Self-Paced Clustering Ensemble

Peng Zhou, Liang Du, *Member, IEEE*, Xinwang Liu, Yi-Dong Shen, Mingyu Fan, and Xuejun Li

***Abstract*— The clustering ensemble has emerged as an important extension of the classical clustering problem. It provides an elegant framework to integrate multiple weak base clusterings to generate a strong consensus result. Most existing clustering ensemble methods usually exploit all data to learn a consensus clustering result, which does not sufficiently consider the adverse effects caused by some difficult instances. To handle this problem, we propose a novel self-paced clustering ensemble (SPCE) method, which gradually involves instances from easy to difficult ones into the ensemble learning. In our method, we integrate the evaluation of the difficulty of instances and ensemble learning into a unified framework, which can automatically estimate the difficulty of instances and ensemble the base clusterings. To optimize the corresponding objective function, we propose a joint learning algorithm to obtain the final consensus clustering result. Experimental results on benchmark data sets demonstrate the effectiveness of our method.**

***Index Terms*— Clustering ensemble, consensus learning, self-paced learning.**

I. Introduction

CLUSTERING is a fundamental unsupervised problem in machine learning tasks. It has been widely used in various applications and demonstrated promising performance. However, according to [1], conventional single clustering algorithms usually suffer from the following problems: 1) given a data set, different structures may be discovered by various clustering methods due to their different objective functions; 2) for a single clustering method, since no ground truth is available, it could be hard to validate the clustering results; and 3) some methods, e.g., k-means, highly depend on their initializations. To address these problems, the idea of a clustering ensemble has been proposed.

Clustering ensemble provides an elegant framework for combining multiple weak base clusterings of a data set to generate a consensus clustering [2]. In recent years, many clustering ensemble methods have been proposed [3]–[7]. For example, Strehl *et al.* and Topchy *et al.* proposed information theoretic-based clustering ensemble methods, respectively, in [3] and [4]; Fern *et al.* extended graph cut method into clustering ensemble [8]; and Ren *et al.* proposed a weighted-object graph partitioning algorithm for clustering ensemble [9].

These methods try to learn the consensus clustering result from all instances by taking advantage of diversity between base clusterings and reducing the redundancy in the clustering ensemble. However, since the base clustering results may not be entirely reliable, it is inappropriate to always use all data for clustering ensemble. Intuitively, some instances are difficult for clustering or even outliers, which leads to the poor performance of the base clusterings. At the beginning of learning, these difficult instances may mislead the model because the early model may not have the ability to handle these difficult instances.

To tackle this problem, we ensemble the base clusterings in a curriculum learning framework. Curriculum learning is proposed by Bengio *et al.* [10], which incrementally involves instances (from easy to difficult ones) into learning. The key idea is that, in the beginning, the model is relatively weak, and thus, it needs some easy instances for training. Then, the ability of the model becomes increasingly strong as time goes on so that it can handle more and more difficult instances. Finally, it is strong enough to handle almost all instances. To formulate this key idea of curriculum learning, we propose a novel self-paced clustering ensemble (SPCE) method, which can automatically evaluate the difficulty of instances and gradually include instances from easy to difficult ones into the ensemble learning.

In our method, we estimate the difficulty of instances with the agreement of base clustering results, i.e., if many base clustering results agree with each other in some instances, these instances may be easy for clustering. We adapt this idea to the ensemble method and propose a self-paced learning method that evaluates the difficulty of instances automatically in the process of the ensemble. On the one hand, easy instances can be helpful to ensemble learning; on the other hand, with the learning process, more and more instances become easy for learning. Since the clustering result represents the relation between two instances, i.e., it indicates whether two instances belong to the same cluster or not, we transform all base clustering results into connective matrices and try to learn

Manuscript received June 23, 2019; revised December 5, 2019 and March 15, 2020; accepted March 28, 2020. This work was supported in part by the National Natural Science Fund of China under Grant 61806003, Grant 61976129, Grant 61922088, Grant 61976205, Grant 61772373, and Grant 61972001 and in part by the Key Natural Science Project of the Anhui Provincial Education Department under Grant KJ2018A0010. *(Corresponding author: Yi-Dong Shen.)*

Peng Zhou is with the School of Computer Science and Technology, Anhui University, Hefei 230601, China, and also with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China (e-mail: zhoupeng@ahu.edu.cn).

Liang Du is with the School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China (e-mail: duliang@sxu.edu.cn).

Xinwang Liu is with the College of Computer, National University of Defense Technology, Changsha 410073, China (e-mail: xinwangliu@nudt.edu.cn).

Yi-Dong Shen is with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China (e-mail: ydshen@ios.ac.cn).

Mingyu Fan is with the College of Maths and Information Science, Wenzhou University, Wenzhou 325035, China (e-mail: fanmingyu@amss.ac.cn).

Xuejun Li is with the School of Computer Science and Technology, Anhui University, Hefei 230601, China (e-mail: xjli@ahu.edu.cn).

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

Digital Object Identifier 10.1109/TNNLS.2020.2984814

a consensus connective matrix from them. We use a weight matrix to represent the difficulty of all pairs in the connective matrix, i.e., the larger the weight of a pair is, the easier to decide whether such two instances belong to the same cluster is. Then, we integrate the weight matrix learning and the consensus connective matrix learning into a unified objective function. To optimize this objective function, we provide a block coordinate descent schema that can jointly learn the consensus connective matrix and the weight matrix.

The extensive experiments are conducted on benchmark data sets, and the results demonstrate the effectiveness of our self-paced learning method.

This article is organized as follows. Section II describes some related work. Section III presents in detail the main algorithm of our method. Section IV shows the experimental results, and Section V concludes this article.

II. Related Work

In this section, we first present the basic notations and then introduce some related works. Throughout this article, we use boldface uppercase and lowercase letters to denote matrices and vectors, respectively. The (i, j)th element of a matrix $\mathbf{M}$ is denoted as M_{ij}, and the ith element of a vector $\mathbf{v}$ is denoted as v_i. Given a matrix $\mathbf{M} \in \mathbb{R}^{n\times d}$, we use $\|\mathbf{M}\|_F = (\sum_{i=1}^{n}\sum_{j=1}^{d} M_{ij}^2)^{1/2}$ to denote its Frobenius norm. We use $\|\mathbf{M}\|_0$ to denote its ℓ_0-norm, which is the number of nonzero elements in $\mathbf{M}$. Since ℓ_0-norm is nonconvex and discontinuous, ℓ_1-norm is often used as an approximation of ℓ_0-norm. ℓ_1-norm of $\mathbf{M}$ is defined as $\|\mathbf{M}\|_1 = \sum_{i=1}^{n}\sum_{j=1}^{d} |M_{ij}|$.

A. Clustering Ensemble

Ensemble learning trains multiple learners and tries to combine their predictions to achieve better learning performance [11]. Since the generalization ability of the ensemble method could be better than the base learners [12], ensemble learning has been applied to various domains, such as image analysis [13], [14], medical diagnosis [15], and multiview data analysis [16]–[20]. At an early age, many ensemble methods were designed for supervised learning, in which the labels of training data were known. For example, Freund and Schapire [21] proposed the famous AdaBoost method that evaluated the base learners and then applied the evaluation results to weight each base learner and change the training data distribution; Friedman [22] proposed the gradient boosting decision tree method that ensembled the results of multiple decision trees. In these methods, the labels of training data are necessary for eliminating the ambiguity when combining the base learners [5].

However, in unsupervised learning, due to the lack of training labels, it is more challenging to design the ensemble methods. Moreover, as introduced earlier, conventional single clustering methods often suffer from stable and robust problems. Therefore, the clustering ensemble has attracted increasing attention in recent years. At an early age, some information theoretic-based methods are proposed. For example, Strehl and Ghosh [3] first introduced the clustering ensemble task and formalized clustering ensemble as a combinatorial optimization problem in terms of shared mutual information; then, Topchy *et al.* [4] combined clusterings based on the observation that the consensus function of clustering ensemble is related to classical intraclass variance criterion using the generalized mutual information definition.

In this article, we follow the problem setting of clustering ensemble defined in [3] and [4]. In more detail, let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a data set of n data points. Suppose that we are given a set of m clusterings $\mathcal{C} = \{\mathcal{C}^1, \mathcal{C}^2, \ldots, \mathcal{C}^m\}$ of the data in $\mathcal{X}$, each clustering $\mathcal{C}^i$ consisting of a set of clusters $\{\pi_1^i, \pi_2^i, \ldots, \pi_k^i\}$, where k is the number of clusters in $\mathcal{C}^i$ and $\mathcal{X} = \bigcup_{j=1}^{k} \pi_j^i$. Note that the number of clusters k could be different for different clusterings. According to [2]–[4], the goal of clustering ensemble is to learn a consensus partition of the data set from the m base clusterings $\mathcal{C}^1, \ldots, \mathcal{C}^m$.

In recent years, to learn the consensus partition, more and more techniques have been applied to ensemble base clustering results. For example, Zhou and Tang [5] proposed an alignment method to combine multiple k-means clustering results. Some works applied the famous matrix factorization to the clustering ensemble. For instance, Li *et al.* [23] and Li and Ding [24] factorized the connective matrix into two indicator matrices by symmetric nonnegative matrix factorization. Besides k-means and matrix factorization, spectral clustering was also extended into clustering ensemble tasks, such as in [25]–[27]. Some methods introduced a probabilistic graphical model into the clustering ensemble. For example, Wang *et al.* [28] applied a Bayesian method to clustering ensemble; Huang *et al.* [29] learned a consensus clustering result with a factor graph. Since the clustering diversity and quality are essential in ensemble learning, many methods made full use of the diversity and quality to combine base clusterings. For example, Abbasi *et al.* [30] proposed a new stability measure called edited normalized mutual information (NMI) and used it to ensemble base clusterings; Bagherinia *et al.* [31] provided a fuzzy clustering ensemble by considering the diversity and quality of base clusterings.

Besides these works that ensembled all base clustering results, some works tried to select some informative and nonredundant base clustering results for the ensemble. For example, Azimi and Fern [32] proposed an adaptive clustering ensemble selection method to select the base results; Hong *et al.* [33] selected base clusterings by a resampling method; Parvin and Minaei-Bidgoli [34], [35] proposed a weighted locally adaptive clustering for clustering ensemble selection; Yu *et al.* [36] transferred the clustering selection to feature selection and designed a hybrid strategy to select base results; Zhao *et al.* [37] proposed internal validity indices for clustering ensemble selection; and Shi *et al.* [38] extended the transfer learning into clustering ensemble leading to a transfer clustering ensemble selection method.

In this article, we will propose a clustering ensemble method based on connective matrices. Since the clustering result represents the relation between two instances as introduced earlier, from $\mathcal{C}$, following [24], [39]–[42], we can construct the connective matrix $\mathbf{S}^{(i)} \in \mathbb{R}^{n\times n}$ for partition $\mathcal{C}^i$ as

$$S_{pq}^{(i)} = \begin{cases} 1, & \text{if } \mathbf{x}_p \text{ and } \mathbf{x}_q \text{ belong to the same cluster,} \\ 0, & \text{otherwise.} \end{cases}$$

The target of our clustering ensemble method is to learn a consensus matrix $\mathbf{S}$ from $\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \ldots, \mathbf{S}^{(m)}$ and then obtain the final clustering result from the consensus matrix $\mathbf{S}$. Traditional connective matrix-based methods [24], [39]–[42] constructed coassociation matrix by linearly combining all connective matrices and then obtain the consensus clustering result from coassociation matrix. Different from them, which use all instances for the ensemble, we ensemble the base clusterings in a curriculum learning framework, which incrementally involves instances (from easy to difficult ones) into ensemble learning.

B. Self-Paced Learning

Inspired by the learning process of humans, Bengio *et al.* [10] proposed note of curriculum learning. The idea is to incrementally involve instances into learning, where easy instances are involved first and harder ones are then introduced gradually. One benefit of this strategy is that it helps alleviate the local optimum problem in nonconvex optimization, as introduced in [43] and [44].

To formulate the key principle of curriculum learning that gradually includes instances from easy to difficult ones, Kumar *et al.* [45] proposed the self-paced learning. More formally, given a data set $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$ containing n instances, where $\mathbf{x}_i \in \mathbb{R}^d$ is the d-dimensional feature vector of the ith instance and y_i is its label, $L(y_i, g(\mathbf{x}_i, \boldsymbol{\theta}))$ is denoted as the loss function that indicates the cost between the ground truth y_i and the estimated label $g(\mathbf{x}_i, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents the model parameters in the decision function g. According to [46] and [47], the self-paced learning introduces a weighted loss term on all instances and a general regularizer term on instance weights, which is in the following form:

$$\min_{\boldsymbol{\theta}, \mathbf{w}} \sum_{i=1}^{n} (w_i L(y_i, g(\mathbf{x}_i, \boldsymbol{\theta})) + f(w_i, \lambda)) \tag{1}$$

where λ is an age parameter for controlling the learning pace, w_i is the weight of the ith instance, and $f(\mathbf{w}, \lambda)$ is the self-paced regularizer. When fixing $\boldsymbol{\theta}$, supposing that $l_i = L(y_i, g(\mathbf{x}_i, \boldsymbol{\theta}))$ and $w_i^*(\lambda, l_i)$ is the optimum weight of the ith instance, which is relative with λ and l_i, $f(w_i, \lambda)$ should satisfy that $w_i^*(\lambda, l_i)$ is monotonically decreasing with l_i and increasing with λ, as suggested in [48]–[50]. Since $w_i^*(\lambda, l_i)$ is monotonically decreasing with l_i, easy instances, which have low loss, will have large weight, which means that we learn from these instances first. $w_i^*(\lambda, l_i)$ is monotonically increasing with λ indicates that with the learning process (λ grows), more and more instances are used for learning.

