Communication-Optimal Distributed Clustering

Jiecao Chen
Department of Computer Science
Indiana University
Bloomington, IN 47401
jiecchen@indiana.edu

He Sun
Department of Computer Science
University of Bristol
Bristol, BS8 1UB, UK
h.sun@bristol.ac.uk

David P. Woodruff
IBM Research Almaden
San Jose, CA 95120
dpwoodru@us.ibm.com

Qin Zhang
Department of Computer Science
Indiana University
Bloomington, IN 47401
qzhangcs@indiana.edu

Abstract

Clustering large datasets is a fundamental problem with a number of applications in machine learning. Data is often collected on different sites and clustering needs to be performed in a distributed manner with low communication. We would like the quality of the clustering in the distributed setting to match that in the centralized setting for which all the data resides on a single site. In this work, we study both graph and geometric clustering problems in two distributed models: (1) a point-to-point model, and (2) a model with a broadcast channel. We give protocols in both models which we show are nearly optimal by proving almost matching communication lower bounds. Our work highlights the surprising power of a broadcast channel for clustering problems; roughly speaking, to cluster n points or n vertices in a graph distributed across s servers, for a worst-case partitioning the communication complexity in a point-to-point model is $n \cdot s$, while in the broadcast model it is $n + s$. We implement our algorithms and demonstrate this phenomenon on real life datasets, showing that our algorithms are also very efficient in practice.

1 Introduction

Clustering is a fundamental task in machine learning with widespread applications in data mining, computer vision, and social network analysis. Example applications of clustering include grouping similar webpages by search engines, finding users with common interests in a social network, and identifying different objects in a picture or video. For these applications, one can model the objects that need to be clustered as points in Euclidean space $\mathbb{R}^d$, where the similarities of two objects are represented by the Euclidean distance between the two points. Then the task of clustering is to choose k points as centers, so that the total distance between all input points to their corresponding closest center is minimized. Depending on different distance objective functions, three typical problems have been studied: k-means, k-median, and k-center.

The other popular approach for clustering is to model the input data as vertices of a graph, and the similarity between two objects is represented by the weight of the edge connecting the corresponding vertices. For this scenario, one is asked to partition the vertices into clusters so that the "highly connected" vertices belong to the same cluster. A widely-used approach for graph clustering is *spectral clustering*, which embeds the vertices of a graph into the points in $\mathbb{R}^k$ through the bottom k eigenvectors of the graph's Laplacian matrix, and applies k-means on the embedded points.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Both the spectral clustering and the geometric clustering algorithms mentioned above have been widely used in practice, and have been the subject of extensive theoretical and experimental studies over the decades. However, these algorithms are designed for the centralized setting, and are not applicable in the setting of large-scale datasets that are maintained remotely by different sites. In particular, collecting the information from all the remote sites and performing a centralized clustering algorithm is infeasible due to high communication costs, and new distributed clustering algorithms with low communication cost need to be developed.

There are several natural communication models, and we focus on two of them: (1) a point-to-point model, and (2) a model with a broadcast channel. In the former, sometimes referred to as the *message-passing model*, there is a communication channel between each pair of users. This may be impractical, and the so-called *coordinator model* can often be used in place; in the coordinator model there is a centralized site called the coordinator, and all communication goes through the coordinator. This affects the total communication by a factor of two, since the coordinator can forward a message from one server to another and therefore simulate a point-to-point protocol. There is also an additional additive $O(\log s)$ bits per message, where s is the number of sites, since a server must specify to the coordinator where to forward its message. In the model with a broadcast channel, sometimes referred to as the *blackboard model*, the coordinator has the power to send a single message which is received by all s sites at once. This can be viewed as a model for single-hop wireless networks.

In both models we study the total number of bits communicated among all sites. Although the blackboard model is at least as powerful as the message-passing model, it is often unclear how to exploit its power to obtain better bounds for specific problems. Also, for a number of problems the communication complexity is the same in both models, such as computing the sum of s length-n bit vectors modulo two, where each site holds one bit vector [17], or estimating large moments [19]. Still, for other problems like set disjointness it can save a factor of s in the communication [5].

Our contributions. We present algorithms for graph clustering: for any n-vertex graph whose edges are arbitrarily partitioned across s sites, our algorithms have communication cost $\widetilde{O}(ns)$ in the message passing model, and have communication cost $\widetilde{O}(n+s)$ in the blackboard model, where the $\widetilde{O}$ notation suppresses polylogarithmic factors. The algorithm in the message passing model has each site send a *spectral sparsifier* of its local data to the coordinator, who then merges them in order to obtain a spectral sparsifier of the union of the datasets, which is sufficient for solving the graph clustering problem. Our algorithm in the blackboard model is technically more involved, as we show a particular recursive sampling procedure for building a spectral sparsifier can be efficiently implemented using a broadcast channel. It is unclear if other natural ways of building spectral sparsifiers can be implemented with low communication in the blackboard model. Our algorithms demonstrate the surprising power of the blackboard model for clustering problems. Since our algorithms compute sparsifiers, they also have applications to solving symmetric diagonally dominant linear systems in a distributed model. Any such system can be converted into a system involving a Laplacian (see, e.g., [1]), from which a spectral sparsifier serves as a good preconditioner.

