Distributed Information Theoretic Clustering

Pengcheng Shen and Chunguang Li

Abstract—**Distributed data collection and analysis over networks are ubiquitous, especially over the wireless sensor networks (WSNs). Distributed clustering is one of the most important topics in distributed data analysis. It is desired to explore the hidden structure of the data collected/stored in geographically distributed nodes. In recent years, several distributed data clustering techniques have been developed based on the K-means algorithm or the Gaussian mixture model. In these methods, data structures are captured by measures only based on the first and the second order statistics. When the structure of cluster data is complicated, these statistics are insufficient and may lead to unsatisfactory clustering results. In such a case, using information theoretic measures can achieve better clustering performance since they take the whole distribution of cluster data into account. In this work, we incorporate an information theoretic measure into the cost function of the distributed clustering, to present a linear and a kernel distributed clustering algorithms. In the algorithms, each node solves a local clustering problem through diffusion cooperation with its neighboring nodes. In order to preserve privacy and save communication costs, in the cooperation, nodes merely exchange a few parameters instead of original data with their one-hop neighbors. Simulation results show that the proposed distributed algorithms can achieve almost as good clustering results as the corresponding centralized information theoretic clustering algorithms on both synthetic and real data.**

Index Terms—**Diffusion cooperation, discriminative clustering, distributed clustering, information theory, mutual information.**

I. Introduction

DATA clustering is to explore the hidden structure of data and group data items into a few clusters in an unsupervised way. Given the whole data, many (centralized) clustering algorithms have been proposed to solve the unsupervised learning problem [1]–[6]. However, in many cases, large amounts of data are not centrally collected/stored in one source but dispersedly collected/stored in geographically distributed nodes over networks. The widely used wireless sensor network (WSN) is a typical example. Due to the limited energy, communication, computation and storage resources, centralizing the whole distributed data to one fusion node to perform centralized clustering may be impractical. Thus, there is a great demand for distributed data clustering algorithms in which the global clustering problem can be solved at each individual node based on local data and limited information exchanges among nodes. In addition, compared with the centralized clustering, distributed clustering is more flexible and robust to node and/or link failure.

In recent years, several distributed data clustering algorithms have been proposed [7]–[17]. Most of the existing algorithms are based on the K-means method or the Gaussian mixture model (GMM). In K-means based clustering algorithms, the cost functions usually measure the sum of distances (squared differences) between data items and estimated centroids of clusters [7], [9], [17]. In GMM based clustering algorithms, the distribution of an individual cluster is assumed to be Gaussian, which is fully specified by its mean/centroid and variance [12], [13]. In these kinds of methods, data structures are captured by measures only based on the first and the second order statistics. However, real data structures are usually complicated and not in accord with assumed models. In such cases, these clustering algorithms may not achieve satisfactory clustering results.

Information theory provides a general framework to establish clustering criteria. With information theoretic measures (e.g. divergence and mutual information), data structure can be captured beyond the first and the second order statistics, by taking the whole probability distribution function (pdf) of cluster data into consideration. Existing researches have validated the performance improvement brought by introducing information theoretic measures into centralized clustering [18], [19], [21]–[28]. However, to the best of our knowledge, information theory based approaches have not been developed in the field of distributed data clustering yet.

In this work, we present distributed clustering algorithms based on an information theoretic measure. We incorporate the maximum mutual information (MMI) criterion into the cost function in distributed clustering, to present distributed MMI-based (DMMI) clustering algorithms. In our method, each node solves a local clustering problem through cooperation with its single-hop neighboring nodes. By the limited local cooperation/communication, information of local clusters at individual nodes can be gradually diffused over the whole network. Thus each node can utilize global information to help clustering its local data, at a low communication cost. Besides, in the cooperation, nodes do not transmit original data but merely exchange a few parameters of clusters. Hence, clustering task is performed under privacy preservation, which is important in some practical distributed applications [29], [30]. We present both linear and kernel DMMI algorithms. The performance of the proposed algorithms is evaluated on three different synthetic datasets and one real dataset. Simulation re-

Manuscript received March 24, 2014; revised May 17, 2014; accepted May 20, 2014. Date of publication May 29, 2014; date of current version June 17, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Chong-Yung Chi. This work was supported by the National Natural Science Foundation of China (Grant Nos. 61171153 and 61101045), the Zhejiang Provincial Natural Science Foundation of China (Grant LR12F01001), the Open Research Grants by Key Laboratory of Visual Media Intelligent Processing Technology of Zhejiang Province, and the National Program for Special Support of Eminent Professionals. (*Corresponding author: C. Li.*)

The authors are with the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, People's Republic of China (e-mail: equipment777@zju.edu.cn; cgli@zju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2014.2327010

sults show that the proposed distributed algorithms can achieve almost as good clustering results as the centralized MMI-based clustering algorithms.

Compared with our previous work [36], the problem being considered in this paper is basically different. Specifically, we deal with an unsupervised learning problem in this paper, while in the previous publication we consider a supervised learning problem. For the new kind of distributed learning problem, we come up with new cost functions and optimization solutions which can meet specific demands of the distributed clustering, though some optimization techniques used in the two papers are similar. The choice of using mutual information (based on discriminative clustering functions) in the distributed cost function is deliberate (a detailed discussion is provided in Section II).

The rest of this paper is organized as follows. In Section II, to make the paper self-contained, we briefly introduce some centralized clustering algorithms based on different information theoretic measures. Besides, we interpret the motivation for choosing the MMI criterion to develop distributed clustering algorithms. Afterwards, in Section III, we present the distributed MMI-based (DMMI) clustering algorithms in detail, including types of linear DMMI and kernel DMMI. Results of numerical simulations are shown in Section IV to illustrate the effectiveness and advantages of the proposed algorithms. Finally, conclusions are drawn in Section V.

Notation: In this paper, we use boldface and normal letters to denote the vectors and scalars, respectively. We use decorated letters or the notation $\{\cdot\}$ to denote a set. Besides, superscript $\hat{(\cdot)}$ denotes an estimator, superscript $(\cdot)^T$ denotes transposition, and $|\cdot|$ stands for the set cardinality. Other notations will be introduced if necessary.

II. Preliminaries and Motivation

In the existing researches on centralized clustering, there are various well-performing algorithms based on different information theoretic measures [18], [20], [22], [23], [25], [28]. The two most frequently-used measures in corresponding literatures are divergence and mutual information [31].

For the divergence-based clustering, there are roughly two types of algorithms, which are the parametric type and the nonparametric type, respectively. The Bregman soft clustering algorithm is a representative and typical sample for the former [20]. In [20], the authors model the data source with a mixture of exponential family distributions (one component for one cluster), and pose the clustering problem as a parameter estimation problem for the mixture model. They find the correspondence between exponential families and regular Bregman divergences, and thereby bring up a Bregman divergence viewpoint for learning the maximum likelihood parameters of the mixture model. The algorithm provides a framework for clustering different datasets by using different Bregman divergences (or equivalently, parametric models of different exponential distributions). For a given application (dataset), to obtain good clustering performance, it is expected to artificially choose a specific Bregman divergence (or equivalently, parametric model of a specific exponential distribution) which matches the generative model of current data. However, the prior knowledge for the generative models of real datasets can be lacking, which makes it hard to choose an appropriate parametric model. In this case, using nonparametric models is more flexible and applicable. The algorithm proposed in [18] is a typical sample for the nonparametric divergence-based clustering. In the literature, the authors use divergence to measure the ‘distance’ between distributions of data belonging to different clusters. For a clustering result, large divergence means there are obvious differences or boundaries between data items belonging to different clusters. Hence, their goal is to maximize the divergence, by adjusting the assignment of cluster/class label on each data item. In this kind of method, calculating divergence relies on unknown *conditional pdfs of cluster data*, $p(\mathrm{data}|\mathrm{cluster\ label})$, which need to be estimated during clustering. In order to make clustering adaptable to datasets of different data structures, the authors choose to directly estimate the conditional pdfs from labeled data in a *nonparametric* manner [32], [33], rather than to model them by predefined parametric models, e.g. exponential family distributions. Accordingly, the optimization of corresponding cost functions are directly related to the cluster label of each data item. Note that, when the algorithms are extended to the distributed clustering field, this characteristic would lead to request for transmission of original data (it may be not necessary under some kind of modification, however, we have not yet found an efficient modification scheme to avoid the transmission of data while maintaining good clustering performances). As mentioned in the introduction, privacy preservation and communication resource saving (original data can be large) are usually important in real distributed applications. So, directly transmitting original data is not preferred. Fortunately, this does not have to be the case for mutual information-based distributed clustering, as explained below.

