \in (0,1)$. Then $X := \{\sum_{q \in S_i} \frac{w(q)q}{\sum_{p \in S_i} w(p)}\}_{i=1}^k$, where $\{S_1, \ldots, S_k\} = \pi_1^{-1}(\mathcal{P}_{\pi_1(S),X'})$, is a $(1+\epsilon)^3/(1-\epsilon)$-approximation for k-means clustering of P.

Discussion: If P resides on a data source and k-means is performed by a server, then we only need to transmit the coreset (S, Δ, w) to implement Algorithm 3. Compared to the approach of applying DR first as in Algorithm 1, applying CR first allows us to substantially reduce the communication cost for datasets with large cardinalities. However, as shown below, utilizing suitably selected CR/DR methods can further reduce the communication cost.

2) Communication-efficient CR+DR: Recall that applying DR first can reduce the communication cost to be logarithmic in the size of the original dataset (Theorem III.2). Below we will show that when applying CR first, this cost can be reduced to a constant, i.e., independent of both the cardinality n and the dimension d of the original dataset.

We again choose JL projection as the DR method to avoid transmitting the projection matrix at runtime and FSS as the CR method, which generates an ϵ-coreset with the minimum cardinality among the existing CR methods for k-means. The algorithm, shown in Algorithm 4, differs from Algorithm 2 in that the order of applying DR and CR is reversed.

We now analyze the performance of Algorithm 4, starting with a counterpart of Lemma III.2 (see proof in [25]).

Lemma III.3. *Let $\pi : \mathbb{R}^d \to \mathbb{R}^{d'}$ be a JL projection. Then there exists $d' = O(\epsilon^{-2} \log(n'k/\delta))$ such that for any coreset $\mathbf{S} := (S, \Delta, w)$, where $S \subset \mathbb{R}^d$ with $|S| = n'$, $w : S \to \mathbb{R}$, and $\Delta \in \mathbb{R}$, and any $X, X^* \subset \mathbb{R}^d$ with $|X| = |X^*| = k$, the following holds with probability at least $1 - \delta$:*

$$cost(\mathbf{S}, X) \approx_{(1+\epsilon)^2} cost((\pi(S), \Delta, w), \pi(X)), \tag{26}$$

$$cost(\mathbf{S}, X^*) \approx_{(1+\epsilon)^2} cost((\pi(S), \Delta, w), \pi(X^*)). \tag{27}$$

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, π_1 satisfies Lemma III.3, π_2 generates an ϵ-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 an $(1+\epsilon)^5/(1-\epsilon)$-approximation for k-means clustering of P with probability $\geq (1-\delta)^2$,*
2) *the communication cost is $\tilde{O}(k^3/\epsilon^6)$, and*
3) *the complexity at the data source is $O(nd \cdot \min(n,d))$,*

assuming $\min(n,d) \gg k, 1/\epsilon$, and $1/\delta$.

Proof. For 1), let X^* be the optimal k-means centers of P and $\mathbf{S} := (S, \Delta, w)$ generated in line 2 of Algorithm 4. With probability at least $(1-\delta)^2$, $\mathbf{S}$ is an ϵ-coreset of P, and π_1 satisfies (26, 27). Thus, with this probability,

$$\begin{aligned}
\text{cost}(P, X) &\leq \frac{1}{1-\epsilon} \text{cost}(\mathbf{S}, X) && (28)\\
&\leq \frac{(1+\epsilon)^2}{1-\epsilon} \text{cost}((S', \Delta, w), X') && (29)\\
&\leq \frac{(1+\epsilon)^2}{1-\epsilon} \text{cost}((S', \Delta, w), \pi_1(X^*)) && (30)\\
&\leq \frac{(1+\epsilon)^4}{1-\epsilon} \text{cost}(\mathbf{S}, X^*) && (31)\\
&\leq \frac{(1+\epsilon)^5}{1-\epsilon} \text{cost}(P, X^*), && (32)
\end{aligned}$$