Next we show that $\Omega(ns)$ bits of communication is necessary in the message passing model to even recover a constant fraction of a cluster, and $\Omega(n+s)$ bits of communication is necessary in the blackboard model. This shows the optimality of our algorithms up to poly-logarithmic factors.

We then study clustering problems in constant-dimensional Euclidean space. We show for any $c > 1$, computing a c-approximation for k-median, k-means, or k-center correctly with constant probability in the message passing model requires $\Omega(sk)$ bits of communication. We then strengthen this lower bound, and show even for *bicriteria* clustering algorithms, which may output a constant factor more clusters and a constant factor approximation, our $\Omega(sk)$ bit lower bound still holds. Our proofs are based on communication and information complexity. Our results imply that existing algorithms [3] for k-median and k-means with $\widetilde{O}(sk)$ bits of communication, as well as the folklore parallel guessing algorithm for k-center with $\widetilde{O}(sk)$ bits of communication, are optimal up to poly-logarithmic factors. For the blackboard model, we present an algorithm for k-median and k-means that achieves an $O(1)$-approximation using $\widetilde{O}(s+k)$ bits of communication. This again separates the models.

We give empirical results which show that using spectral sparsifiers preserves the quality of spectral clustering surprisingly well in real-world datasets. For example, when we partition a graph with over 70 million edges (the `Sculpture` dataset) into 30 sites, only 6% of the input edges are communicated in the blackboard model and 8% are communicated in the message passing model, while the values

of the normalized cut (the objective function of spectral clustering) given in those two models are at most 2% larger than the one given by the centralized algorithm, and the visualized results are almost identical. This is strong evidence that spectral sparsifiers can be a powerful tool in practical, distributed computation. When the number of sites is large, the blackboard model incurs significantly less communication than the message passing model, e.g., in the `Twomoons` dataset when there are 90 sites, the message passing model communicates 9 times as many edges as communicated in the blackboard model, illustrating the strong separation between these models that our theory predicts.

Related work. There is a rich literature on spectral and geometric clustering algorithms from various aspects (see, e.g., [2, 15, 16, 18]). Balcan et al. [3, 4] and Feldman et al. [9] study distributed k-means ([3] also studies k-median), and present provable guarantees on the clustering quality. Cohen et al. [7] study dimensionality reduction techniques for the input data matrices that can be used for distributed k-means. The main takeaway from previous work is that there is no previous work which develops protocols for spectral clustering in the common message passing and blackboard models, and lower bounds are lacking as well. For geometric clustering, while upper bounds exist (e.g., [3, 4, 9]), no provable lower bounds in either model existed, and our main contribution is to show that previous algorithms are optimal. We also develop a new protocol in the blackboard model.

2 Preliminaries

Let $G = (V, E, w)$ be an undirected graph with n vertices, m edges, and weight function $V \times V \to \mathbb{R}_{\geq 0}$. The set of neighbors of a vertex v is represented by $N(v)$, and its degree is $d_v = \sum_{u \sim v} w(u, v)$. The maximum degree of G is defined to be $\Delta(G) = \max_v\{d_v\}$. For any set $S \subseteq V$, let $\mu(S) \triangleq \sum_{v \in S} d_v$. For any sets $S, T \subseteq V$, we define $w(S, T) \triangleq \sum_{u \in S, v \in T} w(u, v)$ to be the total weight of edges crossing S and T. For two sets X and Y, the symmetric difference of X and Y is defined as $X \triangle Y \triangleq (X \setminus Y) \cup (Y \setminus X)$.

For any matrix $A \in \mathbb{R}^{n \times n}$, let $\lambda_1(A) \leq \cdots \leq \lambda_n(A) = \lambda_{\max}(A)$ be the eigenvalues of A. For any two matrices $A, B \in \mathbb{R}^{n \times n}$, we write $A \preceq B$ to represent $B - A$ is positive semi-definite (PSD). Notice that this condition implies that $x^\intercal A x \leq x^\intercal B x$ for any $x \in \mathbb{R}^n$. Sometimes we also use a weaker notation $(1-\varepsilon)A \preceq_r B \preceq_r (1+\varepsilon)A$ to indicate that $(1-\varepsilon)x^\intercal A x \leq x^\intercal B x \leq (1+\varepsilon)x^\intercal A x$ for all x in the row span of A.