Therefore, the process of self-paced learning is to optimize (1) via alternating minimization. Fixing $\boldsymbol{\theta}$ and solving $\mathbf{w}$ are to learn the weight of each instance; fixing $\mathbf{w}$ and solving $\boldsymbol{\theta}$ are to learn the model using the easy instances. Due to the promising performance, self-paced learning has been applied to handle many machine learning tasks, such as multitask learning [51], [52] and robust classification [53]. In this article, we will extend this self-paced learning framework into unsupervised ensemble learning.

TABLE I
NOTATIONS AND DESCRIPTIONS USED IN OUR METHOD

Notation	Description
n	number of instances
m	number of base clusterings
c	number of clusters
$\mathbf{S}^{(i)} \in \mathbb{R}^{n\times n}$	the connective matrix of the i-th base clustering
$\hat{\mathbf{S}} \in \mathbb{R}^{n\times n}$	the co-association matrix
$\mathbf{S} \in \mathbb{R}^{n\times n}$	the learned consensus matrix
$\boldsymbol{\Omega} \in \{0,1\}^{n\times n}$	the indicator matrix, $\Omega_{ij} = 1$ if $\hat{S}_{ij} = 0$ or 1, $\Omega_{ij} = 0$ otherwise
$\mathbf{L} \in \mathbb{R}^{n\times n}$	the Laplacian matrix of $\mathbf{S}$
$\mathbf{W} \in \mathbb{R}^{n\times n}$	the weight matrix to show the difficulty of each pair

III. SELF-PACED CLUSTERING ENSEMBLE

In this section, we provide the framework of our SPCE method. The main notations and their descriptions used in this section are shown in Table I.

A. Mining the Most Certain Information

As we know, self-paced learning gradually incorporates easy to more complex samples into training. In our task, since we handle m connective matrices $\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \ldots, \mathbf{S}^{(m)}$, the samples are the data pairs appearing in $\mathbf{S}^{(i)}$. To apply the self-paced learning, we first find the easiest or the most certain data pairs. Here, a voting method is used to find the most certain pairs. Given any instance pair $\mathbf{x}_p$ and $\mathbf{x}_q$, if all m connective matrices agree that $\mathbf{x}_p$ and $\mathbf{x}_q$ belong to or not belong to the same cluster, we regard this pair as the most certain pair. More formally, we compute the coassociation matrix $\hat{\mathbf{S}}$ as

$$\hat{\mathbf{S}} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{S}^{(i)}. \tag{2}$$

It is easy to verify that for any $1 \le p, q \le n$, we have $0 \le \hat{S}_{pq} \le 1$. $\hat{S}_{pq} = 1$ indicates that all connective matrices agree that $\mathbf{x}_p$ and $\mathbf{x}_q$ belong to the same cluster, and $\hat{S}_{pq} = 0$ indicates that all connective matrices agree that they belong to the different clusters. Thus, the most certain pairs are the elements in $\hat{\mathbf{S}}$, which are either 1 or 0. To make the learned consensus matrix $\mathbf{S}$ to preserve such certain information, we directly set $\mathbf{S}$ as

$$S_{pq} = \begin{cases} 1, & \text{if } \hat{S}_{pq} = 1, \\ 0, & \text{if } \hat{S}_{pq} = 0, \\ \text{missing}, & \text{otherwise.} \end{cases}$$

Therefore, the consensus matrix $\mathbf{S}$ can be learned by solving a matrix completion problem, i.e., we fill the missing values in $\mathbf{S}$ by self-paced consensus learning.

B. Self-Paced Consensus Learning

Since $\mathbf{S}$ is the consensus matrix, we wish to minimize the disagreement between it and all connective matrices.

A natural idea is to minimize $\sum_{i=1}^{m} \|\mathbf{S}-\mathbf{S}^{(i)}\|_F^2$. However, this objective treats all m clustering results equally, which may be inappropriate. Intuitively, the quality of each clustering result is different, and we wish the better clustering results contribute more in the consensus learning. Thus, we can modify the objective to $\sum_{i=1}^{m} \alpha_i \|\mathbf{S}-\mathbf{S}^{(i)}\|_F^2$, where α_i is the weight of the ith base clustering result. Next, we should decide the weight of each clustering result. Inspired by the autoweighted technique proposed in [54] and [55], we can define $\alpha_i = (1/(\|\mathbf{S}-\mathbf{S}^{(i)}\|_F))$, which means the closer $\mathbf{S}^{(i)}$ to the consensus matrix, the larger the weight of the ith clustering result is. Thus, we obtain the following objective function:

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{i=1}^{m} \left\|\mathbf{S}-\mathbf{S}^{(i)}\right\|_F \\ &\text{s.t. } \mathbf{S}\odot\mathbf{\Omega} = \hat{\mathbf{S}}\odot\mathbf{\Omega}, \quad 0 \le S_{pq} \le 1 \quad \forall p,q \end{aligned} \tag{3}$$

where $\mathbf{\Omega} \in \mathbb{R}^{n\times n}$ is an indicator matrix, whose element $\mathbf{\Omega}_{pq} = 1$ if $\hat{S}_{pq} = 0$ or 1 and $\mathbf{\Omega}_{pq} = 0$ otherwise. $\odot$ is the Hadamard product, which means the elementwise production of two matrices. The constraint is to make sure that the consensus matrix $\mathbf{S}$ preserves the certain information.

To modify (3) into the self-paced learning framework, we should decide which data pairs are easy samples. Here, we follow the idea of voting introduced before. Given a data pair $\mathbf{x}_p$ and $\mathbf{x}_q$, if most $\mathbf{S}^{(i)}$ agrees with each other, we believe that this pair is an easy pair. More formally, the smaller $\sum_{i=1}^{m}(S_{pq}-S_{pq}^{(i)})^2$ is, the easier the pair $(\mathbf{x}_p, \mathbf{x}_q)$ is. To this end, we introduce a weight matrix $\mathbf{W}$ whose element $0 \le W_{pq} \le 1$ indicates the weight of the pair $(\mathbf{x}_p, \mathbf{x}_q)$. The larger W_{pq} is, the easier this pair is. Then, following the self-paced learning framework, we set $f(\omega,\lambda)$ in (1) as $f(\mathbf{W},\lambda) = -\lambda\|\mathbf{W}\|_1$ and obtain the following formula:

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{W}} \sum_{i=1}^{m} \|(\mathbf{S}-\mathbf{S}^{(i)})\odot\mathbf{W}\|_F - \lambda\|\mathbf{W}\|_1 \\ &\text{s.t. } \mathbf{S}\odot\mathbf{\Omega} = \hat{\mathbf{S}}\odot\mathbf{\Omega}, \quad 0 \le S_{pq} \le 1 \quad \forall p,q \\ &\quad 0 \le W_{pq} \le 1 \quad \forall p,q \end{aligned} \tag{4}$$

where λ is the age parameter and becomes increasingly larger in the process of optimization.

C. Overall Objective Function

Equation (4) provides a self-paced framework to learn the consensus matrix, and now, we need to transform it into the final clustering result. Suppose that we want to partition the instances into c clusters; the easiest way is to make $\mathbf{S}$ contain just c connected components. Note that we obtain the clustering result by finding the c connected components without any uncertain discretization procedures, such as k-means.

Let $\mathbf{L}$ be the Laplacian matrix of $\mathbf{S}$, i.e., $\mathbf{L} = \mathbf{D} - ((\mathbf{S}+\mathbf{S}^T)/2)$, where $\mathbf{D}$ is a diagonal matrix whose pth diagonal element is $D_{pp} = \sum_q((S_{pq}+S_{qp})/2)$. If $\mathbf{S}$ is nonnegative, then the Laplacian matrix $\mathbf{L}$ has an important property as follows.

Theorem 1 [56]: The multiplicity c of the eigenvalue 0 of the Laplacian matrix $\mathbf{L}$ is equal to the number of connected components in the consensus matrix $\mathbf{S}$.

Theorem 1 indicates that if $\text{rank}(\mathbf{L}) = n-c$, where $\text{rank}(\mathbf{L})$ denotes the rank of matrix $\mathbf{L}$, then we already partition the instances into c clusters based on $\mathbf{S}$ without any discretization procedures, such as k-means. Motivated by this theorem, we add the constraint $\text{rank}(\mathbf{L}) = n-c$ to the objective function

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{W}} \sum_{i=1}^{m} \|(\mathbf{S}-\mathbf{S}^{(i)})\odot\mathbf{W}\|_F - \lambda\|\mathbf{W}\|_1 \\ &\text{s.t. } \mathbf{S}\odot\mathbf{\Omega} = \hat{\mathbf{S}}\odot\mathbf{\Omega}, \quad 0 \le S_{pq} \le 1 \quad \forall p,q, \\ &\quad \text{rank}(\mathbf{L}) = n-c, \\ &\quad 0 \le W_{pq} \le 1 \quad \forall p,q. \end{aligned} \tag{5}$$

Last but not least, to obtain a clearer clustering structure, we wish the consensus matrix $\mathbf{S}$ to be as sparse as possible so that $\mathbf{S}$ can represent a clear graph structure. To achieve this, we impose a sparse regularized term $\|\mathbf{S}\|_0$ on the objective function and obtain the following formula:

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{W}} \sum_{i=1}^{m} \|(\mathbf{S}-\mathbf{S}^{(i)})\odot\mathbf{W}\|_F - \lambda\|\mathbf{W}\|_1 + \gamma\|\mathbf{S}\|_0 \\ &\text{s.t. } \mathbf{S}\odot\mathbf{\Omega} = \hat{\mathbf{S}}\odot\mathbf{\Omega}, \quad 0 \le S_{pq} \le 1 \quad \forall p,q, \\ &\quad \text{rank}(\mathbf{L}) = n-c, \\ &\quad 0 \le W_{pq} \le 1 \quad \forall p,q \end{aligned} \tag{6}$$

where γ is a balancing hyperparameter that can adjust the sparsity of $\mathbf{S}$. Note that we use ℓ_0-norm here instead of ℓ_1-norm or any other convex or nonconvex approximation, which can make $\mathbf{S}$ as sparse as possible.

Since (6) involves the Frobenius norm and rank function that are difficult to optimize, we should relax (6) to simplify the optimization.

First, we handle the Frobenius norm. By introducing auxiliary weight variables $\boldsymbol{\alpha}$, where $0 \le \alpha_i \le 1$ and $\sum_{i=1}^{m}\alpha_i = 1$, we have the following theorem.

Theorem 2: $\min_{\mathbf{S},\mathbf{W}} \sum_{i=1}^{m} \|(\mathbf{S}-\mathbf{S}^{(i)})\odot\mathbf{W}\|_F$ is equivalent to $\min_{\mathbf{S},\mathbf{W},\boldsymbol{\alpha}} \sum_{i=1}^{m}(1/\alpha_i)\|(\mathbf{S}-\mathbf{S}^{(i)})\odot\mathbf{W}\|_F^2$.

Proof: Let $\mathbf{S}^*$, $\mathbf{W}^*$ and $\boldsymbol{\alpha}^*$ denote the optima of $\min_{\mathbf{S},\mathbf{W},\boldsymbol{\alpha}} \sum_{i=1}^{m}(1/\alpha_i)\|(\mathbf{S}-\mathbf{S}^{(i)})\odot\mathbf{W}\|_F^2$. To prove that $\mathbf{S}^*$ and $\mathbf{W}^*$ are also the optima of $\min_{\mathbf{S},\mathbf{W}} \sum_{i=1}^{m}\|(\mathbf{S}-\mathbf{S}^{(i)})\odot\mathbf{W}\|_F$, we need to prove that given any $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{W}}$, we have

$$\sum_{i=1}^{m} \|(\mathbf{S}^*-\mathbf{S}^{(i)})\odot\mathbf{W}^*\|_F \le \sum_{i=1}^{m} \|(\tilde{\mathbf{S}}-\mathbf{S}^{(i)})\odot\tilde{\mathbf{W}}\|_F.$$

Consider that

$$\begin{aligned} &\left(\sum_{i=1}^{m} \|(\mathbf{S}^*-\mathbf{S}^{(i)})\odot\mathbf{W}^*\|_F\right)^2 \\ &\quad \le \left(\sum_{i=1}^{m} \frac{1}{\alpha_i^*}\|(\mathbf{S}^*-\mathbf{S}^{(i)})\odot\mathbf{W}^*\|_F^2\right)\left(\sum_{i=1}^{m}\alpha_i^*\right) \\ &\quad = \sum_{i=1}^{m} \frac{1}{\alpha_i^*}\|(\mathbf{S}^*-\mathbf{S}^{(i)})\odot\mathbf{W}^*\|_F^2 \\ &\quad \le \sum_{i=1}^{m} \frac{1}{\tilde{\alpha}_i}\|(\tilde{\mathbf{S}}-\mathbf{S}^{(i)})\odot\tilde{\mathbf{W}}\|_F^2. \end{aligned} \tag{7}$$

The first inequality is due to the Cauchy–Schwarz inequality, the second equality is due to $\sum_{i=1}^{m}\alpha_i^*=1$, and the third inequality is because since $\mathbf{S}^*$, $\mathbf{W}^*$ and $\boldsymbol{\alpha}^*$ are the optima of $\min_{\mathbf{S},\mathbf{W},\boldsymbol{\alpha}}\sum_{i=1}^{m}(1/\alpha_i)\|(\mathbf{S}-\mathbf{S}^{(i)})\odot\mathbf{W}\|_F^2$, given any $\tilde{\alpha}_i$, $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{W}}$, we have $\sum_{i=1}^{m}(1/\alpha_i^*)\|(\mathbf{S}^*-\mathbf{S}^{(i)})\odot\mathbf{W}^*\|_F^2\le\sum_{i=1}^{m}(1/\tilde{\alpha}_i)\|(\tilde{\mathbf{S}}-\mathbf{S}^{(i)})\odot\tilde{\mathbf{W}}\|_F^2$.