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

For 2), note that by Theorem II.2, the cardinality of S needs to be $n' = O(k^3 \epsilon^{-4} \log^2(k) \log(1/\delta))$. By Lemma III.3, the dimension of S' needs to be $d' = O(\epsilon^{-2} \log(n'k/\delta))$. Thus, the cost of transmitting (S', Δ, w), dominated by the cost of transmitting S', is

$$\begin{aligned}
O(n'd') &= O\left(\frac{k^3 \log^2 k}{\epsilon^6} \log(\frac{1}{\delta}) \Big(\log k + \log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta}) \Big) \right)\\
&= \tilde{O}\left(\frac{k^3}{\epsilon^6}\right). && (33)
\end{aligned}$$

For 3), we know from Theorem II.2 that line 2 of Algorithm 4 takes time $O(\min(nd^2, n^2 d) + nk\epsilon^{-2}(d + k\log(1/\delta)))$. Given a projection matrix $\Pi \in \mathbb{R}^{d \times d'}$ such that $\pi_1(S) := A_S \Pi$, line 3 takes time $O(n'dd')$. Thus, the total complexity at the data source is

$$\begin{aligned}
&O\left(\min(nd^2, n^2 d) + \frac{k}{\epsilon^2} nd + \frac{k^2 \log k}{\epsilon^2} n + \frac{k^3 \log^3 k (\log k + \log(\frac{1}{\epsilon}))}{\epsilon^6} d\right)\\
&= O\left(nd \cdot \min(n,d)\right). && (34)
\end{aligned}$$

□

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 $O(k\epsilon^{-4}\log n)$ and a complexity of $\tilde{O}(\epsilon^{-2}nd)$, while CR+DR (Algorithm 4) incurs a communication cost of $\tilde{O}(k^3\epsilon^{-6})$ 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[5] a communication cost of $O(k\epsilon^{-2}d)$ and 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 *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 $\Sigma_i^{(t_1)}$ and $V_i^{(t_1)}$ to the server ($\Sigma_i^{(t_1)}$ and $V_i^{(t_1)}$ contain the first t_1 columns of Σ_i and V_i, respectively);
2) the server constructs $Y^T=[Y_1^T,\ldots,Y_m^T]$, with $Y_i=\Sigma_i^{(t_1)}(V_i^{(t_1)})^T$, computes a global SVD $Y=U\Sigma V^T$;
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 *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 $\text{cost}(P_i,X_i)$;
2) the server allocates a global sample size s to each data source proportionally to its cost, i.e., $s_i=s\cdot\text{cost}(P_i,X_i)/\big(\sum_{j=1}^m\text{cost}(P_j,X_j)\big)$;
3) each data source i draws s_i i.i.d. samples S_i from P_i with probability proportional to $\text{cost}(\{p\},X_i)$, and reports $S_i\cup X_i$ with their weights $w:\ S_i\cup X_i\to\mathbb{R}$, 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 disPCA, followed by disSS with $s=O(\epsilon^{-4}(k^2/\epsilon^2+\log(1/\delta))+mk\log(mk/\delta))$ to the dimension-reduced dataset $\{A_{P_i}V^{(t_2)}(V^{(t_2)})^T\}_{i=1}^m$ 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 disPCA and $\tilde{P}_i$ be the projected dataset at data source i (i.e., the set of rows of $A_{P_i}V^{(t_2)}(V^{(t_2)})^T$). Then there exists a constant $\Delta\geq 0$ such that for any set $X\subset\mathbb{R}^d$ with $|X|=k$,*

$$(1-\epsilon)\textit{cost}(P,X)\leq\textit{cost}(\tilde{P},X)+\Delta\leq(1+\epsilon)\textit{cost}(P,X), \quad (35)$$

where $P:=\bigcup_{i=1}^m P_i$ and $\tilde{P}:=\bigcup_{i=1}^m\tilde{P}_i$.

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 disSS is an ϵ-coreset of $\bigcup_{i=1}^m P_i$ of size*

$$|S|=O\left(\frac{1}{\epsilon^4}\Big(kd+\log(\frac{1}{\delta})\Big)+mk\log(\frac{mk}{\delta})\right). \quad (36)$$

Theorems IV.1 and IV.2 bound the performance of disPCA and 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, disPCA satisfies Theorem IV.1 for the input dataset $\{P_i\}_{i=1}^m$ and disSS satisfies Theorem IV.2 for the input dataset $\{\tilde{P}_i\}_{i=1}^m$. 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 ϵ and δ for FSS. However, this will not affect the order of the required complexity and communication cost wrt n, d, and k.