Graph Laplacian. The Laplacian matrix of G is an $n \times n$ matrix L_G defined by $L_G = D_G - A_G$, where A_G is the adjacency matrix of G defined by $A_G(u, v) = w(u, v)$, and D_G is the $n \times n$ diagonal matrix with $D_G(v, v) = d_v$ for any $v \in V[G]$. Alternatively, we can write L_G with respect to a *signed edge-vertex incidence matrix*: we assign every edge $e = \{u, v\}$ an arbitrary orientation, and let $B_G(e, v) = 1$ if v is e's head, $B_G(e, v) = -1$ if v is e's tail, and $B_G(e, v) = 0$ otherwise. We further define a diagonal matrix $W_G \in \mathbb{R}^{m \times m}$, where $W_G(e, e) = w_e$ for any edge $e \in E[G]$. Then, we can write L_G as $L_G = B_G^\intercal W_G B_G$. The *normalized Laplacian matrix* of G is defined by $\mathcal{L}_G \triangleq D_G^{-1/2} L_G D_G^{-1/2} = I - D_G^{-1/2} A_G D_G^{-1/2}$. We sometimes drop the subscript G when the underlying graph is clear from the context.

Spectral sparsification. For any undirected and weighted graph $G = (V, E, w)$, we say a subgraph H of G with proper reweighting of the edges is a $(1 + \varepsilon)$-spectral sparsifier if

$$(1 - \varepsilon)L_G \preceq L_H \preceq (1 + \varepsilon)L_G. \tag{1}$$

By definition, it is easy to show that, if we decompose the edge set of a graph $G = (V, E)$ into $E_1, \ldots, E_\ell$ for a constant ℓ and H_i is a spectral sparsifier of $G_i = (V, E_i)$ for any $1 \leq i \leq \ell$, then the graph formed by the union of edge sets from H_i is a spectral sparsifier of G. It is known that, for any undirected graph G of n vertices, there is a $(1 + \varepsilon)$-spectral sparsifier of G with $O(n/\varepsilon^2)$ edges, and it can be constructed in almost-linear time [12]. We will show that a spectral sparsifier preserves the cluster structure of a graph.

Models of computation. We will study distributed clustering in two models for distributed data: the message passing model and the blackboard model. The message passing model represents those distributed computation systems with point-to-point communication, and the blackboard model represents those where messages can be broadcast to all parties.

More precisely, in the message passing model there are s sites $\mathcal{P}_1, \ldots, \mathcal{P}_s$, and one coordinator. These sites can talk to the coordinator through a two-way private channel. In fact, this is referred to

as the coordinator model in Section 1, where it is shown to be equivalent to the point-to-point model up to small factors. The input is initially distributed at the s sites. The computation is in terms of rounds: at the beginning of each round, the coordinator sends a message to some of the s sites, and then each of those sites that have been contacted by the coordinator sends a message back to the coordinator. At the end, the coordinator outputs the answer. In the alternative blackboard model, the coordinator is simply a blackboard where these s sites $\mathcal{P}_1, \ldots, \mathcal{P}_s$ can share information; in other words, if one site sends a message to the coordinator/blackboard then all the other $s-1$ sites can see this information without further communication. The order for the sites to speak is decided by the contents of the blackboard.

For both models we measure the *communication cost* as the total number of bits sent through the channels. The two models are now standard in multiparty communication complexity (see, e.g., [5, 17, 19]). They are similar to the congested clique model [13] studied in the distributed computing community; the main difference is that in our models we do not post any bandwidth limitations at each channel but instead consider the total number of bits communicated.

3 Distributed graph clustering

In this section we study distributed graph clustering. We assume that the vertex set of the input graph $G = (V, E)$ can be partitioned into k clusters, where vertices in each cluster S are highly connected to each other, and there are fewer edges between S and $V \setminus S$. To formalize this notion, we define the conductance of a vertex set S by $\phi_G(S) \triangleq w(S, V \setminus S)/\mu(S)$. Generalizing the Cheeger constant, we define the *k-way expansion constant* of graph G by $\rho(k) \triangleq \min_{\text{partition } A_1, \ldots, A_k} \max_{1 \le i \le k} \phi_G(A_i)$. Notice that a graph G has k clusters if the value of $\rho(k)$ is small.

Lee et al. [11] relate the value of $\rho(k)$ to $\lambda_k(\mathcal{L}_G)$ by the following higher-order Cheeger inequality:

$$\frac{\lambda_k(\mathcal{L}_G)}{2} \le \rho(k) \le O(k^2)\sqrt{\lambda_k(\mathcal{L}_G)}.$$

Based on this, a large gap between $\lambda_{k+1}(\mathcal{L}_G)$ and $\rho(k)$ implies (i) the existence of a k-way partition $\{S_i\}_{i=1}^k$ with smaller value of $\phi_G(S_i) \le \rho(k)$, and (ii) any $(k+1)$-way partition of G contains a subset with high conductance $\rho(k+1) \ge \lambda_{k+1}(\mathcal{L}_G)/2$. Hence, a large gap between $\lambda_{k+1}(\mathcal{L}_G)$ and $\rho(k)$ ensures that G has *exactly* k clusters.