Note that $\tilde{\alpha}_i$ can take any value that satisfies $\sum_{i=1}\tilde{\alpha}_i=1$ in (7). Specially, we set $\tilde{\alpha}_i=(\|(\tilde{\mathbf{S}}-\mathbf{S}^{(i)})\odot\tilde{\mathbf{W}}\|_F)/(\sum_{i=1}^{m}\|(\tilde{\mathbf{S}}-\mathbf{S}^{(i)})\odot\tilde{\mathbf{W}}\|_F)$. Then, we have

$$\begin{aligned}&\left(\sum_{i=1}^{m}\|(\mathbf{S}^*-\mathbf{S}^{(i)})\odot\mathbf{W}^*\|_F\right)^2\\&\le\sum_{i=1}^{m}\frac{\sum_{i=1}^{m}\|(\tilde{\mathbf{S}}-\mathbf{S}^{(i)})\odot\tilde{\mathbf{W}}\|_F}{\|(\tilde{\mathbf{S}}-\mathbf{S}^{(i)})\odot\tilde{\mathbf{W}}\|_F}\|(\tilde{\mathbf{S}}-\mathbf{S}^{(i)})\odot\tilde{\mathbf{W}}\|_F^2\\&=\left(\sum_{i=1}^{m}\|(\tilde{\mathbf{S}}-\mathbf{S}^{(i)})\odot\tilde{\mathbf{W}}\|_F\right)^2\end{aligned}\tag{8}$$

which leads to that

$$\sum_{i=1}^{m}\|(\mathbf{S}^*-\mathbf{S}^{(i)})\odot\mathbf{W}^*\|_F\le\sum_{i=1}^{m}\|(\tilde{\mathbf{S}}-\mathbf{S}^{(i)})\odot\tilde{\mathbf{W}}\|_F.$$

This completes the proof. □

According to Theorem 2, we relax (6) as

$$\begin{aligned}\min_{\mathbf{S},\mathbf{W},\boldsymbol{\alpha}}\ &\sum_{i=1}^{m}\frac{1}{\alpha_i}\|(\mathbf{S}-\mathbf{S}^{(i)})\odot\mathbf{W}\|_F^2-\lambda\|\mathbf{W}\|_1+\gamma\|\mathbf{S}\|_0\\\text{s.t. }&\mathbf{S}\odot\boldsymbol{\Omega}=\hat{\mathbf{S}}\odot\boldsymbol{\Omega},\quad 0\le S_{pq}\le 1\quad\forall p,q,\\&\operatorname{rank}(\mathbf{L})=n-c,\\&0\le W_{pq}\le 1\quad\forall p,q.\\&0\le\alpha_i\le 1,\sum_{i=1}^{m}\alpha_i=1.\end{aligned}\tag{9}$$

Then, we handle the rank function. According to [56], since we wish the rank of $\mathbf{L}$ is $n-c$, i.e., the c smallest eigenvalues of $\mathbf{L}$ are 0s, we try to minimize $\sum_{i=1}^{c}\sigma_i(\mathbf{L})$, where $\sigma_i(\mathbf{L})$ denotes the ith smallest eigenvalues of $\mathbf{L}$. According to Ky Fan's theorem [57], we have $\sum_{i=1}^{c}\sigma_i(\mathbf{L})=\min_{\mathbf{Y}\in\mathbb{R}^{n\times c},\mathbf{Y}^T\mathbf{Y}=\mathbf{I}}\operatorname{tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y})$. Thus, by introducing orthogonal matrix $\mathbf{Y}\in\mathbb{R}^{n\times c}$, we can reformulate (9) as follows:

$$\begin{aligned}\min_{\mathbf{S},\mathbf{W},\boldsymbol{\alpha},\mathbf{Y}}\ &\sum_{i=1}^{m}\frac{1}{\alpha_i}\|(\mathbf{S}-\mathbf{S}^{(i)})\odot\mathbf{W}\|_F^2\\&-\lambda\|\mathbf{W}\|_1+\gamma\|\mathbf{S}\|_0+\rho\operatorname{tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y})\\\text{s.t. }&\mathbf{S}\odot\boldsymbol{\Omega}=\hat{\mathbf{S}}\odot\boldsymbol{\Omega},\quad 0\le S_{pq}\le 1\quad\forall p,q,\\&0\le W_{pq}\le 1\quad\forall p,q\\&0\le\alpha_i\le 1,\quad\sum_{i=1}^{m}\alpha_i=1\\&\mathbf{Y}^T\mathbf{Y}=\mathbf{I}\end{aligned}\tag{10}$$

where ρ is a large enough parameter to make sure that the rank of $\mathbf{L}$ is $n-c$. Note that different from some conventional ensemble methods that linearly combine all base connective matrices, our $\mathbf{S}$ is more like a nonparametric consensus matrix, which is more flexible and can effectively enlarge the region from which an optimal consensus matrix can be chosen. The similar results can also be found in some previous research, such as in [58]–[60].

D. Optimization

Now, we introduce how to optimize (10). Since (10) involves four groups of variables, we will use a block coordinate descent schema to optimize it. More specifically, we will optimize one group of variables while fixing the other variables and repeat this procedure until it converges.

1) Optimize $\mathbf{W}$*:* We remove the terms that are irrelative to $\mathbf{W}$ and obtain

$$\begin{aligned}\min_{\mathbf{W}}\ &\sum_{i=1}^{m}\frac{1}{\alpha_i}\|\mathbf{A}^{(i)}\odot\mathbf{W}\|_F^2-\lambda\|\mathbf{W}\|_1\\\text{s.t. }&0\le W_{pq}\le 1\quad\forall p,q\end{aligned}\tag{11}$$

where $\mathbf{A}^{(i)}=\mathbf{S}-\mathbf{S}^{(i)}$.

It is easy to verify that (11) can be decoupled into $n\times n$ independent subproblems. Considering one of them

$$\begin{aligned}\min_{W_{pq}}\ &B_{pq}W_{pq}^2-\lambda W_{pq}\\\text{s.t. }&0\le W_{pq}\le 1\end{aligned}\tag{12}$$

where $B_{pq}=\sum_{i=1}^{m}(A_{pq}^2/\alpha_i)$.

Setting the partial derivative of (12) with respect to W_{pq} to zero, we obtain that $W_{pq}=(\lambda/2B_{pq})$. Since $B_{pq}\ge 0$, $W_{pq}\ge 0$. If $(\lambda/2B_{pq})>1$, i.e., $B_{pq}W_{pq}^2-\lambda W_{pq}$ decreases monotonically in the range $[0,1]$, so the optima is 1. To sum up, we optimize W_{pq} as

$$W_{pq}=\min\left(\frac{\lambda}{2B_{pq}},1\right).\tag{13}$$

From (13), we see that λ corresponds to the "age" of the model. When λ is small, the weight of the most samples is small, and only easy samples with small losses (B_{pq}) influence the model much. As λ grows, more samples with large losses will gradually influence the model. This accords with the motivation of self-paced learning.

2) Optimize $\mathbf{S}$*:* When other variables are fixed, we rewrite (10) as

$$\begin{aligned}\min_{\mathbf{S}}\ &\sum_{i=1}^{m}\frac{1}{\alpha_i}\|(\mathbf{S}-\mathbf{S}^{(i)})\odot\mathbf{W}\|_F^2+\gamma\|\mathbf{S}\|_0\\&+\rho\sum_{p,q=1}^{n}\|\mathbf{y}_p-\mathbf{y}_q\|_2^2S_{pq}\\\text{s.t. }&\mathbf{S}\odot\boldsymbol{\Omega}=\hat{\mathbf{S}}\odot\boldsymbol{\Omega},\quad 0\le S_{pq}\le 1\quad\forall p,q\end{aligned}\tag{14}$$

where $\mathbf{y}_p$ and $\mathbf{y}_q$ are the pth and qth row vectors in $\mathbf{Y}$, respectively.

Define a function $g(x)$ where $g(x)=1$ if $x\ne 0$ and $g(x)=0$ otherwise. Equation (14) can also be decoupled into $n\times n$ independent subproblems. Since $\mathbf{S}\odot\boldsymbol{\Omega}=\hat{\mathbf{S}}\odot\boldsymbol{\Omega}$, we just need to consider S_{pq} whose $\boldsymbol{\Omega}_{pq}=0$

$$\begin{aligned}\min_{S_{pq}}\ &\sum_{i=1}^{m}\frac{1}{\alpha_i}\left(S_{pq}-S_{pq}^{(i)}\right)^2W_{pq}^2+\gamma\,g(S_{pq})+\rho\|\mathbf{y}_p-\mathbf{y}_q\|_2^2S_{pq}\\\text{s.t. }&0\le S_{pq}\le 1.\end{aligned}\tag{15}$$

Equation (15) can be simplified further as

$$\min_{S_{pq}} (S_{pq} - C_{pq})^2 + \tau\ g(S_{pq}) \quad \text{s.t. } 0 \le S_{pq} \le 1 \tag{16}$$

where

$$C_{pq} = \frac{\sum_{i=1}^{m} \frac{S_{pq}^{(i)}}{\alpha_i} - \frac{\rho \|\mathbf{y}_p - \mathbf{y}_q\|_2^2}{2W_{pq}^2}}{\sum_{i=1}^{m} \frac{1}{\alpha_i}}$$

and $\tau = (\gamma / (\sum_{i=1}^{m} (W_{pq}^2 / \alpha_i)))$.

Equation (16) has a closed-form solution

$$S_{pq} = \begin{cases} 1, & \text{if } C_{pq} \ge 1 \\ C_{pq}, & \text{if } \sqrt{\tau} \le C_{pq} < 1 \\ 0, & C_{pq} < \sqrt{\tau}. \end{cases} \tag{17}$$

3) Optimize $\mathbf{Y}$: When optimizing $\mathbf{Y}$, we have

$$\min_{\mathbf{Y}} \operatorname{tr}(\mathbf{Y}^T \mathbf{L} \mathbf{Y}) \quad \text{s.t. } \mathbf{Y}^T \mathbf{Y} = \mathbf{I}. \tag{18}$$

According to Ky Fan's theorem [57], the solution of $\mathbf{Y}$ is formed by the c eigenvectors of $\mathbf{L}$ corresponding to the c smallest eigenvalues.

4) Optimize $\boldsymbol{\alpha}$: Let d_i denote $\|(\mathbf{S} - \mathbf{S}^{(i)}) \odot \mathbf{W}\|_F^2$, and we have

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{m} \frac{d_i}{\alpha_i} \quad \text{s.t. } 0 \le \alpha_i \le 1, \quad \sum_{i=1}^{m} \alpha_i = 1. \tag{19}$$

According to the Cauchy–Schwarz inequality, we have

$$\sum_{i=1}^{m} \frac{d_i}{\alpha_i} = \left(\sum_{i=1}^{m} \frac{d_i}{\alpha_i}\right)\left(\sum_{i=1}^{m} \alpha_i\right) \ge \left(\sum_{i=1}^{m} \sqrt{d_i}\right)^2. \tag{20}$$

The equality in (20) holds when $\alpha_i \propto \sqrt{d_i}$. Thus, the closed-form solution of (19) is

$$\alpha_i = \frac{\sqrt{d_i}}{\sum_{j=1}^{m} \sqrt{d_j}}. \tag{21}$$

E. Discussion

In this section, we first introduce how to initialize the variables involved in our objective function and then discuss how to choose the hyperparameter; at last, we discuss the relations and differences of our method and robust clustering ensemble methods.

We initialize $\mathbf{S} = \sum_{i=1}^{m} (1/m) \mathbf{S}^{(i)}$ and construct $\mathbf{L}$ from $\mathbf{S}$. Then, we initialize $\mathbf{Y}$ by solving (18). We set $\rho = 1$ at first and adjust it automatically by observing the rank of $\mathbf{L}$. We initialize $\alpha_i = (1/m)$.

We initialize $\mathbf{W}$ by (13). However, in (13), we need to decide λ first. In the initialization, we have set $\mathbf{S}$ as mean of $\mathbf{S}^{(i)}$ and $\alpha_i = (1/m)$, and then, we take a closer look at B_{pq}. Suppose that for the pair $(\mathbf{x}_p, \mathbf{x}_q)$, there are k clustering results agrees that they should belong to a cluster and the other $m-k$ results agrees that they belong to different clusters. Then, we have

$$B_{pq} = \sum_{i=1}^{m} \frac{(S_{pq} - S_{pq}^{(i)})^2}{\alpha_i} = \left(\left(\frac{k}{m} - 1\right)^2 k + \left(\frac{k}{m}\right)^2 (m-k)\right) m. \tag{22}$$

Let r denote $r = (k/m)$, we get $B_{pq} = ((r-1)^2 r + r^2(1-r))m^2$. Obviously, r indicates how many results agree with each other. For example, $r = 0.9$ means that 90% results agrees that the pair belongs to the same cluster. Thus, the larger r is, the easier the pair is. In our method, we initialize $r = 0.9$, and set λ

$$\lambda = 2B_{pq} = 2((r-1)^2 r + r^2(1-r))m^2 \tag{23}$$

which means for $\mathbf{x}_p$ and $\mathbf{x}_q$, if 90% clustering results agrees with each other, then we set $W_{pq} = 1$, i.e., we use this pair completely.

In the following learning, we gradually increase λ by decreasing r from 0.9 to 0.5.

Then, we discuss how to choose the hyperparameter γ. As we know, γ controls the sparsity of $\mathbf{S}$. From (17), we find that if $C_{pq} < \sqrt{\tau}$, we have $S_{pq} = 0$ and γ is proportional to τ; thus, γ plays a role as a threshold. More specifically, $S_{pq} = 0$ when

$$C_{pq} < \sqrt{\tau} = \sqrt{\frac{\gamma}{\sum_{i=1}^{m} \frac{W_{pq}^2}{\alpha_i}}} \approx \sqrt{\frac{\gamma}{\sum_{i=1}^{m} \frac{1}{\alpha_i}}}. \tag{24}$$

The approximate equals sign is due to that, at last, almost all pairs are involved in the learning; thus, all $W_{pq} \approx 1$. Then, according to the harmonic mean inequality, we have

$$C_{pq} < \sqrt{\frac{\gamma}{\sum_{i=1}^{m} \frac{1}{\alpha_i}}} \le \sqrt{\frac{\gamma \sum_{i=1}^{m} \alpha_i}{m^2}} = \frac{\sqrt{\gamma}}{m}. \tag{25}$$

Denote $\theta = (\sqrt{\gamma}/m)$. θ can be viewed as a threshold, i.e., S_{pq} is nonzero when $C_{pq} > \theta$. Therefore, we can easily give a threshold θ and compute γ by $\gamma = m^2\theta^2$ instead of directly setting γ. For example, if we wish to keep S_{pq} nonzero when $S_{pq} > 0.5$, we can easily set $\theta = 0.5$ and obtain $\gamma = m^2\theta^2 = 0.25\ m^2$.