Algorithm 5: Distributed k-Means under DR+CR

input : Distributed dataset $\{P_i\}_{i=1}^m$, number of centers k, JL projection π_1, BKLW-based CR method π_2
output: Centers for k-means clustering of P

1 **each data source** i **(**$i = 1, \ldots, m$**):**
2 $\quad P_i' \leftarrow \pi_1(P_i)$;
3 run π_2 on the distributed dataset $\{P_i'\}_{i=1}^m$, which results in each data source i reporting a local coreset $(S_i', 0, w)$ to the server;
4 **server:**
5 $\quad X' \leftarrow \text{kmeans}(\bigcup_{i=1}^m S_i', w, k)$;
6 $\quad X \leftarrow \pi_1^{-1}(X')$;
7 $\quad$ return X;

2) *the total communication cost over all the data sources is* $O(mkd/\epsilon^2)$*, and*

3) *the complexity at each data source is* $O(nd \cdot \min(n, d))$*, assuming* $\min(n, d) \gg m, k, 1/\epsilon$*, and* $1/\delta$*.*

2) Enhancements: It is easy to see that each data source can apply JL projection independently at no additional communication cost at runtime. Following the ideas in Algorithms 2 and 4, we wonder: (i) Can we improve BKLW by combining it with JL projection? (ii) Is there an optimal order of applying BKLW and JL projection?

To this end, we first consider applying JL projection before invoking BKLW. For consistency with Algorithm 2, we only use the first two steps of BKLW, i.e., disPCA and disSS, that construct a coreset, which we refer to as a *BKLW-based CR method*. The algorithm, shown in Algorithm 5, is essentially the distributed counterpart of Algorithm 2.

We now analyze the performance of Algorithm 5, starting from a coreset-like property of π_2 (see proof in [25]).

Lemma IV.1. *Let* $P := \bigcup_{i=1}^m P_i$ *be the union of the input datasets for the BKLW-based CR method* π_2 *and* $\mathbf{S} := (S, 0, w)$ *be the resulting coreset reported to the server. For any* $\epsilon \in (0, 1/3)$ *and* $\delta \in (0, 1)$*,* $\exists t_1 = t_2 = O(k/\epsilon^2)$*,* $s = O(\epsilon^{-4}(k^2/\epsilon^2 + \log(1/\delta)) + mk\log(mk/\delta))$*, and* $\Delta \geq 0$*, such that with probability at least* $1 - \delta$*,* π_2 *with parameters* t_1*,* t_2*, and* s *satisfies*

$$(1-\epsilon)^2 cost(P, X) \leq cost(\mathbf{S}, X) + \Delta \leq (1+\epsilon)^2 cost(P, X) \quad (37)$$

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 π_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,* π_1 *satisfies Lemma III.2,* π_2 *satisfies Lemma IV.1, and 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* $O(mk\epsilon^{-4}\log n)$*, and*
3) *the complexity at each data source is* $\tilde{O}(nd\epsilon^{-4})$*, assuming* $\min(n, d) \gg m, k, 1/\epsilon$*, and* $1/\delta$*.*

Proof. For 1), let $\mathbf{S}' := (\bigcup_{i=1}^m S_i', \Delta, w)$, where $(\bigcup_{i=1}^m S_i', 0, w)$ is the overall coreset constructed by line 3 of Algorithm 5, and Δ is a constant satisfying Lemma IV.1 for the input dataset $\{P_i'\}_{i=1}^m$ 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

$$\begin{aligned}
\text{cost}(P, X) &\leq (1+\epsilon)^2 \text{cost}(\pi_1(P), X') && (38)\\
&\leq \frac{(1+\epsilon)^2}{(1-\epsilon)^2} \text{cost}(\mathbf{S}', X') && (39)\\
&\leq \frac{(1+\epsilon)^2}{(1-\epsilon)^2} \text{cost}(\mathbf{S}', \pi_1(X^*)) && (40)\\
&\leq \frac{(1+\epsilon)^4}{(1-\epsilon)^2} \text{cost}(\pi_1(P), \pi_1(X^*)) && (41)\\
&\leq \frac{(1+\epsilon)^6}{(1-\epsilon)^2} \text{cost}(P, X^*), && (42)
\end{aligned}$$