As for mutual information, in the context of clustering, it can be used to measure the information shared by data items and cluster labels. In more detail, it measures the uncertainty about cluster labels reduced by knowing the data items, or the uncertainty about data items reduced by knowing the corresponding cluster labels. Large mutual information means that the structure information contained in data items is well preserved by the clustering result. Hence, MMI-based clustering algorithms seek the clustering result that maximizes the mutual information. Calculation of mutual information can be performed based on conditional distributions of cluster data, $p(\mathrm{data}|\mathrm{cluster\ label})$, or based on *discriminative clustering functions*, $f(\mathrm{cluster\ label}|\mathrm{data})$ (we show the detail in the next section) [28]. Using the former type would make the algorithm face the same problems as the above-mentioned divergence-based clustering methods, which focus on modeling the distributions of cluster data (so does the GMM-based clustering algorithms). In comparison, the discriminative clustering functions do not directly model the cluster data, but only make assumptions on the boundary between clusters. The distributions of cluster data could be complicated, while the boundaries among clusters might be a simple curve. Hence, the discriminative clustering functions can be modeled in a *parametric* manner without losing much applicability of algorithms on different datasets. Accordingly, the cost functions can be optimized by adjusting a few parameters, rather than the cluster labels of all data items. When decentralizing the

MMI-based clustering algorithms, this characteristic makes it possible to avoid transmitting original data, by transmitting limited number of parameters instead. The number of data items is usually much larger than the number of parameters used in a parametric model.

In this paper, we want to develop distributed information theoretic clustering algorithms which are applicable to different datasets while avoiding transmitting original data. As discussed above, the MMI criterion based on discriminative clustering functions satisfies the demands, thus it is a good choice for developing the distributed clustering algorithms. In the next section, we describe and formulate the new algorithms in detail.

III. Distributed MMI-Based Clustering

In this part, we first state the distributed clustering problem in mathematics, and then we formulate the framework for distributed MMI-based clustering. Afterwards, we present the derivations of linear DMMI and kernel DMMI algorithms, respectively.

A. Problem Formulation

We considered a network composed of J nodes distributed over a geographic region. Every node j collects/stores a set of data items denoted by $\mathcal{X}_j = \{\boldsymbol{x}_{j,n}, n = 1, \ldots, N_j\}$, where $\boldsymbol{x}_{j,n} = (x_{j,n,1}, \ldots, x_{j,n,D})^T \in \mathbb{R}^D$ are D-dimensional data items, or named by feature vectors, with components $x_{j,n,d}$. For each node j, the N_j data items, $\{\boldsymbol{x}_{j,n}, n = 1, \ldots, N_j\}$, are considered to be samples of a random variable $\boldsymbol{X}_j$ with probability measure $p(\boldsymbol{x}_j)$, and the random variables $\{\boldsymbol{X}_j, j = 1, \ldots, J\}$ are supposed to follow the same probability measure. In other words, the N_j data items, $\{\boldsymbol{x}_{j,n}, n = 1, \ldots, N_j\}$, can also be viewed as part of samples of a global random variable $\boldsymbol{X}$ with probability measure $p(\boldsymbol{x})$. The total number of data samples for $\boldsymbol{X}$ over the whole network is $N = \sum_j N_j$. Without loss of generality, we model the network by a connected graph $\mathcal{G}(\mathcal{J}, \mathcal{E})$, where $\mathcal{J}$ denotes the node set and $\mathcal{E}$ denotes the edge set [7]. If two nodes are connected by an edge, then they are the one-hop-communication neighbor for each other. All the one-hop neighbors of node j and itself constitute its neighbor set $\mathcal{B}_j$. Node j is supposed to cluster its local data into M different classes based on cooperation with nodes belonging to $\mathcal{B}_j$. In other words, each data item $\boldsymbol{x}_{j,n}$ stored in node j needs to be attached with a class label. The class label K is also considered to be a random variable with probability measure $p(k), k \in \{1, \ldots, M\}$.

Note that though direct cooperation is limited within one-hop neighbors, in a connected graph, information shared by one node can still be diffused over the whole network in the following steps. Thus each node actually can utilize global information in its local clustering, which makes it possible that distributed clustering algorithms achieve as good clustering results as the corresponding centralized clustering algorithms.

In centralized MMI-based clustering, with the whole N data items available, $\{\boldsymbol{x}_l, l = 1, \ldots, N\}/\{\boldsymbol{x}_{j,n}, j = 1, \ldots, J, n = 1, \ldots, N_j\}$ (note that $\{\boldsymbol{x}_l\}$ and $\{\boldsymbol{x}_{j,n}\}$ are two different notations for one same dataset of N data items), algorithms seek the global clustering solution by maximizing the mutual information between data $\boldsymbol{X}$ and class label K, as shown below [28]

$$\begin{aligned}\max_{\mathcal{W}} J^{cent}(\mathcal{W}) &= I_{\mathcal{W}}(\boldsymbol{X}; K) \\ &= H_{\mathcal{W}}(K) - H_{\mathcal{W}}(K|\boldsymbol{X}), \qquad (1)\end{aligned}$$

where $I_{\mathcal{W}}(\cdot)$,$H_{\mathcal{W}}(\cdot)$ respectively denote functions calculating mutual information and entropy, and $\mathcal{W}$ is the set of parameters modeling the conditional model $p(k|\boldsymbol{x}; \mathcal{W})$, which is also called the discriminative clustering model. There are many standard discriminative functions can be used for $p(k|\boldsymbol{x}; \mathcal{W})$, such as the logistic regression.

To decentralize the above cost function, we approximate the empirical estimate of the global mutual information

$$I_{\mathcal{W}}^{global}(\boldsymbol{X}; K) = \int\limits_{\boldsymbol{x}} \sum_{k=1}^{M} p(\boldsymbol{x}) p(k|\boldsymbol{x}) \log \frac{p(k|\boldsymbol{x})}{p(k)} \mathrm{d}\boldsymbol{x}$$

as

$$\begin{aligned}&\hat{I}_{\mathcal{W}}^{global}(\boldsymbol{X}; K) \\ &= \sum_{l=1}^{N} \sum_{k=1}^{M} \frac{1}{N} p(k|\boldsymbol{x}_l) \log \frac{p(k|\boldsymbol{x}_l)}{\hat{p}(k)} \\ &= \sum_{j=1}^{J} \sum_{n=1}^{N_j} \sum_{k=1}^{M} \frac{1}{\sum_j N_j} p(k|\boldsymbol{x}_{j,n}) \log \frac{p(k|\boldsymbol{x}_{j,n})}{\hat{p}(k)} \\ &= \sum_{j=1}^{J} \frac{N_j}{\sum_j N_j} \left(\frac{1}{N_j} \sum_{n=1}^{N_j} \sum_{k=1}^{M} p(k|\boldsymbol{x}_{j,n}) \log \frac{p(k|\boldsymbol{x}_{j,n})}{\hat{p}(k)} \right) \\ &\approx \sum_{j=1}^{J} \frac{N_j}{\sum_j N_j} \hat{I}_{\mathcal{W}}^{local}(\boldsymbol{X}_j; K), \qquad (2)\end{aligned}$$

where $\hat{p}(k) = \frac{1}{N} \sum_{l=1}^{N} p(k|\boldsymbol{x}_l)$ is an empirical distribution of class labels based on the whole data items. The approximation in the formula comes up for the reason that when estimating local mutual information

$$\hat{I}_{\mathcal{W}}^{local}(\boldsymbol{X}_j; K) = \frac{1}{N_j} \sum_{n=1}^{N_j} \sum_{k=1}^{M} p(k|\boldsymbol{x}_{j,n}) \log \frac{p(k|\boldsymbol{x}_{j,n})}{\hat{p}_j(k)},$$

actually only local data can be used in calculating the empirical distribution of class labels

$$\hat{p}_j(k) = \frac{1}{N_j} \sum_{n=1}^{N_j} p(k|\boldsymbol{x}_{j,n}).$$

This approximation is relatively accurate if the whole data items are randomly (roughly uniformly) distributed in different nodes. The reason is that under this assumption, the distributions of local data at individual nodes would be similar to the distribution of global data, thus the local empirical distributions of class labels $\hat{p}_j(k)$ would also be similar to the global empirical distribution $\hat{p}(k)$. Even if this assumption is invalid in some cases, though the accuracy of the approximation decreases, the performance of the distributed algorithms based on this approximation would not degrade much, as shown and explained in Section IV-C. By this approximation, we have transformed the global mutual information into a linear combination of local mutual information. In the following, we come up with the cost function of information theoretic distributed clustering based on the above results.

B. Algorithms

In order to maintain a "global-like" cost function in distributed clustering while meeting relevant limitation requirements, we propose that each node j seeks the optimal clustering solution by maximizing a linear combination of local mutual information within its neighbor set $\mathcal{B}_j$, as below

$$\max_{\mathcal{W}} J_j^{loc}(\mathcal{W}) = \sum_{l\in\mathcal{B}_j} c_{l,j} \hat{I}_{\mathcal{W}}^{local}(\boldsymbol{X}_l; K), \tag{3}$$

where $\{c_{l,j}\}$ are some non-negative combination coefficients satisfying the condition $\sum_{l\in\mathcal{B}_j} c_{l,j} = 1$, $c_{l,j} = 0$ if $l \notin \mathcal{B}_j$. Compared with (2), here the nodes being considered are restricted within the neighbor set, to guarantee a local cooperation. Accordingly, the combination coefficients $\{c_{l,j}\}$ are set as $\left\{ \frac{N_l}{\sum_{l\in\mathcal{B}_j} N_l} \right\}$. For the sake of simplicity, we replace the notation $\hat{I}_{\mathcal{W}}^{local}(\boldsymbol{X}_j; K)$ by $\hat{I}_j(\boldsymbol{X}_j; K)$ in the following derivation, in the case of no confusion.