Last but not least, it is worthy to discuss another related method here, called the robust clustering ensemble, which focuses on the robustness of the clustering ensemble methods. It captures the noises from data or base clustering results and recovers clean results for the ensemble. For example, Zhou *et al.* [39] learned a robust consensus clustering result via minimizing the Kullback–Leibler divergence among each base result; Tao *et al.* [40], [61] presented robust clustering ensemble methods based on spectral clustering; Huang *et al.* [62] applied probability trajectories to robust clustering ensemble; and Liu *et al.* [63] proposed an ensemble method on incomplete data. In these methods, they only focus on the noises or outliers without distinguishing between uncontaminated instances. In our method, the contaminated instances can be viewed as the most difficult instances

because they may contribute nothing to the learning. Besides, the uncontaminated instances can also be handled in order of difficulty. Therefore, our method handles instances more finely. Moreover, the difficulty of instances is always changing in the process of learning, i.e., most instances become increasingly easier as time passes by. In our method, we estimate the difficulty of instances ($\mathbf{W}$) automatically in the process of the ensemble. From (13), we observe that $\mathbf{W}$ is proportional to λ, i.e., when time goes on (λ increases), $\mathbf{W}$ will also increases until it reaches 1. Therefore, this property can be well characterized by our method.

F. Whole Algorithm

Algorithm 1 summarizes the whole process of the SPCE method. Note that in the inner iteration (Lines 6–17), the algorithm optimizes $\mathbf{S}$, $\mathbf{Y}$, and $\boldsymbol{\alpha}$. Since the solution of each step is the global optima of the corresponding subproblem, which makes the objective function decrease monotonically and the objective function has a lower bound, the iteration will always converge.

Algorithm 1 SPCE

Input: m connective matrices $\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(m)}$, number c of clusters, threshold θ.
Output: Final clusters.

1: Construct $\hat{\mathbf{S}}$ by Eq.(2) and construct $\boldsymbol{\Omega}$ from $\hat{\mathbf{S}}$.
2: Initialize the parameters as introduced in Section III-E.
3: **for** $r = 0.9, 0.8, \cdots, 0.5$ **do**
4: Compute λ by Eq.(23).
5: Compute $\mathbf{W}$ by Eq.(13).
6: **while** not converge **do**
7: Compute $\mathbf{S}$ by Eq.(17).
8: Compute $\mathbf{Y}$ by solving Eq.(18).
9: Compute $\boldsymbol{\alpha}$ by Eq.(21).
10: **if** The rank of $\mathbf{L}$ is larger than $n - c$ **then**
11: $\rho = 2\rho$.
12: **else if** The rank of $\mathbf{L}$ is smaller than $n - c$ **then**
13: $\rho = \rho/2$.
14: **else**
15: Break.
16: **end if**
17: **end while**
18: **end for**
19: Obtain the final clusters from the c connective component in $\mathbf{S}$.

G. Time and Space Complexity

Since we need to save and handle m connective matrices $\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(m)}$, the space complexity is $O(mn^2)$.

When computing $\mathbf{W}$, we need to compute $\mathbf{A}^{(i)}$ ($i = 1, \ldots, m$); thus, the time complexity is $O(mn^2)$. Computing $\mathbf{S}$ costs $O(n^2c + n^2m)$ since we need to compute $\mathbf{C}$ first. Computing $\mathbf{Y}$ involves an eigenvector decomposition that costs $O(n^2c)$ time. When computing $\boldsymbol{\alpha}$, we need to compute $\mathbf{d}$ in $O(n^2m)$ time. Therefore, the whole time complexity is $O((n^2m + n^2c)t)$, where t is the number of iterations.

TABLE II
DESCRIPTION OF THE DATA SETS

	#instances	#features	#classes
AR	840	768	120
Coil20	1440	1024	20
GLIOMA	50	4434	4
K1b	2340	21839	6
Lung	203	3312	5
Medical	706	1449	17
Tr41	878	7454	10
Tdt2	10212	36771	96
TOX	171	5748	4
UMIST	575	644	20
WebACE	2340	1000	20
WarpAR	130	2400	10

The time and space complexity of our method is comparable with the existing connective/coassociation matrix-based methods [39]–[41]. Despite this, we plan to reduce the computation complexity of the proposed method in future work.

IV. EXPERIMENTS

In this section, we compare our SPCE with several state-of-the-art clustering ensemble methods on benchmark data sets.

A. Data Sets

We use totally 12 data sets to evaluate the effectiveness of our proposed SPCE, including AR [64], Coil20 [65], GLIOMA [66], K1b [67], Lung [68], Medical [39], Tr41 [67], Tdt2 [69], TOX [66], UMIST [70], WebACE [71], and WarpAR [66]. Data sets from different areas serve as a good test bed for a comprehensive evaluation. The basic information of these data sets is summarized in Table II.

B. Compared Methods

We compare our SPCE with the following algorithms.

1) *KM/SC:* These are the average of all the base k-means and spectral clustering results, respectively.
2) *KM-best/SC-best:* These are the best result of all the base k-means and spectral clustering results, respectively.
3) *KC:* It represents the results of applying k-means to a consensus similarity matrix, and it is often used as a baseline in clustering ensemble methods, such as [24], [39].
4) *Cluster-Based Similarity Partitioning Algorithm (CSPA) [3]:* It signifies a relationship between objects in the same cluster and can, thus, be used to establish a measure of pairwise similarity.
5) *Hypergraph Partitioning Algorithm (HGPA) [3]:* It approximates the maximum mutual information objective with a constrained minimum cut objective.
6) *Metaclustering Algorithm (MCLA) [3]:* It transforms the integration into a cluster correspondence problem.

TABLE III
ACC RESULTS ON ALL THE DATA SETS (k-MEANS BASED)

Methods	AR	Coil20	GLIOMA	K1b	Lung	Medical	Tr41	Tdt2	Tox	UMIST	WebACE	WarpAR
KM	0.290 ±0.011	0.547 ±0.047	0.424 ±0.035	0.673 ±0.098	0.711 ±0.094	0.400 ±0.036	0.563 ±0.072	0.410 ±0.019	0.423 ±0.032	0.397 ±0.021	0.361 ±0.037	0.245 ±0.034
KM-best	0.310 ±0.005	0.635 ±0.021	0.488 ±0.019	0.856 ±0.025	0.867 ±0.037	0.471 ±0.027	0.695 ±0.047	0.446 ±0.008	0.482 ±0.015	0.436 ±0.009	0.419 ±0.013	0.319 ±0.020
KC	0.317 ±0.010	0.580 ±0.072	0.410 ±0.029	0.561 ±0.049	0.593 ±0.122	0.361 ±0.028	0.644 ±0.032	0.320 ±0.024	0.425 ±0.031	0.375 ±0.027	0.333 ±0.039	0.230 ±0.016
CSPA [3]	0.331 ±0.003	0.638 ±0.024	0.410 ±0.027	0.453 ±0.003	0.414 ±0.015	0.350 ±0.015	0.521 ±0.028	0.285 ±0.005	0.425 ±0.037	0.407 ±0.017	0.277 ±0.009	0.238 ±0.016
HGPA [3]	0.331 ±0.007	0.550 ±0.028	0.418 ±0.039	0.533 ±0.047	0.503 ±0.033	0.295 ±0.028	0.489 ±0.055	0.296 ±0.005	0.385 ±0.029	0.403 ±0.026	0.259 ±0.019	0.252 ±0.027
MCLA [3]	0.334 ±0.007	0.663 ±0.035	0.400 ±0.013	0.738 ±0.091	0.708 ±0.048	0.402 ±0.020	0.570 ±0.039	0.400 ±0.009	0.415 ±0.024	0.405 ±0.014	0.281 ±0.063	0.232 ±0.013
NMFC [24]	0.332 ±0.009	0.646 ±0.032	0.414 ±0.021	0.586 ±0.035	0.676 ±0.108	0.379 ±0.018	0.632 ±0.037	0.372 ±0.017	0.427 ±0.023	0.398 ±0.014	0.346 ±0.028	0.223 ±0.029
BCE [28]	0.008 ±0.000	0.629 ±0.057	0.428 ±0.017	0.635 ±0.067	0.670 ±0.133	0.397 ±0.043	0.621 ±0.049	0.181 ±0.000	0.414 ±0.019	0.395 ±0.012	0.358 ±0.025	0.229 ±0.027
RCE [39]	0.331 ±0.010	0.630 ±0.024	0.426 ±0.010	0.689 ±0.037	0.714 ±0.071	0.385 ±0.030	0.639 ±0.023	-	0.411 ±0.026	0.406 ±0.015	0.354 ±0.022	0.205 ±0.020
MEC [41]	0.280 ±0.014	0.573 ±0.046	0.394 ±0.037	0.819 ±0.090	0.738 ±0.124	0.363 ±0.017	0.656 ±0.044	-	0.430 ±0.031	0.413 ±0.028	0.376 ±0.034	0.259 ±0.037
LWEA [72]	0.313 ±0.010	0.594 ±0.019	0.432 ±0.014	0.828 ±0.076	0.746 ±0.096	0.421 ±0.008	0.672 ±0.047	0.574 ±0.027	0.423 ±0.013	0.405 ±0.011	0.332 ±0.039	0.213 ±0.029
LWGP [72]	0.332 ±0.008	0.655 ±0.029	0.432 ±0.010	0.717 ±0.077	0.650 ±0.050	0.405 ±0.014	0.648 ±0.034	0.429 ±0.010	0.419 ±0.026	0.418 ±0.018	0.346 ±0.031	0.229 ±0.021
RSEC [61]	0.286 ±0.010	0.536 ±0.077	0.418 ±0.050	0.841 ±0.051	0.822 ±0.052	0.349 ±0.030	0.637 ±0.044	0.422 ±0.077	0.404 ±0.024	0.313 ±0.063	0.310 ±0.052	0.242 ±0.029
DREC [73]	0.331 ±0.009	0.546 ±0.032	0.428 ±0.010	0.646 ±0.065	0.638 ±0.072	0.393 ±0.020	0.624 ±0.027	0.368 ±0.004	0.421 ±0.041	0.417 ±0.014	0.343 ±0.026	0.206 ±0.016
SPCE-W	0.262 ±0.014	0.582 ±0.067	0.412 ±0.039	0.845 ±0.031	**0.909** ±0.008	0.431 ±0.035	0.663 ±0.100	0.580 ±0.050	0.384 ±0.066	0.375 ±0.012	0.331 ±0.058	0.218 ±0.041
SPCE-fixW	0.348 ±0.007	0.689 ±0.024	0.436 ±0.016	0.856 ±0.020	**0.907** ±0.018	0.436 ±0.019	0.691 ±0.098	0.603 ±0.061	0.426 ±0.028	0.412 ±0.022	0.345 ±0.048	0.262 ±0.014
SPCE	**0.360** ±0.007	**0.700** ±0.015	**0.442** ±0.020	**0.866** ±0.020	**0.913** ±0.005	**0.453** ±0.013	**0.735** ±0.076	**0.665** ±0.074	**0.449** ±0.019	**0.434** ±0.022	**0.393** ±0.017	**0.283** ±0.013

7) *Nonnegative Matrix Factorization-Based Consensus Clustering (NMFC) [24]:* It uses NMF to aggregate clustering results.
8) *Bayesian Clustering Ensemble (BCE) [28]:* It is a Bayesian model for ensemble.
9) *Robust Clustering Ensemble (RCE) [39]:* It learns a robust consensus clustering result via minimizing the Kullback–Leibler divergence among each base result.
10) *Multiview Ensemble Clustering (MEC) [41]:* It is a robust multiview clustering ensemble method using low-rank and sparse decomposition to ensemble base clustering and detect the noises.
11) *Locally Weighted Evidence Accumulation (LWEA) [72]:* It is a hierarchical agglomerative clustering ensemble method based on ensemble-driven cluster uncertainty estimation and local weighting strategy.
12) *Locally Weighted Graph Partitioning (LWGP) [72]:* It is a graph partition method based on the local weighting strategy.
13) *Robust Spectral Ensemble Clustering (RSEC) [61]:* It is a robust clustering ensemble method based on spectral clustering.
14) *Dense Representation Ensemble Clustering (DREC) [73]:* It learns a dense representation for clustering ensemble.
15) *SPCE-W:* It is our method without $\mathbf{W}$. To evaluate the effectiveness of self-paced learning in our method, we also run SPCE-W to see the performance of our method without self-paced learning, i.e., in each iteration, all weights in $\mathbf{W}$ are fixed to 1s.
16) *SPCE-fixW:* It is our method with fixed $\mathbf{W}$, where W_{ij} is proportional to the frequency of the two instances occurring in the clusters from the given base clusterings. To further evaluate the effectiveness of self-paced learning in our method, we also run SPCE-fixW with fixed $\mathbf{W}$, i.e., in each iteration, $\mathbf{W}$ is fixed as the initial value as introduced in Section III-E instead of learning automatically.