where (38) is by Lemma III.2 (note that $\pi_1(X) = X'$), (39) is by Lemma IV.1 (note that $\text{cost}(\mathbf{S}', X') = \text{cost}((\bigcup_{i=1}^m S_i', 0, w), X') + \Delta$), (40) is because X' is optimal in minimizing $\text{cost}(\mathbf{S}', \cdot)$, (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 $\{P_i'\}_{i=1}^m$ with dimension d' incurs a cost of $O(mkd'/\epsilon^2)$, 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 $O(ndd') = O(nd\log n/\epsilon^2)$. By Theorem IV.3, applying BKLW incurs a complexity of $O(nd' \cdot \min(n, d')) = O(n\log^2 n/\epsilon^4)$ at each data source. Together, the complexity is $O(\frac{nd}{\epsilon^2}\log n + \frac{n}{\epsilon^4}\log^2 n) = \tilde{O}(nd/\epsilon^4)$. □

Discussion: Comparing Theorems IV.4 and IV.3, we see that for $d \gg \log n$ (e.g., $d = \Omega(n)$), 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, using similar analysis, 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 centralized 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 $O(\epsilon^{-4}(k^2/\epsilon^2 + \log(1/\delta)) + mk\log(mk/\delta))$ and the dimension to $O(k/\epsilon^2)$, 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 $\tilde{O}(nd)$ (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 of 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 (in n and d), 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 $\mathbb{R}^{O(\log n/\epsilon^2)}$, and $\pi_1^{(2)}$ projects from $\mathbb{R}^{O(\log n/\epsilon^2)}$ to $\mathbb{R}^{O(\log |S|/\epsilon^2)}$. 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, π_2 generates an ϵ-coreset of its input dataset 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 $(1+\epsilon)^9/(1-\epsilon)$-approximation for k-means clustering of P with probability $\geq (1-\delta)^3$,*
2) *the communication cost is $\tilde{O}(k^3/\epsilon^6)$, and*
3) *the complexity at the data source is $\tilde{O}(nd/\epsilon^2)$,*

assuming $\min(n, d) \gg k, 1/\epsilon$, and $1/\delta$.

Algorithm 6: k-Means under Communication-efficient DR+CR+DR

input : Original dataset P, number of centers k, JL projection $\pi_1^{(1)}$ for P, FSS-based CR method π_2, JL projection $\pi_1^{(2)}$ for the output of π_2
output: Centers for k-means clustering of P

data source:
- $P' \leftarrow \pi_1^{(1)}(P)$;
- $(S, \Delta, w) \leftarrow \pi_2(P')$;
- $S' \leftarrow \pi_1^{(2)}(S)$;
- report (S', Δ, w) to the server;

server:
- $X' \leftarrow \text{kmeans}(S', w, k)$;
- $X \leftarrow (\pi_1^{(2)} \circ \pi_1^{(1)})^{-1}(X')$;
- return X;

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 'ϵ' to ϵ_1 for Algorithms 2 and 4, ϵ_2 for FSS, ϵ_3 for Algorithm 6, ϵ_4 for BKLW, and ϵ_5 for Algorithm 5, where for any $\epsilon \in (0, 1)$, ϵ_1 satisfies $(1+\epsilon_1)^5/(1-\epsilon_1) = 1+\epsilon$, ϵ_2 satisfies $(1+\epsilon_2)/(1-\epsilon_2) = 1+\epsilon$, ϵ_3 satisfies $(1+\epsilon_3)^9/(1-\epsilon_3) = 1+\epsilon$, ϵ_4 satisfies $(1+\epsilon_4)^2/(1-\epsilon_4)^2 = 1+\epsilon$, and ϵ_5 satisfies $(1+\epsilon_5)^6/(1-\epsilon_5)^2 = 1+\epsilon$.