In the following, we assume that $\Upsilon \triangleq \lambda_{k+1}(\mathcal{L}_G)/\rho(k) = \Omega(k^3)$, as this assumption was used in the literature for studying graph clustering in the centralized setting [16].

Both algorithms presented in the section are based on the following *spectral clustering* algorithm: (i) compute the k eigenvectors $f_1, \ldots, f_k$ of $\mathcal{L}_G$ associated with $\lambda_1(\mathcal{L}_G), \ldots, \lambda_k(\mathcal{L}_G)$; (ii) embed every vertex v to a point in $\mathbb{R}^k$ through the embedding $F(v) = \frac{1}{\sqrt{d_v}} \cdot (f_1(v), \ldots, f_k(v))$; (iii) run k-means on the embedded points $\{F(v)\}_{v \in V}$, and group the vertices of G into k clusters according to the output of k-means.

3.1 The message passing model

We assume the edges of the input graph $G = (V, E)$ are arbitrarily allocated among s sites $\mathcal{P}_1, \cdots, \mathcal{P}_s$, and we use E_i to denote the edge set maintained by site $\mathcal{P}_i$. Our proposed algorithm consists of two steps: (i) every $\mathcal{P}_i$ computes a linear-sized $(1+c)$-spectral sparsifier H_i of $G_i \triangleq (V, E_i)$, for a small constant $c \le 1/10$, and sends the edge set of H_i, denoted by E_i', to the coordinator; (ii) the coordinator runs a spectral clustering algorithm on the union of received graphs $H \triangleq \left(V, \bigcup_{i=1}^k E_i'\right)$. The theorem below summarizes the performance of this algorithm, and shows the approximation guarantee of this algorithm is as good as the provable guarantee of spectral clustering known in the centralized setting [16].

Theorem 3.1. *Let $G = (V, E)$ be an n-vertex graph with $\Upsilon = \Omega(k^3)$, and suppose the edges of G are arbitrarily allocated among s sites. Assume $S_1, \cdots, S_k$ is an optimal partition that achieves $\rho(k)$. Then, the algorithm above computes a partition $A_1, \ldots, A_k$ satisfying $\mathrm{vol}(A_i \triangle S_i) = O\left(k^3 \cdot \Upsilon^{-1} \cdot \mathrm{vol}(S_i)\right)$ for any $1 \le i \le k$. The total communication cost of this algorithm is $\widetilde{O}(ns)$ bits.*

Our proposed algorithm is very easy to implement, and the next theorem shows that the communication cost of our algorithm is optimal up to a logarithmic factor.

Theorem 3.2. *Let G be an undirected graph with n vertices, and suppose the edges of G are distributed among s sites. Then, any algorithm that correctly outputs a constant fraction of a cluster in G requires $\Omega(ns)$ bits of communication. This holds even if each cluster has constant expansion.*

As a remark, it is easy to see that this lower bound also holds for constructing spectral sparsifiers: for any $n \times n$ PSD matrix A whose entries are arbitrarily distributed among s sites, any distributed algorithm that constructs a $(1 + \Theta(1))$-spectral sparsifier of A requires $\Omega(ns)$ bits of communication. This follows since such a spectral sparsifier can be used to solve the spectral clustering problem. Spectral sparsification has played an important role in designing fast algorithms from different areas, e.g., machine learning, and numerical linear algebra. Hence our lower bound result for constructing spectral sparsifiers may have applications to studying other distributed learning algorithms.

3.2 The blackboard model

Next we present a graph clustering algorithm with $\widetilde{O}(n + s)$ bits of communication cost in the blackboard model. Our result is based on the observation that a spectral sparsifier preserves the structure of clusters, which was used for proving Theorem 3.1. So it suffices to design a distributed algorithm for constructing a spectral sparsifier in the blackboard model.

Our distributed algorithm is based on constructing a chain of coarse sparsifiers [14], which is described as follows: for any input PSD matrix K with $\lambda_{\max}(K) \leq \lambda_u$ and all the non-zero eigenvalues of K at least λ_ℓ, we define $d = \lceil \log_2(\lambda_u/\lambda_\ell) \rceil$ and construct a chain of $d + 1$ matrices

$$[K(0), K(1), \ldots, K(d)], \tag{2}$$

where $\gamma(i) = \lambda_u/2^i$ and $K(i) = K + \gamma(i)I$. Notice that in the chain above every $K(i-1)$ is obtained by adding weights to the diagonal entries of $K(i)$, and $K(i-1)$ approximates $K(i)$ as long as the weights added to the diagonal entries are small. We will construct this chain recursively, so that $K(0)$ has heavy diagonal entries and can be approximately by a diagonal matrix. Moreover, since K is the Laplacian matrix of a graph G, it is easy to see that $d = O(\log n)$ as long as the edge weights of G are polynomially upper-bounded in n.