C. Experimental Setup

We conduct two groups of experiments that use k-means and spectral clustering results as base clusterings, respectively. In the k-means-based clustering ensemble, following the similar experimental protocol in [28], [39], we run k-means 200 times with different initializations on all instances to obtain 200 base clustering results that are divided evenly into ten subsets, with 20 base results in each subset. Then, we apply all clustering ensemble methods on each subset and report the average results on the ten subsets. In spectral-based clustering ensemble, we use the Gaussian kernel $k(\mathbf{x}_i, \mathbf{x}_j) = e^{-((\|\mathbf{x}_i - \mathbf{x}_j\|_2^2)/2\sigma^2)}$ to construct the Laplacian matrix for spectral clustering. σ is the bandwidth parameter, and in our experiments, we set $\sigma = d * \{0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100\}$ for ensemble, i.e., we

TABLE IV

NMI RESULTS ON ALL THE DATA SETS (k-MEANS BASED)

Methods	AR	Coil20	GLIOMA	K1b	Lung	Medical	Tr41	Tdt2	Tox	UMIST	WebACE	WarpAR
KM	0.648 ±0.008	0.706 ±0.025	0.163 ±0.039	0.549 ±0.061	0.528 ±0.060	0.421 ±0.029	0.584 ±0.051	0.611 ±0.007	0.137 ±0.040	0.585 ±0.019	0.357 ±0.027	0.210 ±0.046
KM-best	0.661 ±0.004	0.753 ±0.009	0.235 ±0.023	0.685 ±0.030	0.656 ±0.062	0.481 ±0.019	0.671 ±0.025	0.624 ±0.006	0.216 ±0.027	0.617 ±0.009	0.409 ±0.013	0.296 ±0.018
KC	0.677 ±0.006	0.732 ±0.031	0.152 ±0.029	0.500 ±0.023	0.454 ±0.072	0.426 ±0.020	0.654 ±0.028	0.580 ±0.009	0.134 ±0.017	0.568 ±0.030	0.376 ±0.019	0.195 ±0.030
CSPA [3]	0.698 ±0.002	0.737 ±0.018	0.172 ±0.028	0.407 ±0.007	0.371 ±0.019	0.399 ±0.013	0.592 ±0.015	0.559 ±0.003	0.144 ±0.045	0.581 ±0.014	0.347 ±0.016	0.204 ±0.023
HGPA [3]	0.693 ±0.004	0.671 ±0.024	0.151 ±0.036	0.392 ±0.074	0.337 ±0.048	0.361 ±0.033	0.508 ±0.035	0.539 ±0.008	0.108 ±0.021	0.582 ±0.023	0.266 ±0.022	0.218 ±0.038
MCLA [3]	0.690 ±0.007	0.761 ±0.017	0.133 ±0.029	0.594 ±0.070	0.526 ±0.017	0.430 ±0.019	0.604 ±0.024	0.607 ±0.004	0.133 ±0.017	0.596 ±0.008	0.186 ±0.162	0.185 ±0.029
NMFC [24]	0.685 ±0.004	0.760 ±0.010	0.155 ±0.027	0.500 ±0.021	0.520 ±0.066	0.426 ±0.019	0.651 ±0.019	0.593 ±0.004	0.143 ±0.029	0.584 ±0.018	0.398 ±0.022	0.197 ±0.031
BCE [28]	0.000 ±0.000	0.751 ±0.024	0.166 ±0.022	0.541 ±0.030	0.498 ±0.067	0.450 ±0.025	0.640 ±0.036	0.000 ±0.000	0.137 ±0.024	0.581 ±0.011	0.396 ±0.011	0.200 ±0.030
RCE [39]	0.676 ±0.004	0.759 ±0.012	0.162 ±0.016	0.607 ±0.010	0.525 ±0.032	0.448 ±0.019	0.650 ±0.018	-	0.134 ±0.020	0.595 ±0.012	0.401 ±0.013	0.178 ±0.026
MEC [41]	0.602 ±0.017	0.736 ±0.026	0.131 ±0.043	**0.682** ±0.071	0.562 ±0.087	0.409 ±0.029	**0.676** ±0.027	-	0.131 ±0.031	0.562 ±0.025	0.397 ±0.030	0.206 ±0.030
LWEA [72]	0.664 ±0.006	0.738 ±0.011	0.169 ±0.021	**0.695** ±0.065	0.536 ±0.062	0.419 ±0.015	0.667 ±0.039	**0.718** ±0.009	0.124 ±0.029	0.606 ±0.013	0.330 ±0.009	0.184 ±0.028
LWGP [72]	0.678 ±0.006	0.764 ±0.013	0.168 ±0.018	0.612 ±0.049	0.499 ±0.023	0.427 ±0.011	0.654 ±0.028	0.627 ±0.005	0.133 ±0.028	0.609 ±0.014	0.387 ±0.014	0.195 ±0.025
RSEC [61]	0.590 ±0.015	0.702 ±0.045	0.154 ±0.046	0.662 ±0.053	0.603 ±0.062	0.404 ±0.049	0.645 ±0.048	0.524 ±0.034	0.118 ±0.014	0.421 ±0.103	0.320 ±0.110	0.188 ±0.028
DREC [73]	0.677 ±0.004	0.722 ±0.017	0.164 ±0.019	0.577 ±0.041	0.465 ±0.039	**0.451** ±0.020	0.651 ±0.017	0.597 ±0.001	0.139 ±0.028	0.604 ±0.008	**0.428** ±0.011	0.184 ±0.019
SPCE-W	0.556 ±0.019	0.711 ±0.050	0.141 ±0.024	0.560 ±0.115	**0.733** ±0.016	0.333 ±0.040	0.609 ±0.135	0.693 ±0.059	0.104 ±0.053	0.542 ±0.014	0.192 ±0.105	0.259 ±0.141
SPCE-fixW	0.737 ±0.002	**0.798** ±0.019	0.176 ±0.022	0.671 ±0.073	**0.730** ±0.037	0.355 ±0.022	0.623 ±0.115	0.661 ±0.051	0.129 ±0.022	0.645 ±0.025	0.232 ±0.107	0.416 ±0.019
SPCE	**0.740** ±0.001	**0.805** ±0.010	**0.301** ±0.015	**0.699** ±0.060	**0.742** ±0.010	**0.455** ±0.013	**0.685** ±0.059	**0.713** ±0.043	**0.196** ±0.014	**0.663** ±0.014	0.408 ±0.049	**0.436** ±0.013

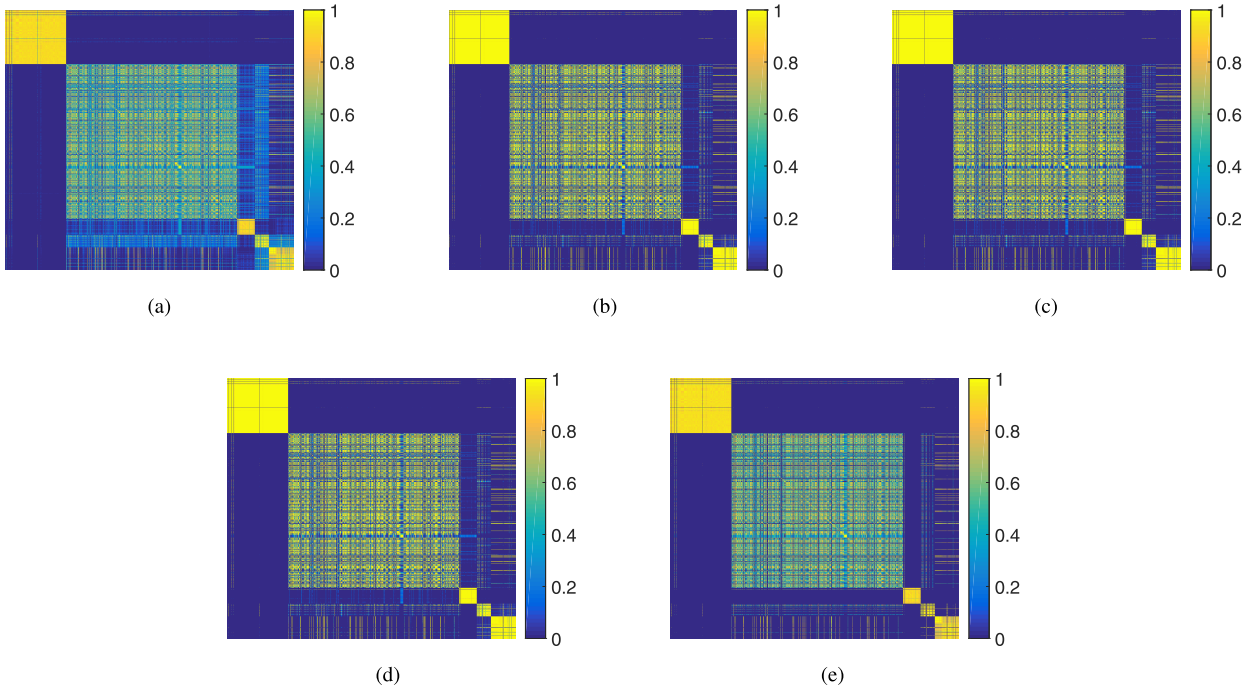


Fig. 1. Illustration of clustering structure in the learned consensus matrices from different methods on the K1b data set. (a) Input coassociation matrix. (b)–(d) Learned consensus matrices by robust methods. (f) $\mathbf{S}$ in our method. Note that the second cluster and the third cluster are separated more clearly in our method and so are the fourth cluster and the fifth cluster. Thus, the consensus matrix of our method is cleaner than those robust methods. (a) Input coassociation matrix $\hat{\mathbf{S}}$. (b) RCE. (c) MEC. (d) RSEC. (e) SPCE.

ensemble nine spectral clustering results, where d is the mean distance between all pairs $(\mathbf{x}_i, \mathbf{x}_j)$. We also repeat it ten times and report the average results. To measure the clustering results, the widely used accuracy (ACC), NMI, and adjusted rand index (ARI) are reported. To validate the statistic significance of results, we also calculate the p-value of t-test.

TABLE V

ARI RESULTS ON ALL THE DATA SETS (k-MEANS BASED)

Methods	AR	Coil20	GLIOMA	K1b	Lung	Medical	Tr41	Tdt2	Tox	UMIST	WebACE	WarpAR
KM	0.118 ±0.008	0.500 ±0.040	0.078 ±0.041	0.476 ±0.121	0.444 ±0.123	0.237 ±0.031	0.431 ±0.090	0.211 ±0.014	0.102 ±0.038	0.279 ±0.021	0.135 ±0.039	0.042 ±0.028
KM-best	0.133 ±0.005	0.576 ±0.014	0.160 ±0.031	0.718 ±0.043	0.686 ±0.077	0.306 ±0.034	0.592 ±0.058	0.239 ±0.010	0.179 ±0.031	0.318 ±0.015	0.210 ±0.025	0.099 ±0.012
KC	0.147 ±0.008	0.545 ±0.058	0.083 ±0.021	0.376 ±0.029	0.348 ±0.147	0.218 ±0.026	0.544 ±0.043	0.173 ±0.016	0.107 ±0.018	0.272 ±0.035	0.183 ±0.058	0.027 ±0.013
CSPA [3]	0.159 ±0.005	0.565 ±0.029	0.077 ±0.018	0.255 ±0.004	0.190 ±0.012	0.195 ±0.011	0.427 ±0.019	0.142 ±0.001	0.111 ±0.039	0.295 ±0.021	0.190 ±0.009	0.030 ±0.015
HGPA [3]	0.169 ±0.007	0.450 ±0.031	0.057 ±0.027	0.290 ±0.063	0.179 ±0.060	0.152 ±0.028	0.341 ±0.048	0.146 ±0.004	0.075 ±0.017	0.286 ±0.025	0.147 ±0.013	0.043 ±0.023
MCLA [3]	0.160 ±0.008	0.600 ±0.031	0.080 ±0.011	0.552 ±0.124	0.427 ±0.038	0.230 ±0.023	0.451 ±0.042	0.191 ±0.009	0.105 ±0.021	0.298 ±0.015	0.126 ±0.112	0.027 ±0.014
NMFC [24]	0.155 ±0.006	0.592 ±0.021	0.078 ±0.025	0.369 ±0.030	0.427 ±0.119	0.223 ±0.021	0.527 ±0.026	0.179 ±0.006	0.113 ±0.026	0.287 ±0.019	0.180 ±0.048	0.031 ±0.014
BCE [28]	0.000 ±0.000	0.579 ±0.044	0.085 ±0.018	0.423 ±0.059	0.401 ±0.124	**0.243** ±0.032	0.512 ±0.054	0.000 ±0.000	0.107 ±0.023	0.275 ±0.011	0.108 ±0.025	0.030 ±0.012
RCE [39]	0.152 ±0.004	0.594 ±0.016	0.092 ±0.007	0.514 ±0.030	0.437 ±0.072	0.234 ±0.023	0.536 ±0.023	-	0.099 ±0.022	0.292 ±0.012	0.113 ±0.027	0.024 ±0.015
MEC [41]	0.081 ±0.007	0.478 ±0.070	0.091 ±0.022	0.656 ±0.124	0.398 ±0.115	0.190 ±0.033	0.502 ±0.076	-	0.096 ±0.023	0.282 ±0.036	**0.197** ±0.074	0.028 ±0.011
LWEA [72]	0.147 ±0.006	0.559 ±0.017	0.096 ±0.011	0.697 ±0.114	0.468 ±0.131	**0.251** ±0.006	0.541 ±0.053	**0.434** ±0.002	0.104 ±0.018	0.311 ±0.011	0.099 ±0.040	0.024 ±0.011
LWGP [72]	0.155 ±0.008	0.618 ±0.031	0.094 ±0.007	0.523 ±0.112	0.368 ±0.043	**0.245** ±0.010	0.546 ±0.033	0.213 ±0.007	0.103 ±0.025	0.302 ±0.023	0.106 ±0.019	0.028 ±0.013
RSEC [61]	0.056 ±0.005	0.464 ±0.052	0.065 ±0.045	0.699 ±0.083	0.601 ±0.097	0.188 ±0.034	0.481 ±0.071	0.180 ±0.079	0.079 ±0.022	0.121 ±0.067	0.126 ±0.075	0.020 ±0.013
DREC [73]	0.153 ±0.004	0.529 ±0.030	0.092 ±0.007	0.463 ±0.066	0.341 ±0.061	**0.243** ±0.019	0.521 ±0.029	0.181 ±0.004	0.109 ±0.031	0.301 ±0.018	0.136 ±0.043	0.025 ±0.011
SPCE-W	0.067 ±0.011	0.505 ±0.064	0.080 ±0.035	0.632 ±0.104	**0.756** ±0.017	0.221 ±0.042	0.535 ±0.156	0.358 ±0.089	0.072 ±0.051	0.248 ±0.024	0.083 ±0.058	0.043 ±0.041
SPCE-fixW	0.177 ±0.010	0.622 ±0.033	0.099 ±0.012	0.727 ±0.060	0.748 ±0.048	0.234 ±0.012	0.575 ±0.147	0.325 ±0.079	0.097 ±0.025	0.339 ±0.052	0.097 ±0.062	**0.090** ±0.010
SPCE	**0.188** ±0.007	**0.646** ±0.022	**0.118** ±0.016	**0.743** ±0.060	**0.766** ±0.011	**0.252** ±0.009	**0.636** ±0.093	**0.420** ±0.079	**0.127** ±0.025	**0.357** ±0.032	0.170 ±0.027	**0.093** ±0.009