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.

TABLE I
SUMMARY OF COMPARISON

Algorithm	Communication cost	Computation complexity
FSS [7]	$O(kd/\epsilon_2^2)$	$O(nd \cdot \min(n,d))$
JL + FSS (Alg. 2)	$O(k \log n/\epsilon_1^4)$	$\tilde{O}(nd/\epsilon_1^2)$
FSS + JL (Alg. 4)	$\tilde{O}(k^3/\epsilon_1^6)$	$O(nd \cdot \min(n,d))$
JL + FSS + JL (Alg. 6)	$\tilde{O}(k^3/\epsilon_3^6)$	$\tilde{O}(nd/\epsilon_3^2)$
BKLW [16]	$O(mkd/\epsilon_4^2)$	$O(nd \cdot \min(n,d))$
JL + BKLW (Alg. 5)	$O(mk \log n/\epsilon_5^4)$	$\tilde{O}(nd/\epsilon_5^4)$

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 $\text{cost}(P, X)/\text{cost}(P, X^*)$, 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

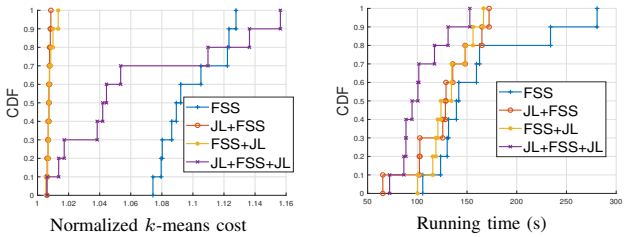

Fig. 1. Single-source case: MNIST

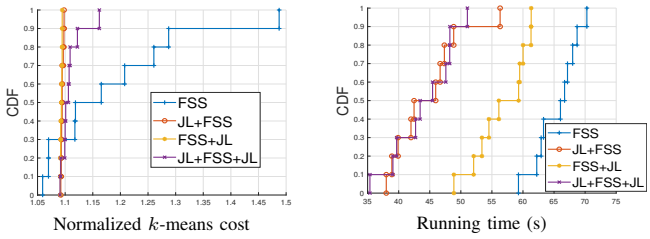

Fig. 2. Single-source case: NeurIPS

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+JL) 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 II
COMMUNICATION COST: SINGLE-SOURCE CASE

Dataset	NR	FSS	JL+FSS	FSS+JL	JL+FSS+JL
MNIST	47,040,000	402,021	179,416	179,416	182,419
NeurIPS	66,622,956	756,741	724,072	724,836	543,373

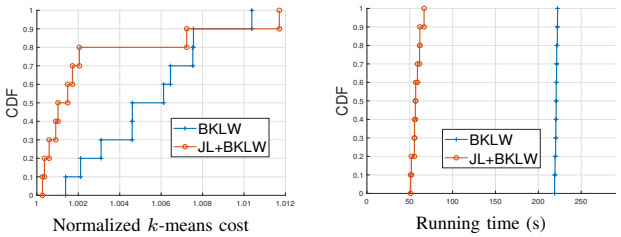

Fig. 3. Multiple-source case: MNIST

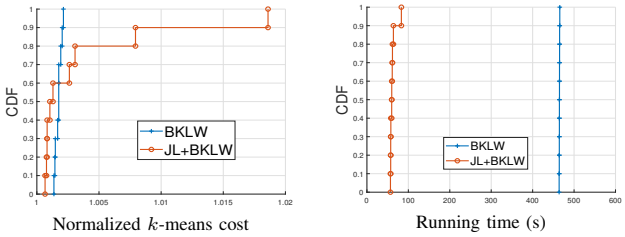

Fig. 4. Multiple-source case: NeurIPS

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.

TABLE III
Communication Cost: Multiple-source Case

Dataset	NR	BKLW	JL+BKLW
MNIST	47,040,000	27,152,484	20,365,463
NeurIPS	66,622,956	20,993,951	14,036,651

References

[1] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[2] H. Lu, M.-J. Li, T. He, S. Wang, V. Narayanan, and K. S. Chan, "Robust coreset construction for distributed machine learning," in *IEEE Globecom*, December 2019.