Having determined the cost function for each node, we further consider how to solve the optimization problem without transmitting original data. Usually, the cost function (3) can be maximized by a certain iterative scheme

$$\mathcal{W}_{j,i} = \mathcal{W}_{j,i-1} + \Delta\mathcal{W}_{j,i}, \tag{4}$$

where the subscript i denotes the iteration step, set $\mathcal{W}_{j,i}$ denotes the guess of model parameters for node j at step i, and $\Delta\mathcal{W}_{j,i}$ denotes the increment of parameters for the ith adjustment (here, operations on a set are performed by operating on each element/parameter in the set). There are many gradient-based methods to calculate the increment $\Delta\mathcal{W}_{j,i}$. For example, by using the steepest-ascent method, we have

$$\Delta\mathcal{W}_{j,i} = \mu \sum_{l\in\mathcal{B}_j} c_{l,j} \frac{\partial \hat{I}_l(\boldsymbol{X}_l; K)}{\partial \mathcal{W}} |_{\mathcal{W}_{j,i-1}}, \tag{5}$$

where μ is a learning step-size. Note that in the above calculation of the gradient at node j, we have to use the original data stored in all neighboring nodes, which is infeasible under the restriction on data transmission. We solve this problem by introducing an assumption which has been widely used in the field of distributed parameter estimation [35]–[39]. That is, the local parameter estimate at node j is assumed to be a linear combination of these estimates:

$$\mathcal{W}_{j,i} = \sum_{l\in\mathcal{B}_j} c_{l,j} \mathcal{V}_{l,i}, \tag{6}$$

where $\mathcal{V}_{l,i}$ denotes the intermediate estimates of model parameters offered by neighboring node l at ith iteration step. These intermediate estimates are adjusted merely based on local data, e.g. $\Delta\mathcal{V}_{l,i} = \mu \frac{\partial \hat{I}_l(\boldsymbol{X}_l;K)}{\partial \mathcal{V}}$. Since the linear combination assumption also holds for step $i-1$, we have

$$\mathcal{W}_{j,i-1} = \sum_{l\in\mathcal{B}_j} c_{l,j} \mathcal{V}_{l,i-1}. \tag{7}$$

By combining (6) with (7), we have

$$\Delta\mathcal{W}_{j,i} = \sum_{l\in\mathcal{B}_j} c_{l,j} \Delta\mathcal{V}_{l,i}. \tag{8}$$

This means that the increments also follow the linear combination assumption. Based on (6) and (8), we can easily decompose the iterative scheme (4) into a two-step iteration

$$\begin{cases} \mathcal{V}_{j,i} = \mathcal{V}_{j,i-1} + \Delta\mathcal{V}_{j,i}, & (a) \\ \mathcal{W}_{j,i} = \sum_{l\in\mathcal{B}_j} c_{l,j} \mathcal{V}_{l,i}. & (b) \end{cases} \tag{9}$$

In the new iterative scheme, nodes first adjust the intermediate estimates based on their local data, respectively, then they transmit the intermediate estimates to their neighboring nodes, finally each node calculates its local estimates by fusing all the available intermediate estimates. Thus the optimization problem is solved by merely transmitting parameters instead of the original data. Note that the obtained two-step iterative scheme is similar to the "adaption-then-combination" (ATC) scheme used in distributed estimation [34], [36]. The corresponding CTA("combination-then-adaption")-like scheme can be obtained in a similar way. Since there is no big difference between these two kinds of schemes and the ATC scheme usually leads to slightly better performances than the CTA scheme, in this paper, we only focus on the ATC-like scheme.

Equation (9) provides a framework for distributed MMI-based clustering. In the formula, we have not yet referred to a specific discriminative model $p(k|\boldsymbol{x}; \mathcal{W})$ or a specific method in calculating the increment $\Delta\mathcal{V}_{j,i}$, so the results obtained are general. In the next two subsections, we derive the linear DMMI and kernel DMMI algorithms based on the proposed general framework, using a linear discriminative model and a kernel discriminative model, respectively. Besides, we use the simple steepest-ascent method to calculate the increment $\Delta\mathcal{V}_{j,i}$ in the algorithms. Other gradient-based optimization methods may also work, but the steepest-ascent method is efficient and has advantage in computational complexity over other methods, which is an important merit for nodes with limited resource in computation and storage.

1) Linear DMMI: In this part, we model the discriminative clustering model $p(k|\boldsymbol{x}; \mathcal{W})$ by the multi-class logistic regression function

$$p(k|\boldsymbol{x}; \mathcal{W}) \propto \exp\left(\boldsymbol{\varphi}_k^T \boldsymbol{x} + b_k\right), \tag{10}$$

where b_k is a scalar and $\boldsymbol{\varphi}_k \in \mathbb{R}^D$ is a D-dimensional parameter vector with component φ_{kd}. In this model, the parameter set $\mathcal{W} = \{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_M; b_1, \ldots, b_M\}$ specifies M hyperplanes. Each hyperplane is supposed to split one cluster from the others. So, this model is applicable to linearly separable problems.

Given the discriminative model, we calculate $\Delta\mathcal{V}_{j,i} = \{\Delta\tilde{\boldsymbol{\varphi}}_{1,j,i}, \ldots, \Delta\tilde{\boldsymbol{\varphi}}_{M,j,i}; \Delta\tilde{b}_{1,j,i}, \ldots, \Delta\tilde{b}_{M,j,i}\}$ by taking the derivatives of $\hat{I}_j(\boldsymbol{X}_j; K)$ ($\tilde{\boldsymbol{\varphi}}$ and $\tilde{b}$ have the same meaning as $\boldsymbol{\varphi}$ and b, respectively. We use tilde upon $\boldsymbol{\varphi}$ and b to distinguish the intermediate quantities $\mathcal{V}$ from the fused quantities $\mathcal{W}$):

$$\begin{aligned} \Delta\tilde{\varphi}_{k,d,j,i} &= \mu \frac{\partial \hat{I}_j(\boldsymbol{X}_j;K)}{\partial \tilde{\varphi}_{k,d}} |_{\mathcal{W}_{j,i-1}} \\ &= \frac{\mu}{N_j} \sum_{n=1}^{N_j} x_{j,n,d} p_{j,k,n} \\ &\quad \cdot \left(\log \frac{p_{j,k,n}}{\hat{p}_{j,k}} - \sum_{c=1}^{M} p_{j,c,n} \log \frac{p_{j,c,n}}{\hat{p}_{j,c}} \right) |_{\mathcal{W}_{j,i-1}}, \end{aligned} \tag{11}$$

$$\begin{aligned} \Delta\tilde{b}_{k,j,i} &= \mu \frac{\partial \hat{I}_j(\boldsymbol{X}_j;K)}{\partial \tilde{b}_{k}} |_{\mathcal{W}_{j,i-1}} \\ &= \frac{\mu}{N_j} \sum_{n=1}^{N_j} p_{j,k,n} \\ &\quad \cdot \left(\log \frac{p_{j,k,n}}{\hat{p}_{j,k}} - \sum_{c=1}^{M} p_{j,c,n} \log \frac{p_{j,c,n}}{\hat{p}_{j,c}} \right) |_{\mathcal{W}_{j,i-1}}, \end{aligned} \tag{12}$$

where $p_{j,k,n}$ ($p_{j,c,n}$) is the shorthand notation for $p_j\,(k|\boldsymbol{x}_{j,n}; \mathcal{V})$ ($p_j(c|\boldsymbol{x}_{j,n}; \mathcal{V})$), and $\hat{p}_{j,c}$ ($\hat{p}_{j,k}$) is the shorthand notation for $\hat{p}_j(c) = \frac{1}{N_j}\sum_n p_j(c|\boldsymbol{x}_{j,n}; \mathcal{V})$ ($\hat{p}_j(k)$). These four terms are functions of the model parameters, and all need to be updated at each iteration. Since the derivation is similar to that in the centralized case, we just present the final results here. Readers can refer to literature [28] for more details. Note that in the formulae above, derivatives are calculated at the last fused estimates $\mathcal{W}_{j,i-1}$ rather than the last intermediate estimates $\mathcal{V}_{j,i-1}$. The reasons are, in our iterative scheme, the fused estimates $\mathcal{W}_{j,i-1}$ are the final guess of parameters at the $(i-1)$th iteration step, thus naturally should be the initial guess for the next iteration step. Besides, $\mathcal{W}_{j,i-1}$ contains more latest clustering information of neighboring nodes than $\mathcal{V}_{j,i-1}$, which is beneficial to the local clustering at node j.

In each iteration, calculating the increment $\Delta\mathcal{V}_{j,i}$ needs $O(N_j M D)$ operations, since there are $M(D+1)$ parameters and computing the gradient for each parameter requires one sum over N_j data items (the term $\sum_{c=1}^{M} p_{j,c,n} \log \frac{p_{j,c,n}}{\hat{p}_{j,c}}$ can be computed once and reused for parameters of different classes, as the authors put in [28]). Besides, combining all the intermediate estimates of neighboring nodes requires $O(|\mathcal{B}_j|)$ operations. So, for each node, the computational complexity of each iteration is $O(N_j M D + |\mathcal{B}_j|)$.

As for the communication cost, in each iteration, every node needs to transmit $M(D+1)$ parameters to its $|\mathcal{B}_j|$ neighbors. Usually, the number of one-hop neighbors is limited in many real networks and the expected number of classes is small for general clustering problems, hence, the communication cost is moderate.