TABLE VI

ACC RESULTS ON ALL THE DATA SETS (SPECTRAL CLUSTERING BASED)

Methods	AR	Coil20	GLIOMA	K1b	Lung	Medical	Tr41	Tdt2	Tox	UMIST	WebACE	WarpAR
SC	0.271 ±0.061	0.490 ±0.167	0.408 ±0.040	0.779 ±0.138	0.691 ±0.139	0.317 ±0.083	0.520 ±0.173	0.336 ±0.112	0.395 ±0.080	0.371 ±0.118	0.250 ±0.045	0.234 ±0.023
SC-best	0.320 ±0.008	0.665 ±0.027	0.444 ±0.013	0.923 ±0.007	0.841 ±0.024	0.405 ±0.012	0.695 ±0.032	0.448 ±0.014	0.460 ±0.003	0.468 ±0.015	0.295 ±0.007	0.275 ±0.016
KC	0.221 ±0.013	0.636 ±0.057	0.422 ±0.020	0.804 ±0.063	0.589 ±0.125	0.331 ±0.023	0.598 ±0.072	0.325 ±0.028	0.449 ±0.022	0.398 ±0.015	0.250 ±0.012	0.231 ±0.015
CSPA [3]	0.223 ±0.009	0.720 ±0.032	**0.444** ±0.021	0.449 ±0.002	0.375 ±0.014	0.354 ±0.017	0.449 ±0.016	0.181 ±0.000	0.448 ±0.020	0.448 ±0.018	0.233 ±0.006	**0.256** ±0.015
HGPA [3]	0.316 ±0.007	0.582 ±0.057	0.420 ±0.061	0.502 ±0.064	0.467 ±0.061	0.327 ±0.028	0.392 ±0.043	0.228 ±0.014	0.354 ±0.035	0.455 ±0.047	0.194 ±0.012	**0.259** ±0.012
MCLA [3]	0.008 ±0.000	0.092 ±0.132	0.430 ±0.025	0.492 ±0.112	0.571 ±0.126	0.221 ±0.003	0.276 ±0.016	0.181 ±0.000	0.374 ±0.078	0.084 ±0.000	0.211 ±0.000	0.168 ±0.054
NMFC [24]	0.228 ±0.011	0.693 ±0.028	0.424 ±0.021	0.797 ±0.044	0.582 ±0.128	0.359 ±0.016	0.622 ±0.040	0.348 ±0.016	**0.455** ±0.014	0.426 ±0.011	0.227 ±0.015	0.229 ±0.018
BCE [28]	0.008 ±0.000	0.504 ±0.243	0.436 ±0.018	0.836 ±0.069	0.620 ±0.151	0.369 ±0.013	0.593 ±0.125	0.181 ±0.000	0.449 ±0.028	0.287 ±0.175	0.268 ±0.032	0.236 ±0.014
RCE [39]	**0.337** ±0.004	**0.737** ±0.018	0.428 ±0.017	0.894 ±0.048	0.739 ±0.170	0.387 ±0.009	0.642 ±0.052	-	**0.460** ±0.009	0.439 ±0.013	0.282 ±0.015	0.229 ±0.014
MEC [41]	0.178 ±0.006	0.644 ±0.045	0.432 ±0.019	0.640 ±0.052	0.530 ±0.117	0.362 ±0.018	0.609 ±0.052	-	**0.459** ±0.018	0.414 ±0.018	0.219 ±0.022	0.222 ±0.022
LWEA [72]	0.322 ±0.004	0.727 ±0.021	0.432 ±0.014	0.904 ±0.047	0.737 ±0.169	0.393 ±0.015	0.652 ±0.031	0.510 ±0.017	**0.458** ±0.003	0.454 ±0.014	0.274 ±0.006	0.231 ±0.015
LWGP [72]	0.328 ±0.006	0.725 ±0.012	0.432 ±0.014	0.899 ±0.050	0.741 ±0.167	0.389 ±0.013	0.661 ±0.017	0.491 ±0.013	**0.457** ±0.003	0.456 ±0.018	0.286 ±0.008	0.230 ±0.016
RSEC [61]	0.271 ±0.013	0.455 ±0.012	0.410 ±0.030	0.645 ±0.068	0.606 ±0.135	0.279 ±0.028	0.513 ±0.056	0.318 ±0.033	0.390 ±0.023	0.403 ±0.043	0.244 ±0.037	0.228 ±0.017
DREC [73]	**0.331** ±0.007	**0.733** ±0.015	0.432 ±0.014	0.909 ±0.045	0.736 ±0.174	0.389 ±0.008	0.636 ±0.032	0.418 ±0.011	**0.458** ±0.003	0.457 ±0.020	0.286 ±0.009	0.228 ±0.012
SPCE-W	0.286 ±0.034	0.594 ±0.098	0.424 ±0.016	0.752 ±0.110	0.779 ±0.078	0.358 ±0.087	0.606 ±0.066	0.366 ±0.043	**0.460** ±0.003	0.394 ±0.050	0.255 ±0.027	0.220 ±0.013
SPCE-fixW	0.296 ±0.024	0.715 ±0.036	0.422 ±0.015	0.887 ±0.014	0.798 ±0.126	0.387 ±0.023	0.664 ±0.033	0.466 ±0.048	0.448 ±0.034	0.438 ±0.025	0.269 ±0.021	0.229 ±0.010
SPCE	0.323 ±0.007	**0.741** ±0.017	0.431 ±0.012	**0.922** ±0.011	**0.838** ±0.033	**0.419** ±0.035	**0.689** ±0.037	**0.532** ±0.035	**0.460** ±0.003	**0.468** ±0.016	**0.303** ±0.015	**0.252** ±0.014

The number of clusters is set to the true number of classes for all data sets and algorithms. In our method, we set the parameter γ as $\gamma = m^2\theta^2$ (where m is the number of base clustering results), as introduced in Section III-E, and tune $\theta = \{0, 0.1, \ldots, 0.9\}$ to control the sparsity. Note that $\theta = 0$ means we drop the regularized term $\|\mathbf{S}\|_0$. We tune

TABLE VII
NMI RESULTS ON ALL THE DATA SETS (SPECTRAL CLUSTERING BASED)

Methods	AR	Coil20	GLIOMA	K1b	Lung	Medical	Tr41	Tdt2	Tox	UMIST	WebACE	WarpAR
SC	0.562 ±0.156	0.618 ±0.220	0.147 ±0.043	0.523 ±0.330	0.383 ±0.221	0.321 ±0.171	0.444 ±0.296	0.416 ±0.270	0.128 ±0.073	0.507 ±0.207	0.216 ±0.124	0.198 ±0.040
SC-best	0.674 ±0.005	0.792 ±0.013	0.190 ±0.012	0.814 ±0.009	0.622 ±0.015	0.464 ±0.006	0.684 ±0.020	0.620 ±0.008	0.202 ±0.005	0.656 ±0.008	0.331 ±0.008	0.251 ±0.015
KC	0.547 ±0.018	0.765 ±0.039	0.170 ±0.020	0.665 ±0.055	0.447 ±0.064	0.392 ±0.028	0.594 ±0.039	0.525 ±0.011	0.161 ±0.036	0.568 ±0.022	0.261 ±0.023	0.207 ±0.015
CSPA [3]	0.626 ±0.006	0.797 ±0.023	0.194 ±0.018	0.426 ±0.008	0.307 ±0.017	0.417 ±0.012	0.522 ±0.016	0.000 ±0.000	0.176 ±0.029	0.608 ±0.014	0.298 ±0.009	0.237 ±0.022
HGPA [3]	0.681 ±0.004	0.687 ±0.045	0.149 ±0.061	0.348 ±0.058	0.287 ±0.042	0.389 ±0.019	0.320 ±0.057	0.377 ±0.012	0.047 ±0.022	0.611 ±0.038	0.163 ±0.010	0.226 ±0.017
MCLA [3]	0.000 ±0.000	0.049 ±0.154	0.154 ±0.035	0.399 ±0.103	0.276 ±0.190	0.021 ±0.045	0.030 ±0.036	0.000 ±0.000	0.106 ±0.072	0.000 ±0.000	0.000 ±0.000	0.092 ±0.070
NMFC [24]	0.553 ±0.020	0.793 ±0.010	0.159 ±0.020	0.667 ±0.054	0.454 ±0.077	0.411 ±0.018	0.623 ±0.037	0.538 ±0.007	0.171 ±0.017	0.598 ±0.011	0.284 ±0.018	0.206 ±0.021
BCE [28]	0.000 ±0.000	0.605 ±0.320	0.181 ±0.026	0.714 ±0.060	0.486 ±0.086	0.431 ±0.011	0.574 ±0.205	0.000 ±0.000	0.166 ±0.034	0.364 ±0.313	0.272 ±0.102	0.203 ±0.018
RCE [39]	0.658 ±0.006	0.824 ±0.008	0.171 ±0.013	0.781 ±0.044	0.560 ±0.094	0.454 ±0.009	0.670 ±0.029	-	0.175 ±0.006	0.649 ±0.011	0.325 ±0.027	0.219 ±0.011
MEC [41]	0.499 ±0.011	0.768 ±0.024	0.176 ±0.013	0.558 ±0.045	0.434 ±0.075	0.410 ±0.018	0.600 ±0.049	-	0.167 ±0.023	0.589 ±0.020	0.224 ±0.024	0.176 ±0.036
LWEA [72]	0.681 ±0.004	0.824 ±0.006	0.173 ±0.010	**0.795** ±0.043	0.558 ±0.093	0.457 ±0.009	**0.683** ±0.014	0.651 ±0.009	0.173 ±0.001	0.658 ±0.010	0.277 ±0.015	0.215 ±0.010
LWGP [72]	0.677 ±0.003	0.825 ±0.004	0.173 ±0.010	0.788 ±0.050	0.560 ±0.096	0.456 ±0.011	**0.687** ±0.015	0.638 ±0.005	0.173 ±0.001	0.657 ±0.007	0.315 ±0.020	0.218 ±0.009
RSEC [61]	0.586 ±0.021	0.575 ±0.070	0.144 ±0.051	0.319 ±0.132	0.384 ±0.056	0.264 ±0.035	0.500 ±0.063	0.472 ±0.018	0.093 ±0.021	0.529 ±0.033	0.216 ±0.030	0.167 ±0.025
DREC [73]	0.683 ±0.004	**0.827** ±0.008	0.172 ±0.010	**0.798** ±0.043	0.559 ±0.095	0.455 ±0.009	0.658 ±0.028	0.607 ±0.005	0.173 ±0.001	0.654 ±0.011	0.318 ±0.022	0.219 ±0.010
SPCE-W	0.623 ±0.054	0.720 ±0.069	0.164 ±0.017	0.334 ±0.227	0.474 ±0.233	0.325 ±0.185	0.524 ±0.125	0.405 ±0.057	0.177 ±0.008	0.581 ±0.063	0.124 ±0.055	0.210 ±0.030
SPCE-fixW	0.654 ±0.042	0.810 ±0.027	0.155 ±0.026	0.696 ±0.054	0.521 ±0.203	0.454 ±0.024	0.667 ±0.042	0.602 ±0.068	0.161 ±0.040	0.641 ±0.024	0.230 ±0.045	0.214 ±0.031
SPCE	**0.716** ±0.003	**0.830** ±0.007	**0.294** ±0.011	**0.808** ±0.022	**0.611** ±0.034	**0.473** ±0.015	**0.685** ±0.017	**0.679** ±0.014	**0.227** ±0.009	**0.673** ±0.015	**0.348** ±0.021	**0.278** ±0.043

the parameters in compared methods, as suggested in their articles.

All experiments are conducted using MATLAB on a PC with Windows 10, 4.2-GHz CPU and 32-GB memory.

D. Experimental Results

The average ACC, NMI, and ARI results and the standard deviation on the k-means-based ensemble are shown in Tables III–V, respectively. The results on spectral-based ensemble are shown in Tables VI–VIII, respectively. Bold fond indicates that the difference is statistically significant, i.e., the p-value of t-test is smaller than 0.05. Note that since we aim to compare with other clustering ensemble methods, we do not calculate the p-value of KM, KM-best, SC, and SC-best. Due to their high space complexity, RCE and MEC yield no results on the largest data set Tdt2 because they run out of memory.

The results reveal some interesting points.