Lemma 3.3 ([14])**.** *The chain* (2) *satisfies the following relations: (1) $K \preceq_r K(d) \preceq_r 2K$; (2) $K(\ell) \preceq K(\ell - 1) \preceq 2K(\ell)$ for all $\ell \in \{1, \ldots, d\}$; (3) $K(0) \preceq 2\gamma(0)I \preceq 2K(0)$.*

Based on Lemma 3.3, we will construct a chain of matrices

$$\left[\widetilde{K}(0), \widetilde{K}(1), \ldots, \widetilde{K}(d)\right] \tag{3}$$

in the blackboard model, such that every $\widetilde{K}(\ell)$ is a spectral sparsifier of $K(\ell)$, and every $\widetilde{K}(\ell + 1)$ can be constructed from $\widetilde{K}(\ell)$. The basic idea behind our construction is to use the relations among different $K(\ell)$ shown in Lemma 3.3 and the fact that, for any $K = B^\intercal B$, sampling rows of B with respect to their leverage scores can be used to obtain a matrix approximating K.

Theorem 3.4. *Let G be an undirected graph on n vertices, where the edges of G are allocated among s sites, and the edge weights are polynomially upper bounded in n. Then, a spectral sparsifier of G can be constructed with $\widetilde{O}(n + s)$ bits of communication in the blackboard model. That is, the chain* (3) *can be constructed with $\widetilde{O}(n + s)$ bits of communication in the blackboard model.*

Proof. Let $K = B^\intercal B$ be the Laplacian matrix of the underlying graph G, where $B \in \mathbb{R}^{m \times n}$ is the edge-vertex incidence matrix of G. We will prove that every $\widetilde{K}(i + 1)$ can be constructed based on $\widetilde{K}(i)$ with $\widetilde{O}(n + s)$ bits of communication. This implies that $\widetilde{K}(d)$, a $(1 + \varepsilon)$-spectral sparsifier of K, can be constructed with $\widetilde{O}(n + s)$ bits of communication, as the length of the chain $d = O(\log n)$.

First of all, notice that $\lambda_u \leq 2n$, and the value of n can be obtained with communication cost $\widetilde{O}(n + s)$ (different sites sequentially write the new IDs of the vertices on the blackboard). In the following we assume that λ_u is the upper bound of $\lambda_{\max}$ that we actually obtained in the blackboard.

Base case of $\ell = 0$: By definition, $K(0) = K + \lambda_u \cdot I$, and $\frac{1}{2} \cdot K(0) \preceq \gamma(0) \cdot I \preceq K(0)$, due to Statement 3 of Lemma 3.3. Let $\oplus$ denote appending the rows of one matrix to another. We

define $B_{\gamma(0)} = B \oplus \sqrt{\gamma(0)} \cdot I$, and write $K(0) = K + \gamma(0) \cdot I = B_{\gamma(0)}^{\intercal} B_{\gamma(0)}$. By defining $\tau_i = b_i^{\intercal} \left(K(0)\right)^{\intercal} b_i$ for each row of $B_{\gamma(0)}$, we have $\tau_i \leq b_i^{\intercal} \left(\gamma(0) \cdot I\right) b_i \leq 2 \cdot \tau_i$. Let $\widetilde{\tau}_i = b_i^{\intercal} \left(\gamma(0) \cdot I\right)^{+} b_i$ be the leverage score of b_i approximated using $\gamma(0) \cdot I$, and let $\widetilde{\tau}$ be the vector of approximate leverage scores, with the leverage scores of the n rows corresponding to $\sqrt{\gamma(0)} \cdot I$ rounded up to 1. Then, with high probability sampling $O(\varepsilon^{-2} n \log n)$ rows of B will give a matrix $\widetilde{K}(0)$ such that $(1-\varepsilon)K(0) \preceq \widetilde{K}(0) \preceq (1+\varepsilon)K(0)$. Notice that, as every row of B corresponds to an edge of G, the approximate leverage scores $\widetilde{\tau}_i$ for different edges can be computed locally by different sites maintaining the edges, and the sites only need to send the information of the sampled edges to the coordinator, hence the communication cost is $\widetilde{O}(n+s)$ bits.