For clarity, the pseudo-code of this algorithm is summarized in Algorithm 1, where the maximum number of iterations is denoted as T.

Algorithm 1 linear DMMI algorithm

```
Input: Data items $\{\boldsymbol{x}_{j,n}\}$, desired number of classes $M$.
Initialization: Initialize $\mathcal{V}_{j,0}$ and $\mathcal{W}_{j,0}$ for each node $j$.
for $i = 1$: $T$
  for $j = 1$: $J$
    Compute $\{\Delta\tilde{\varphi}_{k,d,j,i}\}$ via (11).
    Compute $\{\Delta\tilde{b}_{k,j,i}\}$ via (12).
    Compute $\mathcal{V}_{j,i}$ via (9a).
  end for
  for $j = 1$: $J$
    Broadcast $\mathcal{V}_{j,i}$ to all neighbors in $\mathcal{B}_j$.
  end for
  for $j = 1$: $J$
    Compute $\mathcal{W}_{j,i}$ via (9b).
  end for
end for
```

2) Kernel DMMI: Since real datasets are not always linearly separable, the linear DMMI algorithm presented above may not work well when the boundaries between different clusters are complicated. In the centralized MMI-based clustering algorithms, this problem can be solved by utilizing the kernel multi-logit regression in forming the discriminative clustering model [28]:

$$p(k|\boldsymbol{x}; \mathcal{W}) \propto \exp\left(\sum_n \alpha_{k,n} G(\boldsymbol{x}_n, \boldsymbol{x}) + b_k\right), \tag{13}$$

where $G(\cdot,\cdot)$ is a positive kernel function which evaluates the inner product of two vectors in a high-dimensional space. In this model, the (global) discriminative clustering surfaces are specified by all the data items $\{\boldsymbol{x}_n\}$ with the corresponding weight coefficients $\{\alpha_{k,n}\}$ and bias coefficients $\{b_k\}$. Note that in the distributed clustering case, if we use the same model, then the whole data items need to be available for each node, which is infeasible without transmission of data. However, if using an approximate model based on local data for each node j, the acting objects of weight coefficients $\{\alpha_{k,n,j}\}$ will vary with nodes since the available local data items are different for different nodes. Thus we can not directly fuse weight coefficients offered by different neighboring nodes. This situation conflicts with the proposed framework of DMMI clustering.

In order to eliminate the conflict, we use a modified kernel discriminative model

$$p(k|\boldsymbol{x}; \mathcal{W}) \propto \exp\left(\sum_{h=1}^{L} \alpha_{k,h} G(\boldsymbol{\chi}_h, \boldsymbol{x}) + b_k\right), \tag{14}$$

where $\boldsymbol{\chi_h} \in \mathbb{R}^D$ is a D-dimensional base vector. The set of base vectors $\{\boldsymbol{\chi}_h\}$ is constrained to be the same for all nodes. By this modification, the weight coefficients $\{\alpha_{k,h,j}\}$ of different nodes share the common acting-objects and thus could be directly fused among neighboring nodes.

Now, the problem left is choosing the base vectors. In literature [40], the authors suggested several feasible approaches to design the L base vectors, including the grid-based design and the random design. In the former approach, grid points in the value range of data are chosen as the base vectors, while in the latter approach, base vectors are randomly sampled from the value range of data. Both of the two methods are suitable for the kernel DMMI. The appropriate value of L depends on specific problems. Intuitively, it increases with the complexity of between-cluster boundaries and the dimension of data. Existing researches on distributed kernel support vector machines (SVM) show that the value of L does not have to be large for general classification problems [40]. Besides, our following simulation results indicate that, for low-dimension clustering problems, a few number of base vectors is enough to obtain satisfactory clustering results.

After determining the kernel discriminative model, we can obtain the increment $\Delta\mathcal{V}_{j,i} = \{\{\Delta\tilde{\alpha}_{k,h,j,i}\};\{\Delta\tilde{b}_{k,j,i}\}\}$ in a similar way as that in the linear case. Since the formula of calculating $\{\Delta\tilde{b}_{k,j,i}\}$ is totally the same as (12), we do not present it here repeatedly, for the sake of simplicity. The calculation of $\{\Delta\tilde{\alpha}_{k,h,j,i}\}$ is given by

$$\begin{aligned}\Delta\tilde{\alpha}_{k,h,j,i} &= \mu\frac{\partial\hat{I}_j(\boldsymbol{X}_j;K)}{\partial\tilde{\alpha}_{k,h}}|_{\mathcal{W}_{j,i-1}}\\ &= \frac{\mu}{N_j}\sum_{n=1}^{N_j}K(\boldsymbol{\chi}_h,\boldsymbol{x}_{j,n})p_{j,k,n}\\ &\quad\cdot\left(\log\frac{p_{j,k,n}}{\hat{p}_{j,k}}-\sum_{c=1}^{M}p_{j,c,n}\log\frac{p_{j,c,n}}{\hat{p}_{j,c}}\right)|_{\mathcal{W}_{j,i-1}}. \quad (15)\end{aligned}$$

Next, we consider the computational complexity and communication cost of the kernel DMMI. Since the kernel model contains $M(L+1)$ parameters and each parameter introduces $O(N_j)$ operations in computing its corresponding gradient, the computational complexity in calculating the increment $\Delta\mathcal{V}_{j,i}$ is $O(N_jML)$. Combining with the $O(|\mathcal{B}_j|)$ operations required in fusing parameter estimates, the total computational complexity of one iteration is $O(N_jML+|\mathcal{B}_j|)$ for each node. Besides, we can easily find that for each node, the communication cost per iteration is $O(ML|\mathcal{B}_j|)$. As we have mentioned previously, the value of L usually does not have to be large, thus the cost burdens of computation and communication would not be heavy.

For clarity, the pseudo-code of this algorithm is summarized in Algorithm 2.

Algorithm 2 kernel DMMI algorithm

```
**Input**: Data items $\{\boldsymbol{x}_{j,n}\}$, desired number of classes $K$.
**Initialization**: Initialize $\mathcal{V}_{j,0}$ and $\mathcal{W}_{j,0}$ for each node $j$.
for $i = 1: T$
  for $j = 1: J$
    Compute $\{\Delta\tilde{\alpha}_{k,h,j,i}\}$ via (15).
    Compute $\{\Delta\tilde{b}_{k,j,i}\}$ via (12).
    Compute $\mathcal{V}_{j,i}$ via (9a).
  end for
  for $j = 1: J$
    Broadcast $\mathcal{V}_{j,i}$ to all neighbors in $\mathcal{B}_j$.
  end for
  for $j = 1: J$
    Compute $\mathcal{W}_{j,i}$ via (9b).
  end for
end for
```

IV. Simulation Results

In this section, we conduct a series of simulations to evaluate the performance of the linear DMMI and the kernel DMMI. We compare the results of the proposed algorithms with those of the centralized MMI-based algorithms, the centralized K-means, and the non-cooperative MMI-based algorithms (in which nodes do not exchange any information with other nodes). Note that since the distributions of cluster data in the considered datasets are all non-Gaussian and the selected algorithms could already provide comprehensive comparison in evaluating the proposed DMMI algorithms, we do not present the results of GMM-based clustering algorithms in this paper. In the simulations, we let all the MMI-based algorithms get initial guesses for parameters based on K-means method (for distributed cases, every node performs the K-means initialization respectively on its own data), according to the procedure outlined in [28]. For the distributed iterative algorithms, in this paper, we use a simple distributed termination criterion that all nodes have the same maximum number of iteration T. We set the value of T empirically. We found in the simulations that these algorithms can converge within a few hundred iterations. Besides, if needed, we can use a more complicated distributed termination criterion similar to that used in [43] to make the DMMI algorithms more adaptive.

We consider a network composed of 20 nodes, which are randomly distributed in a region. Unless otherwise stated, we let each node connect to its nearest 4 nodes, and then randomly add some long-range connections with a probability of 0.1. These settings are similar to those used in the literature on distributed information theoretic estimation [36].

The simulations are organized as follows. In Section IV-A and Section IV-B, we evaluate the performance of the DMMIs on synthetic data and real data, respectively. In Section IV-C, we further study the performance of the DMMIs under cases of unbalanced data distribution.

A. Synthetic Data

In this subsection, we test performances of algorithms on three different synthetic datasets, including a linearly separable dataset and two linearly non-separable datasets, by Monte Carlo simulations. Similar datasets are frequently used to evaluate the performances of clustering algorithms in corresponding literatures [18], [19], [41], [42]. For each dataset, we generate data randomly from respective data models in every simulation. These data are randomly grouped into 20 subsets with the same number of data items per subset. These data subsets are then uniformly allocated to 20 nodes. The category of data inherently determined by its generation model is viewed as the ground truth in evaluating clustering results. For the kernel DMMI, Gaussian kernel is employed, and base vectors are designed by the grid-based method. The value of L is set to be 64 for all datasets. Note that though for the linearly separable dataset, a smaller L can still lead to good clustering results, using a unified setting for L avoids the case-by-case considerations. Moreover, it shows that the DMMI algorithms are robust to the choice of L, to some extent.