1) Many clustering ensemble methods perform better than KM and SC, which indicates the benefit of ensemble methods. Many methods cannot outperform the KM-best and SC-best at most times. It may be because many base results are not so good, and these bad clustering results may deteriorate the performance of ensemble learning. However, the performance of our SPCE is usually close to or even better than the result of KM-best and SC-best. In our formulation, we minimize the Frobenius norm instead of the square of the Frobenius norm of the difference between $\mathbf{S}$ and $\mathbf{S}^{(i)}$. It is equivalent to add the weight on each base clustering, which can reduce the side effect of the bad base clusterings. Moreover,

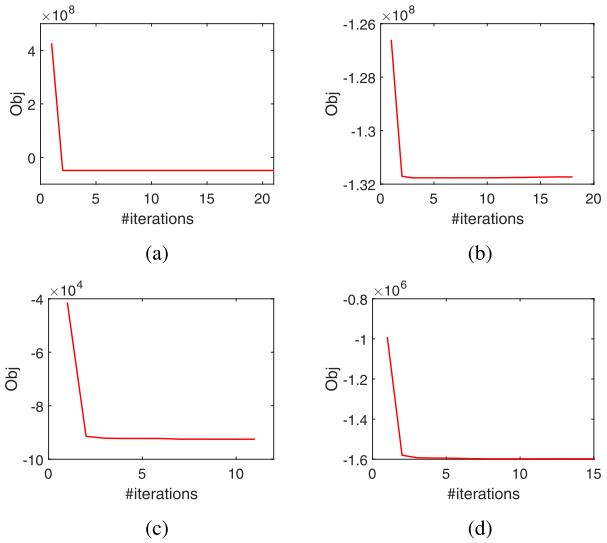


Fig. 2. Convergence curves of our method. (a) AR. (b) Coil20. (c) GLIOMA. (d) Lung.

TABLE VIII
ARI RESULTS ON ALL THE DATA SETS (SPECTRAL CLUSTERING BASED)

Methods	AR	Coil20	GLIOMA	K1b	Lung	Medical	Tr41	Tdt2	Tox	UMIST	WebACE	WarpAR
SC	0.083 ±0.057	0.351 ±0.187	0.070 ±0.039	0.602 ±0.329	0.360 ±0.225	0.134 ±0.106	0.349 ±0.249	0.150 ±0.110	0.090 ±0.066	0.226 ±0.131	0.026 ±0.032	0.032 ±0.023
SC-best	0.134 ±0.006	0.574 ±0.030	0.108 ±0.006	0.885 ±0.008	0.640 ±0.048	0.246 ±0.013	0.575 ±0.037	0.258 ±0.017	0.156 ±0.005	0.249 ±0.016	0.082 ±0.014	0.064 ±0.007
KC	0.025 ±0.007	0.559 ±0.065	0.081 ±0.023	0.748 ±0.078	0.378 ±0.136	0.170 ±0.024	0.475 ±0.069	0.143 ±0.020	0.126 ±0.038	0.276 ±0.019	0.076 ±0.019	0.025 ±0.012
CSPA [3]	0.036 ±0.008	**0.652** ±0.036	0.097 ±0.011	0.268 ±0.004	0.158 ±0.011	0.195 ±0.010	0.353 ±0.019	0.000 ±0.000	0.139 ±0.025	0.326 ±0.022	**0.151** ±0.006	0.048 ±0.013
HGPA [3]	0.142 ±0.005	0.482 ±0.067	0.072 ±0.050	0.276 ±0.048	0.128 ±0.058	0.170 ±0.017	0.201 ±0.045	0.072 ±0.005	0.025 ±0.020	0.334 ±0.052	0.069 ±0.009	0.046 ±0.014
MCLA [3]	0.000 ±0.000	0.014 ±0.044	0.072 ±0.041	0.266 ±0.139	0.176 ±0.211	0.002 ±0.005	0.005 ±0.007	0.000 ±0.000	0.063 ±0.070	0.000 ±0.000	0.000 ±0.000	0.004 ±0.008
NMFC [24]	0.026 ±0.008	0.606 ±0.026	0.090 ±0.021	0.764 ±0.079	0.379 ±0.139	0.194 ±0.020	0.520 ±0.053	0.162 ±0.009	0.137 ±0.018	0.305 ±0.017	0.063 ±0.020	0.024 ±0.013
BCE [28]	0.000 ±0.000	0.431 ±0.232	**0.110** ±0.021	0.792 ±0.086	0.394 ±0.165	0.201 ±0.014	0.469 ±0.175	0.000 ±0.000	0.132 ±0.030	0.184 ±0.159	0.046 ±0.037	0.041 ±0.008
RCE [39]	0.109 ±0.011	**0.646** ±0.023	0.100 ±0.012	0.855 ±0.053	0.544 ±0.163	0.225 ±0.011	0.548 ±0.038	-	0.139 ±0.011	0.333 ±0.013	0.054 ±0.026	0.041 ±0.008
MEC [41]	0.008 ±0.003	0.545 ±0.062	0.101 ±0.014	0.526 ±0.083	0.328 ±0.122	0.186 ±0.017	0.480 ±0.089	-	0.133 ±0.019	0.279 ±0.030	0.032 ±0.021	0.024 ±0.016
LWEA [72]	0.150 ±0.006	0.631 ±0.017	0.102 ±0.009	0.862 ±0.054	0.541 ±0.162	0.231 ±0.017	0.553 ±0.017	0.292 ±0.023	0.136 ±0.001	0.346 ±0.016	0.016 ±0.009	0.037 ±0.007
LWGP [72]	0.142 ±0.005	0.629 ±0.012	0.102 ±0.009	0.853 ±0.061	0.545 ±0.165	0.229 ±0.015	0.552 ±0.014	0.270 ±0.011	0.136 ±0.001	0.338 ±0.014	0.042 ±0.020	0.039 ±0.008
RSEC [61]	0.049 ±0.018	0.291 ±0.077	0.060 ±0.039	0.349 ±0.174	0.312 ±0.085	0.085 ±0.027	0.345 ±0.060	0.135 ±0.028	0.056 ±0.021	0.256 ±0.053	0.044 ±0.039	0.014 ±0.013
DREC [73]	0.146 ±0.005	**0.644** ±0.018	0.102 ±0.009	0.861 ±0.055	0.543 ±0.165	0.226 ±0.008	0.529 ±0.043	0.220 ±0.008	0.136 ±0.001	0.339 ±0.020	0.038 ±0.017	0.041 ±0.006
SPCE-W	0.099 ±0.041	0.462 ±0.084	0.097 ±0.013	0.421 ±0.289	0.500 ±0.239	0.164 ±0.112	0.399 ±0.128	0.066 ±0.032	0.139 ±0.007	0.262 ±0.051	0.010 ±0.012	0.039 ±0.011
SPCE-fixW	0.108 ±0.029	0.590 ±0.059	0.089 ±0.024	0.791 ±0.064	0.551 ±0.227	0.225 ±0.024	0.528 ±0.061	0.250 ±0.076	0.126 ±0.032	0.315 ±0.029	0.014 ±0.018	0.043 ±0.009
SPCE	**0.154** ±0.006	**0.640** ±0.017	**0.115** ±0.010	**0.881** ±0.016	**0.639** ±0.068	**0.249** ±0.023	**0.566** ±0.027	**0.323** ±0.046	**0.152** ±0.009	**0.355** ±0.015	0.041 ±0.014	**0.053** ±0.009

the self-paced learning framework can also reduce the effect of hard (or bad) instances. Note that, SPCE does not need to perform an exhaustive search on the predefined pool of base clusterings. Such results well demonstrate the superiority of our method.

2) On most data sets, our method outperforms other compared methods significantly. Compared with the robust methods RCE, MEC, and RSEC, our method can also usually obtain a better performance. This may be because our method can learn a clearer cluster structure, which is illustrated in Fig. 1. In Fig. 1, we show the input coassociation matrix and the consensus matrices learned from these robust ensemble methods on the K1b data set. We can see that the consensus matrix learned by our SPCE method is cleaner than other robust methods (the second cluster and the third cluster are separated more clearly in our method and so are the fourth cluster and the fifth cluster), which demonstrates the effectiveness of our method.
3) SPCE-fixW performs better than SPCE-W, which means imposing the weights on instances can indeed improve the performance. Compared with SPCE-fixW, SPCE outperforms it on most data sets. It demonstrates the effectiveness of self-paced learning in our framework. Learning from easy instances and involving difficult instances gradually can further improve the performance of the clustering ensemble.

Table IX shows the running time (with 20 base clusterings for ensemble) of the clustering ensemble methods on all data sets. The underlined data means that the corresponding method is slower than ours on that data set. From Table IX, we can find that compared with other connective matrix-based methods, i.e., RCE, MEC, and RSEC, our method is significantly faster than them on most data sets. Note that on the largest data set, Tdt2, RCE, and MEC run out of memory, while our method can still work.

Fig. 2 shows the algorithm convergence curve on AR, Coil20, GLIOMA, and Lung data sets, and the results on other data sets are similar. The example results in Fig. 2 demonstrate that our method converges within a small number of iterations.

E. Parameter Study

Our method contains only one hyperparameter ($0 \leq \theta < 1$), which is needed to set manually. As discussed in Section III-E, θ plays a role as a threshold that controls the sparsity of the consensus matrix $\mathbf{S}$, i.e., the larger θ is, the sparser $\mathbf{S}$ is. We tune θ from $\{0, 0.1, \ldots, 0.9\}$ and show the results in Fig. 3. The results of other data sets are similar. Note that $\theta = 0$ means that we drop the term $\|\mathbf{S}\|_0$ in our formulation, and the results show that our method does not perform well without this term ($\theta = 0$), which demonstrates its necessity. From Fig. 3, we can select θ from $[0.2, 0.6]$, which can often obtain a relatively good performance.

TABLE IX
RUNNING TIME ON ALL THE DATA SETS WITH 20 BASE CLUSTERINGS (S)

Methods	AR	Coil20	GLIOMA	K1b	Lung	Medical	Tr41	Tdt2	Tox	UMIST	WebACE	WarpAR
KC	0.128	0.171	0.002	0.468	0.005	0.036	0.048	55.44	0.004	0.024	0.658	0.004
CSPA [3]	0.403	1.076	0.170	2.490	0.197	0.327	0.502	14.22	0.186	0.249	2.735	0.175
HGPA [3]	2.600	1.206	0.191	0.342	0.196	0.388	0.318	40.91	0.182	1.175	1.230	0.300
MCLA [3]	0.953	0.257	0.180	0.210	0.179	0.219	0.203	3.878	0.185	0.219	0.289	0.185
NMFC [24]	0.218	0.318	0.003	4.132	0.007	0.056	0.317	76.63	0.008	0.037	1.048	0.006
BCE [28]	80.26	6.964	0.092	5.043	0.411	4.373	2.467	983.8	0.358	3.326	15.03	0.438
RCE [39]	79.87	251.1	0.209	804.2	2.154	53.94	97.20	-	1.577	31.73	802.1	0.955
MEC [41]	53.95	207.6	0.089	915.8	0.886	20.85	40.05	-	0.634	13.51	967.4	0.408
LWEA [72]	0.007	0.002	0.001	0.002	0.001	0.001	0.001	0.011	0.001	0.001	0.002	0.001
LWGP [72]	1.911	0.031	0.005	0.014	0.008	0.021	0.014	1.634	0.007	0.024	0.035	0.010
RSEC [61]	11.64	75.82	0.030	412.1	0.317	5.631	10.95	30127	0.232	3.504	411.1	0.130
DREC [73]	8.092	0.809	0.069	1.278	0.123	0.989	0.807	2981	0.101	0.609	2.690	0.028
SPCE	9.495	31.92	0.016	98.55	0.123	4.973	6.806	5423	0.103	3.852	89.82	0.115

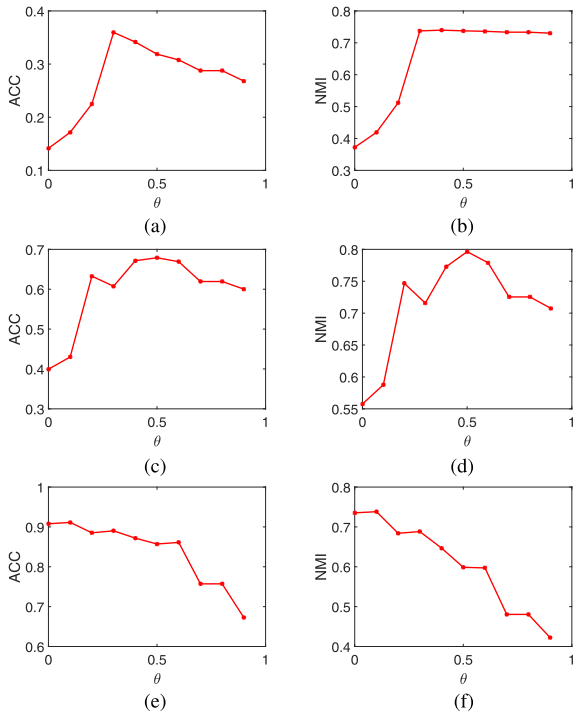


Fig. 3. ACC and NMI with respect to θ. (a) ACC with respect to θ on AR. (b) NMI with respect to θ on AR. (c) ACC with respect to θ on Coil20. (d) NMI with respect to θ on Coil20. (e) ACC with respect to θ on Lung. (f) NMI with respect to θ on Lung.

V. CONCLUSION

In this article, we proposed a novel SPCE method. Different from the conventional clustering ensemble methods, which use all instances in ensemble learning, we gradually involved instances in learning from easy to difficult ones. In the self-paced learning framework, we proposed an effective algorithm to jointly learn the difficulty of instances and the consensus clustering result. We conducted extensive experiments on benchmark data sets, and the experimental results demonstrated that our method not only outperformed the state-of-the-art clustering ensemble methods but also had a closed or even better performance compared with the best base clustering result.

In the future, we will consider the scalable issue and try to reduce the time and space complexity of our method.