Induction step: We assume that $(1-\varepsilon)K(\ell) \preceq_r \widetilde{K}(\ell) \preceq_r (1+\varepsilon)K(\ell)$, and the coordinator maintains the matrix $\widetilde{K}(\ell)$. This implies that $(1-\varepsilon)/(1+\varepsilon) \cdot K(\ell) \preceq_r 1/(1+\varepsilon) \cdot \widetilde{K}(\ell) \preceq_r K(\ell)$. Combining this with Statement 2 of Lemma 3.3, we have that

$$\frac{1-\varepsilon}{2(1+\varepsilon)} K(\ell+1) \preceq_r \frac{1}{2(1+\varepsilon)} \widetilde{K}(\ell) \preceq K(\ell+1).$$

We apply the same sampling procedure as in the base case, and obtain a matrix $\widetilde{K}(\ell+1)$ such that $(1-\varepsilon)K(\ell+1) \preceq_r \widetilde{K}(\ell+1) \preceq_r (1+\varepsilon)K(\ell+1)$. Notice that, since $\widetilde{K}(\ell)$ is written on the blackboard, the probabilities used for sampling individual edges can be computed locally by different sites, and in each round only the sampled edges will be sent to the coordinator in order for the coordinator to obtain $\widetilde{K}(\ell+1)$. Hence, the total communication cost in each iteration is $\widetilde{O}(n+s)$ bits. Combining this with the fact that the chain length $d = O(\log n)$ proves the theorem. □

Combining Theorem 3.4 and the fact that a spectral sparsifier preserves the structure of clusters, we obtain a distributed algorithm in the blackboard model with total communication cost $\widetilde{O}(n+s)$ bits, and the performance of our algorithm is the same as in the statement of Theorem 3.1. Notice that $\Omega(n+s)$ bits of communication are needed for graph clustering in the blackboard model, since the output of a clustering algorithm contains $\Omega(n)$ bits of information and each site needs to communicate at least one bit. Hence the communication cost of our proposed algorithm is optimal up to a poly-logarithmic factor.

4 Distributed geometric clustering

We now consider geometric clustering, including k-median, k-means and k-center. Let P be a set of points of size n in a metric space with distance function $d(\cdot,\cdot)$, and let $k \leq n$ be an integer. In the k-center problem we want to find a set C ($|C| = k$) such that $\max_{p \in P} d(p, C)$ is minimized, where $d(p, C) = \min_{c \in C} d(p, c)$. In k-median and k-means we replace the objective function $\max_{p \in P} d(p, C)$ with $\sum_{p \in P} d(p, C)$ and $\sum_{p \in P} (d(p, C))^2$, respectively.

4.1 The message passing model

As mentioned, for constant dimensional Euclidean space and a constant $c > 1$, there are algorithms that c-approximate k-median and k-means using $\widetilde{O}(sk)$ bits of communication [3]. For k-center, the folklore parallel guessing algorithms (see, e.g., [8]) achieve a 2.01-approximation using $\widetilde{O}(sk)$ bits of communication.

The following theorem states that the above upper bounds are tight up to logarithmic factors. Due to space constraints we defer the proof to the full version of this paper. The proof uses tools from multiparty communication complexity. We in fact can prove a stronger statement that any algorithm that can differentiate whether we have k points or $k+1$ points in total in the message passing model needs $\Omega(sk)$ bits of communication.

Theorem 4.1. *For any $c > 1$, computing c-approximation for k-median, k-means or k-center correctly with probability 0.99 in the message passing model needs $\Omega(sk)$ bits of communication.*

A number of works on clustering consider *bicriteria* solutions (e.g., [10, 6]). An algorithm is a (c_1, c_2)-approximation ($c_1, c_2 > 1$) if the optimal solution costs W when using k centers, then the

output of the algorithm costs at most $c_1 W$ when using at most $c_2 k$ centers. We can show that for k-median and k-means, the $\Omega(sk)$ lower bound holds even for algorithms with bicriteria approximations. The proof of the following theorem can be found in the full version of this paper.

Theorem 4.2. *For any $c \in [1, 1.01]$, computing $(7.1 - 6c, c)$-bicriteria-approximation for k-median or k-means correctly with probability 0.99 in the message passing model needs $\Omega(sk)$ bits of communication.*

4.2 The blackboard model

We can show that there is an algorithm that achieves an $O(1)$-approximation using $\widetilde{O}(s + k)$ bits of communication for k-median and k-means. Due to space constraints we defer the description of the algorithm to the full version of this paper. For k-center, it is straightforward to implement the parallel guessing algorithm in the blackboard model using $\widetilde{O}(s + k)$ bits of communication.

Theorem 4.3. *There are algorithms that compute $O(1)$-approximations for k-median, k-means and k-center correctly with probability 0.9 in the blackboard model using $\widetilde{O}(sk)$ bits of communication.*

5 Experiments

In this section we present experimental results for spectral graph clustering in the message passing and blackboard models. We will compare the following three algorithms. (1) `Baseline`: each site sends all the data to the coordinator directly; (2) `MsgPassing`: our algorithm in the message passing model (Section 3.1); (3) `Blackboard`: our algorithm in the blackboard model (Section 3.2).