Firstly, we show some qualitative results to illustrate the performance of different algorithms. We present some examples

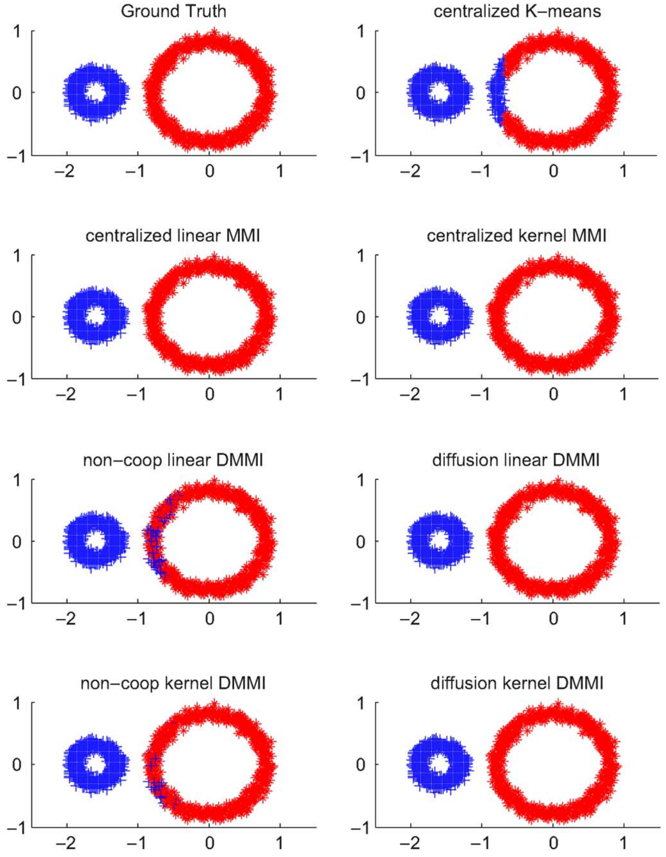


Fig. 1. Examples of clustering results for comparing different algorithms on a linearly separable dataset.

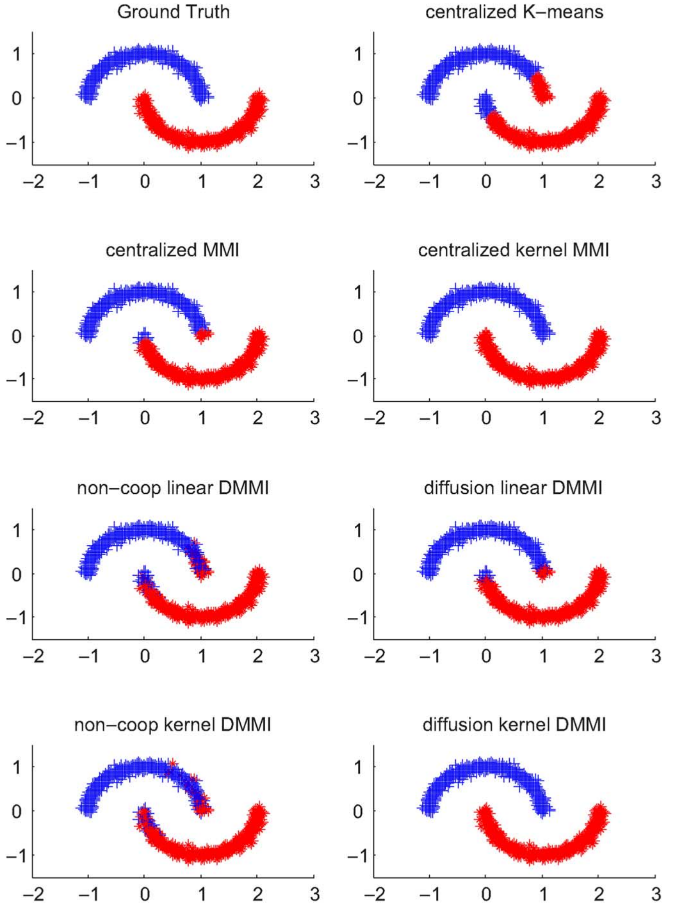


Fig. 2. Examples of clustering results for comparing different algorithms on the “double-moon” dataset.

of clustering results in the simulations for a visual comparison. Fig. 1 and Fig. 2 show examples of clustering results of different algorithms on the two-class linearly separable dataset and the linearly non-separable dataset, respectively. Fig. 3 depicts the examples of clustering results for a three-class nonlinearly separable problem.

Secondly, we present some quantitative results to further compare the performance of different algorithms. For each dataset, we calculate the average number of misclassification samples over 50 independent Monte Carlo simulations. In each simulation, the total numbers of samples used for the three datasets are 2000, 2000, 3000, respectively, with 1000 samples per class. The statistical results are listed in the Table I.

Combining the qualitative results with the quantitative results, we find that the centralized K-means method fails to cluster all these datasets to desired results, since the data structures are too complicated to be captured by the first and the second order statistics. The non-cooperative DMMI algorithms, neither linear nor kernel, also can not well cluster these datasets, which indicates that there is a need for inter-node information exchanges when local data lacks the global information of data distributions. In comparison, the centralized linear MMI algorithm and the linear DMMI algorithm with the diffusion cooperation obtain satisfactory clustering results for the linearly separable dataset, and the centralized kernel MMI algorithm and the kernel DMMI algorithm with the diffusion cooperation achieve good clustering results for all the datasets. Basically, the diffusion-based linear DMMI and kernel DMMI show comparable performance to the centralized linear MMI and kernel MMI algorithms, respectively.

Besides, we study the convergence of the DMMIs under various network connections. For a network composed of 20 nodes, which are randomly distributed in a region, we adjust the number of short-range connections (or the number of nearest neighbors, denoted as NN) and the probability of long-range connections (denoted as p_{long}). When $p_{long} = 1$, the network is full-connected, and when $p_{long} = 0$, $NN = 0$, the network is connectionless. Since there are no true-values for the model parameters, the convergence can not be evaluated by traditional assessment index like mean square error. Here we use the mean mutual information over the network as the assessment index. Its mathematical formula is shown below

$$\frac{1}{J}\sum_{j}\hat{I}^{local}_{\mathcal{W}_j}(\boldsymbol{X}_j;K).$$

In Fig. 4 and Fig. 5, we present the mean convergence curves of 50 independent Monte Carlo simulations for the two-circle dataset and the three-class dataset, respectively. The situation for the half-moon dataset is similar and is not shown here to save space. From the figures, we can see that the diffusion DMMIs can converge within a few hundreds of iterations. Generally, the diffusion kernel DMMIs converge fast than the diffusion linear

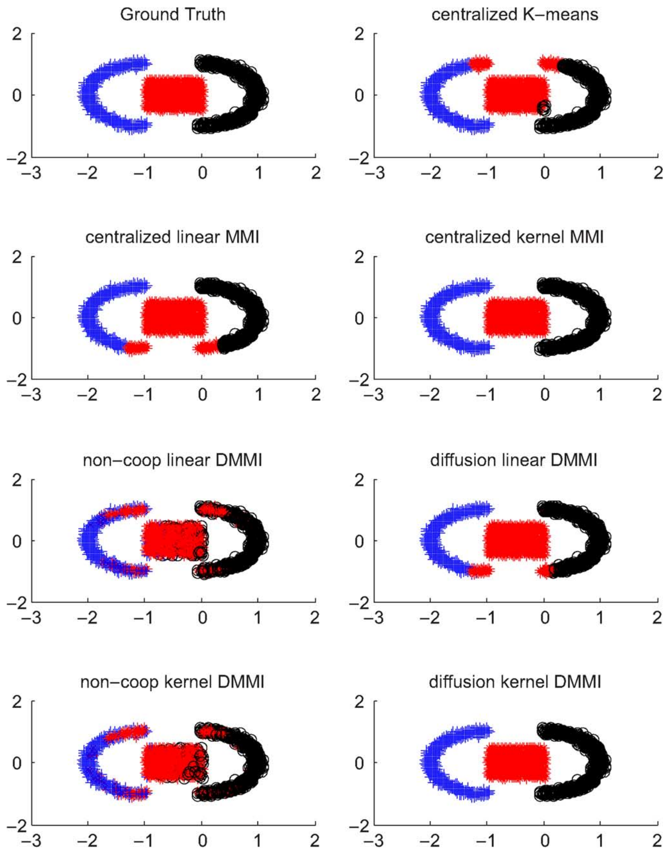


Fig. 3. Examples of clustering results of different algorithms for a three-class nonlinearly separable problem.

TABLE I
The Average Number of Misclassification Samples for Different Algorithm on Different Dataset

Dateset / Algorithm	the two-circle dataset	the double-moon dataset	the three-class dataset
Centralized K-means	214	291	209
Centralized linear MMI	0	102	159
Centralized kernel MMI	0	0	0
Non-coop. linear DMMI	126	229	517
Non-coop. kernel DMMI	44	203	315
Diffusion linear DMMI	0	100	142
Diffusion kernel DMMI	0	1	0

DMMIs. For the (linearly separable) two-circle dataset, the diffusion linear DMMIs eventually converge to the same level as that of the diffusion kernel DMMIs, while for the (linearly non-separable) three-class dataset, the diffusion linear DMMIs finally converge to a level lower than that of the diffusion kernel DMMIs. Besides, for the connectionless cases, the non-cooperative DMMIs can only converge to levels much lower than those of the diffusion DMMIs. These situations are in accord with the results shown in Table I. In addition, under the current data settings, a few connections, e.g. $p_{long} = 0, NN = 2$, are capable of providing individual nodes with enough global data information for correct clustering. In fact, part (but enough) of the global data (information) might be even more beneficial to the convergence of individual nodes than the whole global data (information). The reason is that with the increase of data amounts, there might be more data near or across the between-cluster boundaries, which would reduce the between-cluster distances and increase the difficulty of correct clustering. So, we see that in the above simulations, the diffusion DMMIs under sparse network connections even converge faster than the diffusion DMMIs under full connections. Similarly, in literatures on distributed K-means algorithms, they also find that the distributed K-means algorithms (under sparse network connections) outperform the centralized K-means algorithm in some cases [7], [9].