[3] ——, "Robust coreset construction for distributed machine learning," 2019. [Online]. Available: http://arxiv.org/abs/1904.05961

[4] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "NP-hardness of Euclidean sum-of-squares clustering," *Machine Learning*, vol. 75, no. 2, pp. 245–248, 2009.

[5] M. Mahajan, P. Nimbhorkar, and K. R. Varadarajan, "The planar k-means problem is NP-hard," *Theoretical Computer Science*, vol. 442, no. 13, pp. 13–21, 2012.

[6] K. Makarychev, Y. Makarychev, and I. Razenshteyn, "Performance of Johnson-Lindenstrauss transform for k-means and k-medians clustering," in *STOC*, June 2019.

[7] D. Feldman, M. Schmidt, and C. Sohler, "Turning big data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering," 2018. [Online]. Available: https://arxiv.org/abs/1807.04518

[8] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proceedings of the IEEE*, vol. 107, no. 11, pp. 2204–2239, 2019.

[9] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[10] M. F. Balcan, S. Ehrlich, and Y. Liang, "Distributed k-means and k-median clustering on general topologies," in *NIPS*, December 2013.

[11] C. Boutsidis, A. Zouzias, and P. Drineas, "Random projections for k-means clustering," in *NIPS*, December 2010.

[12] M. B. Cohen, S. Elder, C. Musco, C. Musco, and M. Persu, "Dimensionality reduction for k-means clustering and low rank approximation," in *STOC*, June 2015.

[13] S. Har-Peled and S. Mazumdar, "On coresets for k-means and k-median clustering," in *STOC*, 2004.

[14] D. Feldman and M. Langberg, "A unified framework for approximating and clustering data," in *STOC*, June 2011.

[15] D. Feldman, M. Schmidt, and C. Sohler, "Turning big data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering," in *SODA*, 2013.

[16] M. F. Balcan, V. Kanchanapally, Y. Liang, and D. Woodruff, "Improved distributed principal component analysis," in *NIPS*, December 2014.

[17] Y. Mao, Z. Xu, P. Ping, and L. Wang, "An optimal distributed k-means clustering algorithm based on CloudStack," in *International Conference on Frontier of Computer Science and Technology*, August 2015.

[18] M. C. Naldi and R. J. G. B. Campello, "Distributed k-means clustering with low transmission cost," in *Brazilian Conference on Intelligent Systems*, October 2013.

[19] C. R. Giannella, H. Kargupta, and S. Datta, "Approximate distributed k-means clustering over a peer-to-peer network," *IEEE Trans. KDE*, vol. 21, pp. 1372–1388, October 2009.

[20] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*. Springer, 2003.

[21] W. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," in *Conference in Modern Analysis and Probability*, 1982.

[22] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *ACM STOC*, 1998.

[23] D. Achlioptas, "Database-friendly random projections: Johnson-Lindenstrauss with binary coins," *Journal of Computer and System Sciences*, vol. 66, no. 4, pp. 671–687, 2003.

[24] B. Klartag and S. Mendelson, "Empirical processes and random projections," *Journal of Functional Analysis*, vol. 225, no. 1, pp. 229–245, August 2005.

[25] H. Lu, T. He, S. Wang, C. Liu, M. Mahdavi, V. Narayanan, K. S. Chan, and S. Pasteris, "Communication-efficient k-means for edge-based machine learning," Technical Report, January 2020. [Online]. Available: https://sites.psu.edu/nsrg/files/2020/01/Lu20ICDCSreport.pdf

[26] M. F. Balcan, V. Kanchanapally, Y. Liang, and D. Woodruff, "Improved distributed principal component analysis," 2014. [Online]. Available: https://arxiv.org/abs/1408.5823

[27] Y. LeCun, C. Cortes, and C. Burges, "The MNIST database of handwritten digits," http://yann.lecun.com/exdb/mnist/, 1998.

[28] V. Perrone, P. A. Jenkins, D. Spano, and Y. W. Teh, "Poisson random fields for dynamic feature models," *arXiv preprint arXiv:1611.07460*, 2016.