Besides giving the visualized results of these algorithms on various datasets, we also measure the qualities of the results via the *normalized cut*, defined as $\text{ncut}(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i \in [k]} \frac{w(A_i, V \setminus A_i)}{\text{vol}(A_i)}$, which is a standard objective function to be minimized for spectral clustering algorithms.

We implemented the algorithms using multiple languages, including Matlab, Python and C++. Our experiments were conducted on an IBM NeXtScale nx360 M4 server, which is equipped with 2 Intel Xeon E5-2652 v2 8-core processors, 32GB RAM and 250GB local storage.

Datasets. We test the algorithms in the following real and synthetic datasets.

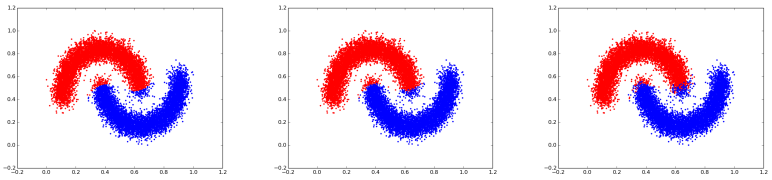
- `Twomoons`: this dataset contains $n = 14,000$ coordinates in $\mathbb{R}^2$. We consider each point to be a vertex. For any two vertices u, v, we add an edge with weight $w(u, v) = \exp\{-\|u - v\|_2^2/\sigma^2\}$ with $\sigma = 0.1$ when one vertex is among the 7000-nearest points of the other. This construction results in a graph with about $110,000,000$ edges.
- `Gauss`: this dataset contains $n = 10,000$ points in $\mathbb{R}^2$. There are 4 clusters in this dataset, each generated using a Gaussian distribution. We construct a complete graph as the similarity graph. For any two vertices u, v, we define the weight $w(u, v) = \exp\{-\|u - v\|_2^2/\sigma^2\}$ with $\sigma = 1$. The resulting graph has about $100,000,000$ edges.
- `Sculpture`: a photo of *The Greek Slave* [1]. We use an 80×150 version of this photo where each pixel is viewed as a vertex. To construct a similarity graph, we map each pixel to a point in $\mathbb{R}^5$, i.e., (x, y, r, g, b), where the latter three coordinates are the RGB values. For any two vertices u, v, we put an edge between u, v with weight $w(u, v) = \exp\{-\|u - v\|_2^2/\sigma^2\}$ with $\sigma = 0.5$ if one of u, v is among the 5000-nearest points of the other. This results in a graph with about $70,000,000$ edges.

In the distributed model edges are randomly partitioned across s sites.

Results on clustering quality. We visualize the clustered results for the `Twomoons` dataset in Figure 1. It can be seen that `Baseline`, `MsgPassing` and `Blackboard` give results of very similar qualities. For simplicity, here we only present the visualization for $s = 15$. Similar results were observed when we varied the values of s.

We also compare the normalized cut (ncut) values of the clustering results of different algorithms. The results are presented in Figure 2. In all datasets, the ncut values of different algorithms are very close. The ncut value of `MsgPassing` slightly decreases when we increase the value of s, while the ncut value of `Blackboard` is independent of s.

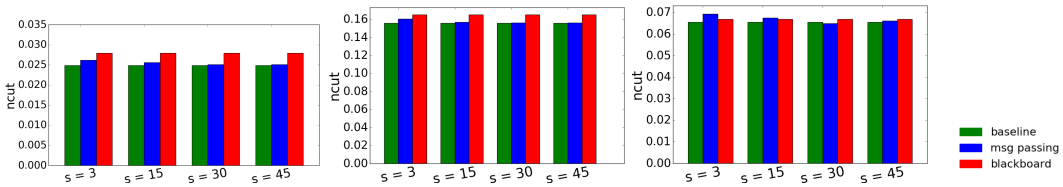
[1] Available in e.g., `http://artgallery.yale.edu/collections/objects/14794`



(a) `Baseline` (b) `MsgPassing` (c) `Blackboard`

`Twomoons`, $k = 2$;