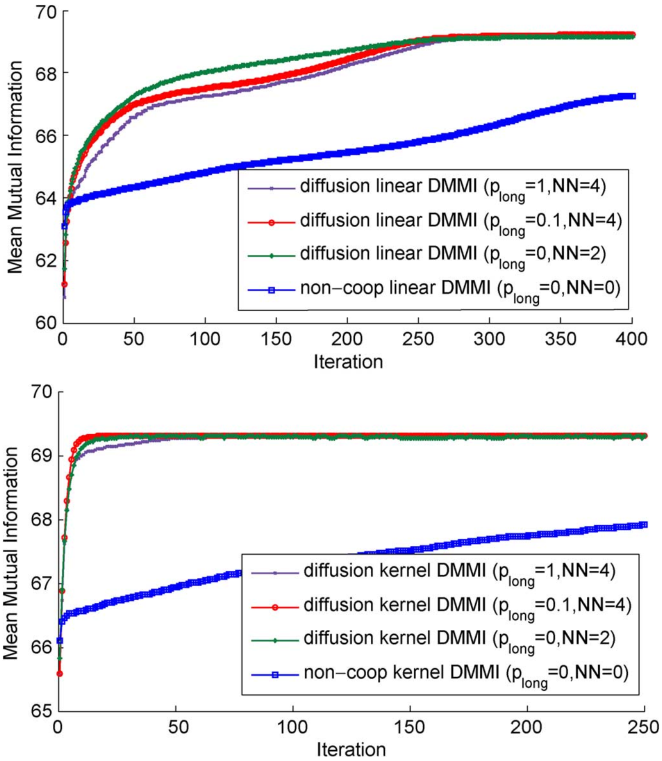


Fig. 4. The convergence curves of the DMMIs for the two-circle dataset under different network connections.

B. Real Data

In this subsection, we show the effectiveness of the proposed DMMI algorithm on a real atmosphere quality evaluation problem. To get a overall evaluation of the quality, we can collect air samples distributedly using a WSN. In this example, the real dataset consists of 2900 measurements of quality of air samples, 1500 for clean air samples and 1400 for slightly polluted air samples. Each data item records the concentrations of three most common pollutants, which are sulfur dioxide, nitrogen dioxide and PM10. All the data items (after normalization) are depicted in Fig. 6, where the star points (in red) denote the measurements of polluted air and the circle points (in blue) denote the measurements of clean air.

To evaluate the performance of the distributed clustering algorithms, we distribute the whole data to 20 nodes in the same way as that in the above simulations. We use the kernel DMMI to cluster the atmosphere data, since it was proven to be more

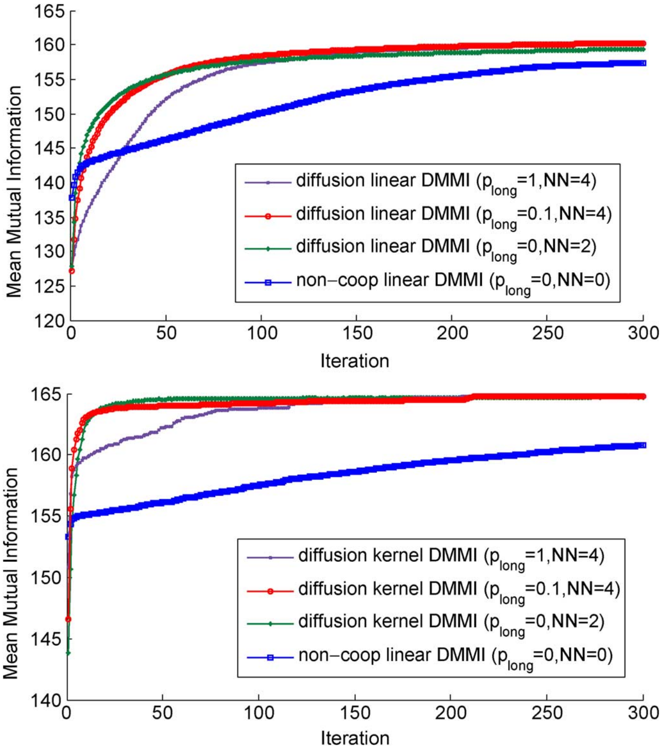


Fig. 5. The convergence curves of the DMMIs for the three-class dataset under different network connections.

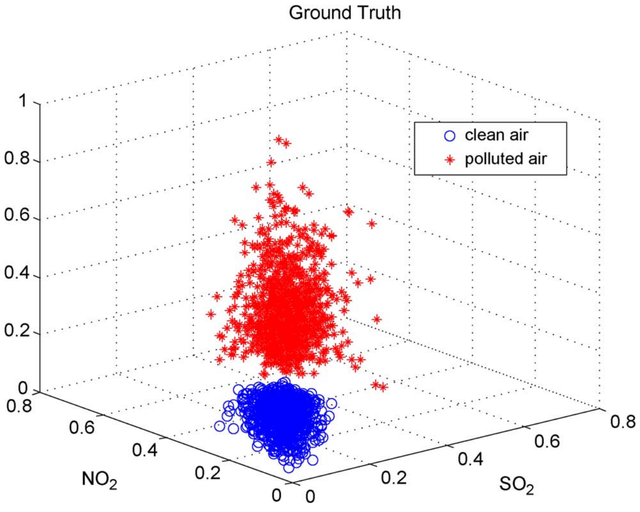


Fig. 6. A 3-D display of all the atmosphere data.

robust to the variations of inter-class boundaries than the linear DMMI. We also use the Gaussian kernel and set L to be 64 for the kernel DMMI. Besides, the performance of the centralized K-means, the centralized kernel MMI, and the non-cooperative kernel DMMI is also studied to provide comparisons.

The evaluation is performed by 50 independent Monte Carlo simulations. For a new simulation, the whole data is re-grouped into 20 subsets randomly, and then distributed to the 20 nodes. In Fig. 7, we show the clustering results of one simulation for different algorithms. Except the qualitative results, we provide the average number of misclassification samples in the Table II. From these results, we see that the diffusion kernel DMMI outperforms the K-means method and the non-cooperative kernel DMMI on the atmosphere dataset. The result achieved by the diffusion kernel DMMI is similar to that of the centralized kernel MMI. Moreover, they are both basically in accordance with the ground truth. This case indicates that the proposed DMMI algorithm is capable of exploring the overall data structure in real distributed application.

C. Cases of Unbalanced Data Distribution

In the above subsections, the local data of different nodes are randomly (roughly uniformly) sampled from the same global datasets, thus the local empirical distribution of class labels $\hat{p}_j(k)$ is similar to the overall distribution of class labels $\hat{p}(k)$. This similarity makes the approximation performed in (2) relatively accurate. Besides, the number of data items is equal for each node. Under these settings, the DMMI algorithms work well on the testing datasets. However, in some practical cases, the similarity between $\hat{p}_j(k)$ and $\hat{p}(k)$ might not be guaranteed, and the number of data items might be different for different nodes. In this subsection, we further evaluate the performance of the algorithms in such two kind of cases. We perform the evaluation on both the two-circle dataset (linearly separable) and the three-class dataset (linearly non-separable), respectively. We take the two datasets as the representatives. The situations for the other two datasets are similar, so we do not show their corresponding results.

1) Cases of Dissimilar Class Distributions: In this part, we consider cases that class distributions are dissimilar among different nodes. In the simulations, we deliberately distribute data subsets with unbalanced class ratios to different nodes. Specifically, the total number of data items contained in each subset is the same, while the respective ratios of data belonging to different classes vary obviously and randomly with nodes. The detailed profiles of class ratios used for nodes are shown in Fig. 8 and Fig. 9. Other parameter settings are kept the same as those adopt in the above simulations.

The evaluation is performed by 50 independent Monte Carlo simulations. In Table III, we show the average number of misclassification samples for the noncooperative DMMIs and the diffusion DMMIs. Compared with the results shown in Table I, we can see that, when the distribution of class labels varies obviously with nodes, the average numbers of misclassification samples for the noncooperative DMMIs increase significantly, while the average numbers of misclassification samples for the diffusion DMMIs still keep low. We consider the reason is that when without cooperation, some nodes have to perform their local clustering based on heavily unbalanced data, thus the misclassification rate increases. In contrast, when with the diffusion cooperation, the additional information provided by the neighboring nodes can help estimate the parameters of inter-cluster boundaries, thus the negative influence brought by the unbalanced data distribution could be avoided to some extent.