REFERENCES

[1] F. Wang, X. Wang, and T. Li, "Generalized cluster aggregation," in *Proc. IJCAI*, 2009, pp. 1279–1284.
[2] A. Topchy, A. K. Jain, and W. Punch, "A mixture model for clustering ensembles," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2004, pp. 379–390.
[3] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 583–617, 2003.
[4] A. Topchy, A. K. Jain, and W. Punch, "Combining multiple weak clusterings," in *Proc. 3rd IEEE Int. Conf. Data Mining*, 2003, pp. 331–338.
[5] Z.-H. Zhou and W. Tang, "Clusterer ensemble," *Knowl.-Based Syst.*, vol. 19, no. 1, pp. 77–83, Mar. 2006.
[6] F.-J. Li, Y.-H. Qian, J.-T. Wang, and J.-Y. Liang, "Multigranulation information fusion: A Dempster-Shafer evidence theory based clustering ensemble method," in *Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC)*, vol. 1, Jul. 2015, pp. 58–63.
[7] H. Liu, M. Shao, S. Li, and Y. Fu, "Infinite ensemble clustering," *Data Mining Knowl. Discovery*, vol. 32, no. 2, pp. 385–416, Mar. 2018.
[8] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, p. 36.
[9] Y. Ren, C. Domeniconi, G. Zhang, and G. Yu, "Weighted-object ensemble clustering: Methods and analysis," *Knowl. Inf. Syst.*, vol. 51, no. 2, pp. 661–689, May 2017.
[10] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. ICML*, 2009, pp. 41–48.
[11] T. G. Dietterich *et al.*, "Ensemble learning," *The Handbook of Brain Theory and Neural Networks*, vol. 2. Cambridge, MA, USA: MIT Press, 2002, pp. 110–125.
[12] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, Oct. 1990.
[13] H. Yu, J. Wang, Y. Bai, W. Yang, and G.-S. Xia, "Analysis of large-scale UAV images using a multi-scale hierarchical representation," *Geo-Spatial Inf. Sci.*, vol. 21, no. 1, pp. 33–44, Jan. 2018.
[14] X. Deng, J. Chen, H. Li, P. Han, and W. Yang, "Log-cumulants of the finite mixture model and their application to statistical analysis of fully polarimetric UAVSAR data," *Geo-Spatial Inf. Sci.*, vol. 21, no. 1, pp. 45–55, Jan. 2018.
[15] Z.-H. Zhou, Y. Jiang, Y.-B. Yang, and S.-F. Chen, "Lung cancer cell identification based on artificial neural network ensembles," *Artif. Intell. Med.*, vol. 24, no. 1, pp. 25–36, Jan. 2002.
[16] P. Zhou, Y.-D. Shen, L. Du, F. Ye, and X. Li, "Incremental multi-view spectral clustering," *Knowl.-Based Syst.*, vol. 174, pp. 73–86, Jun. 2019.
[17] P. Zhou, Y.-D. Shen, L. Du, and F. Ye, "Incremental multi-view support vector machine," in *Proc. SIAM Int. Conf. Data Mining*. Philadelphia, PA, USA: SIAM, 2019, pp. 1–9.
[18] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "Marginalized multiview ensemble clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 600–611, Feb. 2020.

[19] S. Wang *et al.*, "Multi-view clustering via late fusion alignment maximization," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3778–3784.

[20] S. Wang *et al.*, "Efficient multiple kernel k-means clustering with late fusion," *IEEE Access*, vol. 7, pp. 61109–61120, 2019.

[21] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[22] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, pp. 1189–1232, Oct. 2001.

[23] T. Li, C. Ding, and M. I. Jordan, "Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization," in *Proc. 7th IEEE Int. Conf. Data Mining (ICDM)*, Oct. 2007, pp. 577–582.

[24] T. Li and C. Ding, "Weighted consensus clustering," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2008, pp. 798–809.

[25] H. Liu, T. Liu, J. Wu, D. Tao, and Y. Fu, "Spectral ensemble clustering," in *Proc. SIGKDD*, 2015, pp. 715–724.

[26] Z. Tao, H. Liu, and Y. Fu, "Simultaneous clustering and ensemble," in *Proc. AAAI*, 2017, pp. 1546–1552.

[27] D. Huang, C.-D. Wang, J. Wu, J.-H. Lai, and C. K. Kwoh, "Ultra-scalable spectral clustering and ensemble clustering," *IEEE Trans. Knowl. Data Eng.*, early access, Mar. 6, 2019, doi: 10.1109/TKDE.2019.2903410.

[28] H. Wang, H. Shan, and A. Banerjee, "Bayesian cluster ensembles," in *Proc. SDM*, 2009, pp. 211–222.

[29] D. Huang, J. Lai, and C.-D. Wang, "Ensemble clustering using factor graph," *Pattern Recognit.*, vol. 50, pp. 131–142, Feb. 2016.

[30] S.-O. Abbasi, S. Nejatian, H. Parvin, V. Rezaie, and K. Bagherifard, "Clustering ensemble selection considering quality and diversity," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 1311–1340, 2019.

[31] A. Bagherinia, B. Minaei-Bidgoli, M. Hossinzadeh, and H. Parvin, "Elite fuzzy clustering ensemble based on clustering diversity and quality measures," *Appl. Intell.*, vol. 49, no. 5, pp. 1724–1747, 2019.

[32] J. Azimi and X. Fern, "Adaptive cluster ensemble selection," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 992–997.

[33] Y. Hong, S. Kwong, H. Wang, and Q. Ren, "Resampling-based selective clustering ensembles," *Pattern Recognit. Lett.*, vol. 30, no. 3, pp. 298–305, Feb. 2009.

[34] H. Parvin and B. Minaei-Bidgoli, "A clustering ensemble framework based on elite selection of weighted clusters," *Adv. Data Anal. Classification*, vol. 7, no. 2, pp. 181–208, Jun. 2013.

[35] H. Parvin and B. Minaei-Bidgoli, "A clustering ensemble framework based on selection of fuzzy weighted clusters in a locally adaptive clustering algorithm," *Pattern Anal. Appl.*, vol. 18, no. 1, pp. 87–112, Feb. 2015.

[36] Z. Yu *et al.*, "Hybrid clustering solution selection strategy," *Pattern Recognit.*, vol. 47, no. 10, pp. 3362–3375, Oct. 2014.

[37] X. Zhao, J. Liang, and C. Dang, "Clustering ensemble selection for categorical data based on internal validity indices," *Pattern Recognit.*, vol. 69, pp. 150–168, Sep. 2017.

[38] Y. Shi *et al.*, "Transfer clustering ensemble selection," *IEEE Trans. Cybern.*, early access, Dec. 25, 2018, doi: 10.1109/TCYB.2018.2885585.

[39] P. Zhou, L. Du, H. Wang, L. Shi, and Y. Shen, "Learning a robust consensus matrix for clustering ensemble via Kullback-Leibler divergence minimization," in *Proc. IJCAI*, 2015, pp. 4112–4118.

[40] Z. Tao, H. Liu, S. Li, and Y. Fu, "Robust spectral ensemble clustering," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2016, pp. 367–376.

[41] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "From ensemble clustering to multi-view clustering," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2843–2849.

[42] D. Huang, C.-D. Wang, H. Peng, J. Lai, and C.-K. Kwoh, "Enhanced ensemble clustering via fast propagation of cluster-wise similarities," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Nov. 5, 2018, doi: 10.1109/TSMC.2018.2876202.

[43] E. A. Ni and C. X. Ling, "Supervised learning with minimal effort," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Hyderabad, India: Springer, 2010, pp. 476–487.

[44] S. Basu and J. Christensen, "Teaching classification boundaries to humans," in *Proc. AAAI*, 2013, pp. 109–115.

[45] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proc. NIPS*, 2010, pp. 1189–1197.

[46] L. Jiang, D. Meng, S. Yu, Z. Lan, S. Shan, and A. G. Hauptmann, "Self-paced learning with diversity," in *Proc. NIPS*, 2014, pp. 2078–2086.

[47] D. Zhang, D. Meng, and J. Han, "Co-saliency detection via a self-paced multiple-instance learning framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 865–878, May 2017.

[48] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann, "Self-paced curriculum learning," in *Proc. AAAI*, 2015, pp. 2694–2700.

[49] Q. Zhao, D. Meng, L. Jiang, Q. Xie, Z. Xu, and A. G. Hauptmann, "Self-paced learning for matrix factorization," in *Proc. AAAI*, 2015, pp. 3196–3202.

[50] D. Meng, Q. Zhao, and L. Jiang, "A theoretical understanding of self-paced learning," *Inf. Sci.*, vol. 414, pp. 319–328, Nov. 2017.

[51] C. Li, J. Yan, F. Wei, W. Dong, Q. Liu, and H. Zha, "Self-paced multi-task learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2175–2181.

[52] Y. Ren, X. Que, D. Yao, and Z. Xu, "Self-paced multi-task clustering," *Neurocomputing*, vol. 350, pp. 212–220, Jul. 2019.

[53] Y. Ren, P. Zhao, Y. Sheng, D. Yao, and Z. Xu, "Robust softmax regression for multi-class classification with self-paced learning," in *Proc. 26th Int. Joint Conf. Artif. Intell.* Stanford, CA, USA: AAAI Press, Aug. 2017, pp. 2641–2647.

[54] Z. Kang, X. Lu, J. Yi, and Z. Xu, "Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1881–1887.

[55] F. Nie, L. Tian, and X. Li, "Multiview clustering via adaptively weighted procrustes," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2022–2030.

[56] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2014, pp. 977–986.

[57] K. Fan, "On a theorem of Weyl concerning eigenvalues of linear transformations. II," *Proc. Nat. Acad. Sci. USA*, vol. 36, no. 1, pp. 31–35, 1949.

[58] P. Zhou, L. Du, L. Shi, H. Wang, and Y.-D. Shen, "Recovery of corrupted multiple kernels for clustering," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 4105–4111.

[59] X. Liu *et al.*, "Optimal neighborhood kernel clustering with multiple kernels," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2266–2272.

[60] Y. Wang, X. Liu, Y. Dou, and R. Li, "Multiple kernel clustering framework with improved kernels," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2999–3005.

[61] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "Robust spectral ensemble clustering via rank minimization," *ACM Trans. Knowl. Discovery from Data*, vol. 13, no. 1, pp. 1–25, Jan. 2019.

[62] D. Huang, J.-H. Lai, and C.-D. Wang, "Robust ensemble clustering using probability trajectories," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 5, pp. 1312–1326, May 2016.

[63] X. Liu *et al.*, "Late fusion incomplete multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2410–2423, Oct. 2019.

[64] H. Wang, F. Nie, and H. Huang, "Globally and locally consistent unsupervised projection," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1328–1333.

[65] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[66] J. Li *et al.*, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, p. 94, 2018.

[67] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," *Mach. Learn.*, vol. 55, no. 3, pp. 311–331, Jun. 2004.

[68] Z.-Q. Hong and J.-Y. Yang, "Optimal discriminant plane for a small number of samples and design method of classifier on the plane," *Pattern Recognit.*, vol. 24, no. 4, pp. 317–324, Jan. 1991.

[69] D. Cai, X. He, W. V. Zhang, and J. Han, "Regularized locality preserving indexing via spectral regression," in *Proc. 16th ACM Conf. Inf. Knowl. Manage. (CIKM)*, 2007, pp. 741–750.

[70] H. Wechsler, J. P. Phillips, V. Bruce, F. F. Soulié, and T. S. Huang, *Face Recognition: From Theory to Applications*, vol. 163. Springer, 2012.

[71] L. Du, X. Li, and Y.-D. Shen, "Cluster ensembles via weighted graph regularized nonnegative matrix factorization," in *Proc. Int. Conf. Adv. Data Mining Appl.* Beijing, China: Springer, 2011, pp. 215–228.

[72] D. Huang, C.-D. Wang, and J.-H. Lai, "Locally weighted ensemble clustering," *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1460–1473, May 2018.

[73] J. Zhou, H. Zheng, and L. Pan, "Ensemble clustering based on dense representation," *Neurocomputing*, vol. 357, pp. 66–76, Sep. 2019.

Peng Zhou received the B.E. degree in computer science and technology from the University of Science and Technology of China, Hefei, China, in 2011, and the Ph.D. degree in computer science from the Institute of Software, University of Chinese Academy of Sciences, Beijing, China, in 2017.

He is currently a Lecturer with the School of Computer Science and Technology, Anhui University, Hefei. His research interests include machine learning, data mining, and artificial intelligence.

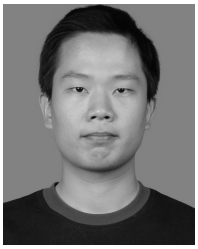
Liang Du (Member, IEEE) received the B.E. degree in software engineering from Wuhan University, Wuhan, China, in 2007, and the Ph.D. degree in computer science from the Institute of Software, University of Chinese Academy of Sciences, Beijing, China, in 2013.

He was a Software Engineer with Alibaba Group, Hangzhou, China, from July 2013 to July 2014. He is currently a Lecturer with Shanxi University, Taiyuan, China. His research interests include machine learning, data mining, and big data analysis.

Xinwang Liu received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China.

He is currently a Professor with the School of Computer, NUDT. He has published more than 60 peer-reviewed articles, including those in highly regarded journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (T-PAMI), the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (T-KDE), the IEEE TRANSACTIONS ON IMAGE PROCESSING (T-IP), the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (T-NNLS), the IEEE TRANSACTIONS ON MULTIMEDIA (T-MM), the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (T-IFS), NeurIPS, the International Conference on Computer Vision (ICCV), the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), the AAAI Conference on Artificial Intelligence (AAAI), and the International Joint Conference on Artificial Intelligence (IJCAI). His current research interests include kernel learning and unsupervised feature learning.

Yi-Dong Shen was a Professor with Chongqing University, Chongqing, China. He is currently a Professor of computer science with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China. His main research interests include knowledge representation and reasoning, semantic web, and data mining.



Mingyu Fan received the B.Sc. degree in information and computing science from the Minzu University of China, Beijing, China, in 2006, and the Ph.D. degree in applied mathematics from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, in 2011.

He is currently a Professor with Wenzhou University, Wenzhou, China. His current research interests include data mining and pattern recognition algorithms and the applications of them on solving industrial problems.

Xuejun Li received the Ph.D. degree from Anhui University, Hefei, China, in 2008.

He is currently a Professor with the School of Computer Science and Technology, Anhui University. His major research interests include workflow systems, cloud computing, and intelligent software.