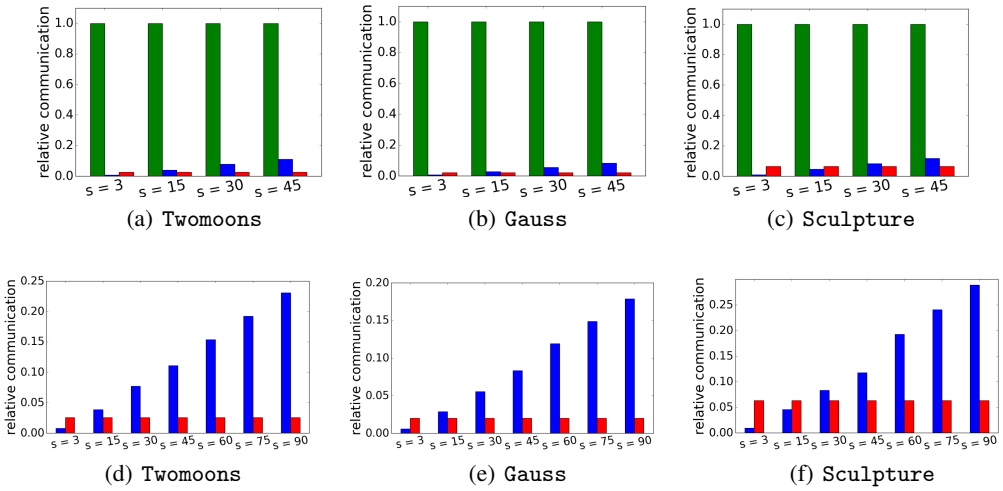
Figure 1: Visualization of the results on `Twomoons`. In the message passing model each site samples $5n$ edges; in the blackboard model all sites jointly sample $10n$ edges and the chain has length 18.



(a) `Twomoons` (b) `Gauss` (c) `Sculpture`

Figure 2: Comparisons on normalized cuts. In the message passing model, each site samples $5n$ edges; in each round of the algorithm in the blackboard model, all sites jointly sample $10n$ edges (in `Twomoons` and `Gauss`) or $20n$ edges (in `Sculpture`) edges and the chain has length 18.

Results on Communication Costs. We compare the communication costs of different algorithms in Figure 3. We observe that while achieving similar clustering qualities as `Baseline`, both `MsgPassing` and `Blackboard` are significantly more communication-efficient (by one or two orders of magnitudes in our experiments). We also notice that the value of s does not affect the communication cost of `Blackboard`, while the communication cost of `MsgPassing` grows almost linearly with s; when s is large, `MsgPassing` uses significantly more communication than `Blackboard`. These confirm our theory.



(a) `Twomoons` (b) `Gauss` (c) `Sculpture`

(d) `Twomoons` (e) `Gauss` (f) `Sculpture`

Figure 3: Comparisons on communication costs. In the message passing model, each site samples $5n$ edges; in each round of the algorithm in the blackboard model, all sites jointly sample $10n$ (in `Twomoons` and `Gauss`) or $20n$ (in `Sculpture`) edges and the chain has length 18.

References

[1] Alexandr Andoni, Jiecao Chen, Robert Krauthgamer, Bo Qin, David P. Woodruff, and Qin Zhang. On sketching quadratic forms. In *ITCS*, pages 311–319, 2016.

[2] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *SODA*, pages 1027–1035, 2007.

[3] Maria-Florina Balcan, Steven Ehrlich, and Yingyu Liang. Distributed k-means and k-median clustering on general communication topologies. In *NIPS*, pages 1995–2003, 2013.

[4] Maria-Florina Balcan, Vandana Kanchanapally, Yingyu Liang, and David P. Woodruff. Improved distributed principal component analysis. *CoRR*, abs/1408.5823, 2014.

[5] Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. A tight bound for set disjointness in the message-passing model. In *FOCS*, pages 668–677, 2013.

[6] Moses Charikar, Samir Khuller, David M. Mount, and Giri Narasimhan. Algorithms for facility location problems with outliers. In *SODA*, pages 642–651, 2001.

[7] Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *STOC*, pages 163–172, 2015.

[8] Graham Cormode, S Muthukrishnan, and Wei Zhuang. Conquering the divide: Continuous clustering of distributed data streams. In *ICDE*, pages 1036–1045, 2007.

[9] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, PCA and projective clustering. In *SODA*, pages 1434–1453, 2013.

[10] Madhukar R. Korupolu, C. Greg Plaxton, and Rajmohan Rajaraman. Analysis of a local search heuristic for facility location problems. In *SODA*, pages 1–10, 1998.

[11] James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multi-way spectral partitioning and higher-order cheeger inequalities. In *STOC*, pages 1117–1130, 2012.

[12] Yin Tat Lee and He Sun. Constructing linear-sized spectral sparsification in almost-linear time. In *FOCS*, pages 250–269, 2015.

[13] Zvi Lotker, Elan Pavlov, Boaz Patt-Shamir, and David Peleg. MST construction in $O(\log \log n)$ communication rounds. In *SPAA*, pages 94–100, 2003.

[14] Gary L. Miller and Richard Peng. Iterative approaches to row sampling. *CoRR*, abs/1211.2713, 2012.

[15] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[16] Richard Peng, He Sun, and Luca Zanetti. Partitioning well-clustered graphs: Spectral clustering works! In *COLT*, pages 1423–1455, 2015.

[17] Jeff M. Phillips, Elad Verbin, and Qin Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. *SIAM J. Comput.*, 45(1):174–196, 2016.

[18] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[19] David P. Woodruff and Qin Zhang. Tight bounds for distributed functional monitoring. In *STOC*, pages 941–960, 2012.