2) Cases of Unequal Data Amounts: In this part, we consider cases that the number of data items are unequal among different nodes. In the simulations, the same global datasets as those used in the above subsections are randomly grouped into 20 subsets with unequal data amounts. Then, these data subsets are uniformly allocated to 20 different nodes. The detailed profiles of data amounts at each node are shown in Fig. 10. Other

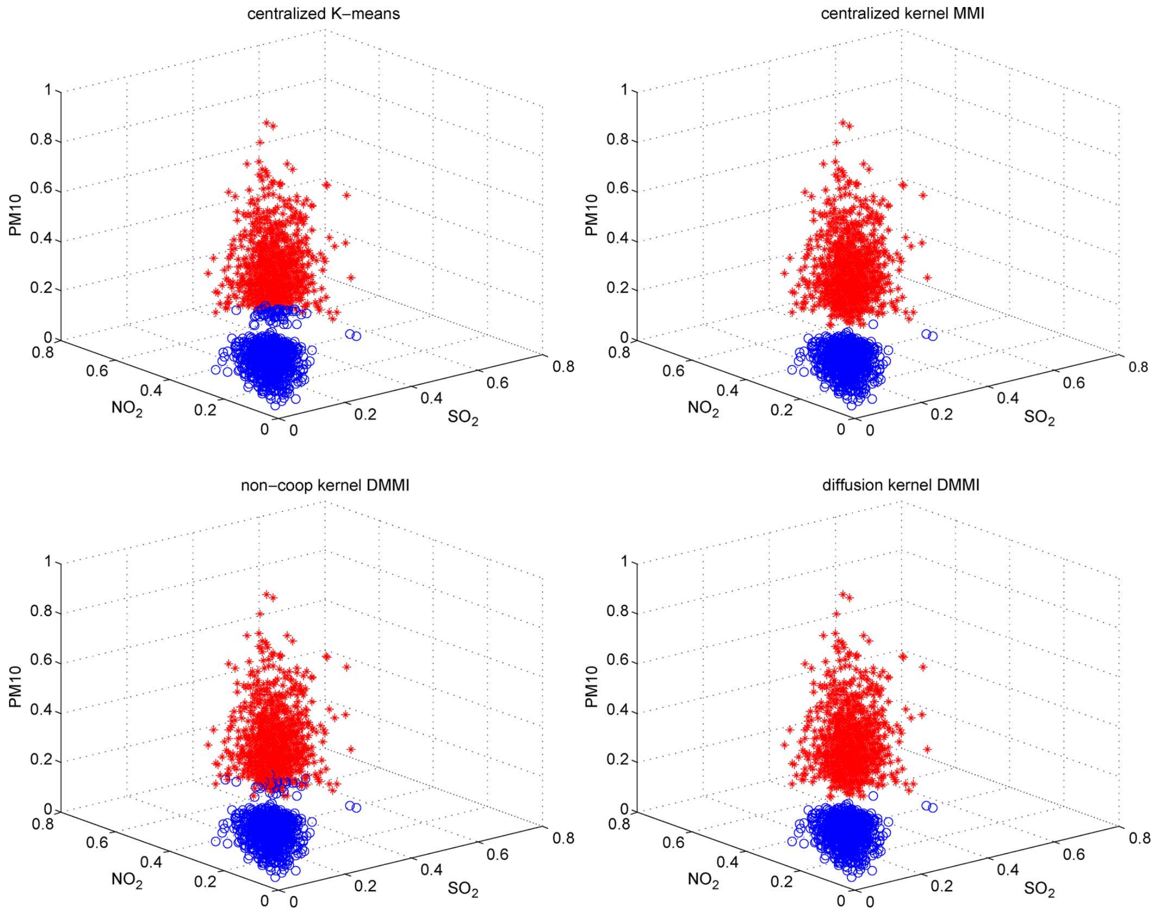


Fig. 7. Examples of clustering results of different algorithms on the atmosphere dataset.

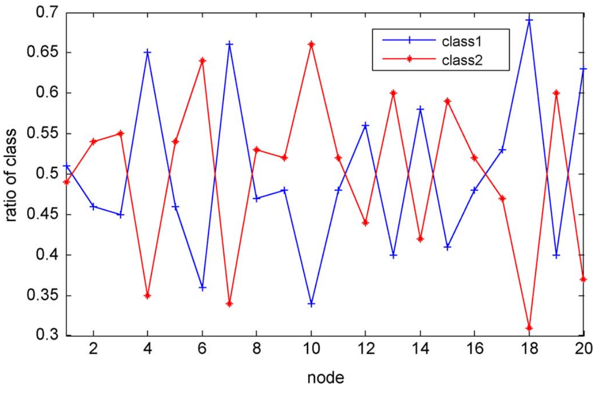


Fig. 8. The detailed profile of class ratios of different nodes for the two-circle dataset.

TABLE II
The Average Number of Misclassification Samples for Different Algorithm on the Atmosphere Data

Algorithm	# misclassification samples
Centralized K-means	85
Centralized kernel MMI	3
Non-coop. kernel DMMI	28
Diffusion kernel DMMI	3

TABLE III
The Average Number of Misclassification Samples for Different Algorithms in the Case of Dissimilar Class Distribution

Dateset Algorithm	the two-circle dataset	the three-class dataset
Non-coop. linear DMMI	168	649
Non-coop. kernel DMMI	92	494
Diffusion linear DMMI	0	178
Diffusion kernel DMMI	0	2

parameter settings are kept the same as those adopt in the above simulations.

The evaluation is performed by 50 independent Monte Carlo simulations. In Table IV, we show the average number of misclassification samples for the noncooperative DMMIs and the diffusion DMMIs. Compared with the results shown in Table I, we can see that, when the data amounts vary with nodes, the average numbers of misclassification samples for the DMMIs do not change much, and the diffusion DMMIs still work well. In the diffusion DMMIs, since the combination coefficients are set according to the data amount at each node, information from nodes with larger data amounts will have larger contributions in the diffusion corporations. We think that this scheme naturally

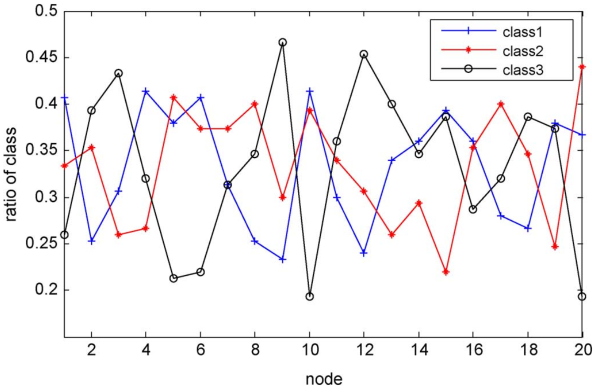


Fig. 9. The detailed profile of class ratios of different nodes for the three-class dataset.

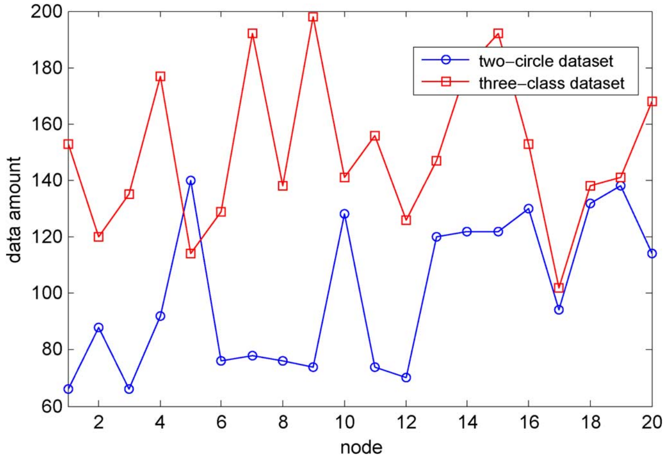


Fig. 10. The detailed profiles of data amounts at each node for the two-circle dataset and the three-class dataset.

TABLE IV
THE AVERAGE NUMBER OF MISCLASSIFICATION SAMPLES FOR DIFFERENT ALGORITHMS IN THE CASE OF UNEQUAL DATA AMOUNTS

Dateset / Algorithm	the two-circle dataset	the three-class dataset
Non-coop. linear DMMI	157	498
Non-coop. kernel DMMI	45	310
Diffusion linear DMMI	0	153
Diffusion kernel DMMI	0	0

takes the effect of data amounts into consideration and makes the algorithms robust to the inequality of data amounts.

All the above results demonstrate the effectiveness of the diffusion cooperation scheme and indicate that the proposed algorithms can still work well when the data distribution over nodes is somewhat unbalanced.

V. DISCUSSION AND CONCLUSION

In this paper, we have considered the MMI criterion in the context of distributed data clustering. Compared with the K-means-based and the GMM-based clustering algorithms, the MMI-based algorithms can capture the data structures beyond the first and the second order statistics, thus leading to more satisfactory clustering results for datasets with complicated data structures. Based on the MMI criterion, we have proposed a “global-like” local cost function for each node. In order to protect the privacy and save communication resource, we have developed a two-step iterative scheme to perform the optimization. Under this scheme, every node accomplishes its local clustering task merely based on its own data and limited information exchanges with its neighboring nodes. The proposed cost function and iterative scheme constitute a general framework for distributed MMI-based clustering. We have realized a linear type of DMMI algorithm upon this framework, by using the linear discriminative clustering model. The linear DMMI algorithm is appreciate for linearly separable problems. In addition, to handle linearly non-separable problems, we have proposed the kernel DMMI algorithm by using the modified kernel discriminative clustering function. The computational complexity and the communication costs of the two DMMI algorithms have been analyzed in detail. The performances of the proposed algorithms are evaluated on three synthetic datasets and one real dataset.

Our simulation results show that the diffusion cooperation based DMMI algorithms, including the linear DMMI and kernel DMMI, outperform the centralized K-means and the non-cooperative DMMI algorithms on all the synthetic datasets. The performance of the diffusion DMMI algorithms is comparable to the centralized MMI algorithms, both of which have low misclassification rates. Besides, the kernel DMMI shows excellent ability in exploring the overall data structure for the real atmosphere dataset, which indicates that the proposed information theoretic clustering algorithms are applicable in real distributed applications like environmental monitoring. Additionally, in our simulations, the proposed two DMMI algorithms maintain good clustering performance in the cases of unbalanced data distribution over nodes, which further reflects the flexibility and applicability of the algorithms for practical cases.

ACKNOWLEDGMENT

The authors would like to thank C. K. Huang and Y. Y. Lin for providing the real atmosphere dataset, which is helpful to evaluate the performances of our proposed algorithms.

REFERENCES

[1] A. K. Jain, “Data clustering: 50 years beyond K-means,” *Pattern Recogn. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[2] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley , 2002.

[3] A. K. Jain, A. Topchy, M. H. C. Law, and J. M. Buhmann, “Landscape of clustering algorithms,” in *Proc. Int. Conf. Pattern Recogniy*, 2004, vol. 1, pp. 260–263.

[4] R. Xu and D. Wunsch-II, “Survey of clustering algorithms,” *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.

[5] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: A review,” *ACM Comput. Surveys*, vol. 31, no. 3, pp. 265–323, 1999.

[6] J. Grabmeier and A. Rudolph, “Techniques of cluster algorithms in data mining,” *Data Mining and Knowl. Discov.*, vol. 6, no. 4, pp. 303–360, 2002.

[7] P. A. Forero, A. Cano, and G. B. Giannakis, “Distributed clustering using wireless sensor networks,” *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 707–724, Aug. 2011.

[8] I. Dhillon and D. Modha, “A data-clustering algorithm on distributed memory multiprocessors,” *Large-Scale Parallel Data Mining, Lecture Notes in Artif. Intell.*, pp. 245–260, 2000.

[9] P. A. Forero, A. Cano, and G. B. Giannakis, “Consensus-based k-means algorithm for distributed learning using wireless sensor networks,” in *Proc. Workshop Sens., Signal, Inf. Process.*, Sedona, AZ, USA, May 11–14, 2008.

[10] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed expectation-maximization algorithm for density estimation and classification using wireless sensor networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Las Vegas, NV, USA, 2008, pp. 1989–1992.

[11] J. Wolfe, A. Haghighi, and D. Klein, "Fully distributed EM for very large datasets," in *Proc. 25th Int. Conf. Mach. Learn.*, Helsinki, Finland, 2008, pp. 1184–1191.

[12] R. D. Nowak, "Distributed EM algorithms for density estimation and clustering in sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2245–2253, Aug. 2003.

[13] D. Gu, "Distributed EM algorithm for Gaussian mixtures in sensor networks," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1154–1166, Jul. 2008.

[14] E. Januzaj, H. P. Kriegel, and P. Martin, "Scalable density-based distributed clustering," in *Proc. Knowl. Discov. Databases: PKDD 2004*, 2004, vol. 19, no. 7, pp. 231–244.

[15] M. Klusch, S. Lodi, and G. Moro, "Distributed clustering based on sampling local density estimates," in *Proc. Int. Joint Conf. Artif. Intell.*, 2003, pp. 485–490.

[16] H. Kargupta, W. Huang, S. Krishnamoorthy, and E. Johnson, "Distributed clustering using collective principal component analysis," *Knowl. Inf. Syst. J.*, vol. 3, no. 4, pp. 422–448, 2001.

[17] S. Bandyopadhyay, C. Giannella, U. Maulik, H. Kargupta, K. Liu, and S. Datta, "Clustering distributed data streams in peer-to-peer environments," *Inf. Sci.*, vol. 176, no. 14, pp. 1952–1985, 2006.

[18] E. Gokcay and J. C. Principe, "Information theoretic clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 158–170, Apr. 2002.

[19] R. Jenssen, T. Eltoft, and J. C. Principe, "Information theoretic clustering: A unifying review of three recent algorithms," in *Proc. 6th Nordic Signal Process. Symp.*, Espoo, Finland, 2004, pp. 292–295.

[20] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *J. Mach. Learn. Res.*, vol. 6, pp. 1705–1749, 2005.

[21] N. Slonim, G. S. Atwal, G. Tkacik, and W. Bialek, "Information based clustering," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 51, pp. 18297–18302, 2005.

[22] X. Bai, S. W. Luo, and Y. B. Zhao, "Entropy based soft K-means clustering," in *Proc. EEE Int. Conf. Granular Comput.*, 2008, pp. 107–110.

[23] Q. Song, "A robust information clustering algorithm," *Neural Comput.*, pp. 2672–2698, 2005.

[24] C. Bohm, C. Faloutsos, J. Y. Pan, and C. Plant, "Robust information-theoretic clustering," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2005, vol. 17, no. 12, pp. 65–75.

[25] S. Gordon, H. Greenspan, and J. Goldberger, "Applying the information bottleneck principle to unsupervised clustering of discrete and continuous image representations," in *Proc. 9th IEEE Int. Conf. Comput. Vision*, 2003, pp. 370–377.

[26] A. Bardera, J. Rigau, I. Boada, M. Feixas, and M. Sbert, "Image segmentation using information bottleneck method," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1601–1612, Jul. 2009.

[27] N. Tishby, F. Pereira, and W. Bialek, "The information bottleneck method," in *Proc. 37th Ann. Allerton Conf. Commun., Contr. Comput.*, 1999, pp. 368–377.

[28] A. Krause, P. Perona, and R. G. Gomes, "Discriminative clustering by regularized information maximization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 775–783.

[29] S. Merugu and J. Ghosh, "A privacy-sensitive approach to distributed clustering," *Pattern Recognit. Lett.*, vol. 26, no. 4, pp. 399–410, 2005.

[30] A. M. Elmisery and H. Fu, "Privacy preserving distributed learning clustering of healthcare data using cryptography protocols," in *Proc. IEEE 34th Ann. Comput. Software and Appl. Conf. Workshops* , 2010, pp. 140–145.

[31] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley, 1991.

[32] J. C. Principe, *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. New York, NY, USA: Springer-Verlag, 2010.

[33] E. Parzen, "On estimation of a probability density function and mode," in *Time Series Analysis Papers*. San Francisco, CA: Holden-Day, 1967.

[34] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[35] C. G. Lopes and A. H. Sayed, "Distributed processing over adaptive networks," in *Proc. Adapt. Sens. Array Process. Workshop*, Lexington, MA, USA, Jun. 2006.

[36] C. Li, P. Shen, Y. Liu, and Z. Zhang, "Diffusion information theoretic learning for distributed estimation over network," *IEEE Trans. Signal Process.*, vol. 61, no. 16, pp. 4011–4024, Aug 15, 2013.

[37] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[38] Y. Liu, C. Li, and Z. Zhang, "Diffusion sparse least-mean squares over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4480–4485, Aug. 2012.

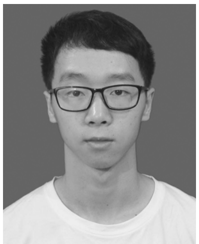
[39] Z. Liu, Y. Liu, and C. Li, "Distributed sparse recursive least-squares over networks," *IEEE Trans. Signal Process.*, vol. 62, no. 6, pp. 1386–1395, Jun. 2014.

[40] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *J. Mach. Learn. Res.*, vol. 11, pp. 1663–1707, 2010.

[41] X. He, D. Cai, Y. Shao, H. Bao, and J. Han, "Laplacian regularized Gaussian mixture model for data clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 9, pp. 1406–1418, 2011.

[42] R. Jenssen, D. Erdogmus, K. E. Hild, J. C. Principe, and T. Eltoft, "Optimizing the Cauchy-Schwarz PDF divergence for information theoretic, non-parametric clustering," in *Proc. Int. Workshop on Energy Minimiz. Methods in Compute. Vision Pattern Recogn.*, St. Augustine, FL, USA, Nov. 2005, pp. 34–45.

[43] S. Datta, C. R. Giannella, and H. Kargupta, "Approximate distributed k-means clustering over a peer-to-peer network," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 10, pp. 1372–1388, Oct. 2009.



Pengcheng Shen received the B.S. degree in Information Science and Electronic Engineering from Zhejiang University, Hangzhou, China, in 2011.

He is currently pursuing the Ph.D. degree with the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. His current research interests include statistical signal processing, wireless sensor network, and machine intelligence.

Chunguang Li received the M.S. degree in Pattern Recognition and Intelligent Systems and the Ph.D. degree in Circuits and Systems from the University of Electronic Science and Technology of China, Chengdu, China, in 2002 and 2004, respectively.

Currently, he is a Professor with the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. His current research interests include statistical signal processing, wireless sensor network, and computational neuroscience.