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
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 $M$ is denoted as $M_{ij}$, and the $i$th element of a vector $v$ is denoted as $v_i$. Given a matrix $M \in \mathbb{R}^{n \times d}$, we use $\|M\|_F = (\sum_{i=1}^{n} \sum_{j=1}^{d} M_{ij}^2)^{1/2}$ to denote its Frobenius norm. We use $\|M\|_0$ to denote its $\ell_0$-norm, which is the number of nonzero elements in $M$. Since $\ell_0$-norm is nonconvex and discontinuous, $\ell_1$-norm is often used as an approximation of $\ell_0$-norm. $\ell_1$-norm of $M$ is defined as $\|M\|_1 = \sum_{i=1}^{n} \sum_{j=1}^{d} |M_{ij}|$.

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 assembled 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 $X = \{x_1, x_2, \ldots, x_n\}$ be a data set of $n$ data points. Suppose that we are given a set of $m$ clusterings $C = \{C_1, C_2, \ldots, C_m\}$ of the data in $X$, each clustering $C_i$ consisting of a set of clusters $\{\pi_1^i, \pi_2^i, \ldots, \pi_k^i\}$, where $k$ is the number of clusters in $C_i$ and $X = \bigcup_{i=1}^{m} \pi_i^i$. 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 $C_1, \ldots, C_m$.

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 ensemble 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 $C$, following [24], [39]–[42], we can construct the connective matrix $S^{(i)} \in \mathbb{R}^{n \times n}$ for partition $C_i$ as

$$S^{(i)}_{pq} = \begin{cases} 1, & \text{if } x_p \text{ and } x_q \text{ belong to the same cluster,} \\ 0, & \text{otherwise.} \end{cases}$$
The target of our clustering ensemble method is to learn a consensus matrix $S$ from $S^{(1)}, S^{(2)}, \ldots, S^{(m)}$ and then obtain the final clustering result from the consensus matrix $S$. 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 $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ containing $n$ instances, where $x_i \in \mathbb{R}^d$ is the $d$-dimensional feature vector of the $i$th instance and $y_i$ is its label, $L(y_i, g(x_i, \theta))$ is denoted as the loss function that indicates the cost between the ground truth $y_i$ and the estimated label $g(x_i, \theta)$, where $\theta$ 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:

$$\min_{\theta, w} \sum_{i=1}^{n} (w_i L(y_i, g(x_i, \theta)) + f(w_i, \lambda))$$

where $\lambda$ is an age parameter for controlling the learning pace, $w_i$ is the weight of the $i$th instance, and $f(w_i, \lambda)$ is the self-paced regularizer. When fixing $\theta$, supposing that $l_i = L(y_i, g(x_i, \theta))$ and $w_i^\ast(\lambda, l_i)$ is the optimum weight of the $i$th instance, which is relative with $\lambda$ and $l_i$, $f(w_i, \lambda)$ should satisfy that $w_i^\ast(\lambda, l_i)$ is monotonically decreasing with $l_i$ and increasing with $\lambda$, as suggested in [48]–[50]. Since $w_i^\ast(\lambda, l_i)$ 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. $w_i^\ast(\lambda, l_i)$ is monotonically increasing with $\lambda$ indicates that with the learning process ($\lambda$ grows), more and more instances are used for learning.

Therefore, the process of self-paced learning is to optimize (1) via alternating minimization. Fixing $\theta$ and solving $w$ are to learn the weight of each instance; fixing $w$ and solving $\theta$ 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.

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 $S^{(1)}, S^{(2)}, \ldots, S^{(m)}$, the samples are the data pairs appearing in $S^{(i)}$. 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 $x_p$ and $x_q$, if all $m$ connective matrices agree that $x_p$ and $x_q$ 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 $S$ as

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

It is easy to verify that for any $1 \leq p, q \leq n$, we have $0 \leq \hat{S}_{pq} \leq 1$. $\hat{S}_{pq} = 1$ indicates that all connective matrices agree that $x_p$ and $x_q$ 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 $\hat{S}$, which are either 1 or 0. To make the learned consensus matrix $S$ to preserve such certain information, we directly set $S$ 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 $S$ can be learned by solving a matrix completion problem, i.e., we fill the missing values in $S$ by self-paced consensus learning.

B. Self-Paced Consensus Learning

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

---

Table I

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>number of instances</td>
</tr>
<tr>
<td>$m$</td>
<td>number of base clusterings</td>
</tr>
<tr>
<td>$c$</td>
<td>number of clusters</td>
</tr>
<tr>
<td>$S^{(i)} \in \mathbb{R}^{n \times n}$</td>
<td>the connective matrix of the $i$-th base clustering</td>
</tr>
<tr>
<td>$S \in \mathbb{R}^{n \times n}$</td>
<td>the co-association matrix</td>
</tr>
<tr>
<td>$S \in \mathbb{R}^{n \times n}$</td>
<td>the learned consensus matrix</td>
</tr>
<tr>
<td>$\Omega \in {0, 1}^{n \times n}$</td>
<td>the indicator matrix, $\Omega_{ij} = 1$ if $\hat{S}<em>{ij} = 0$ or 1, $\Omega</em>{ij} = 0$ otherwise</td>
</tr>
<tr>
<td>$L \in \mathbb{R}^{n \times n}$</td>
<td>the Laplacian matrix of $S$</td>
</tr>
<tr>
<td>$W \in \mathbb{R}^{n \times n}$</td>
<td>the weight matrix to show the difficulty of each pair</td>
</tr>
</tbody>
</table>
A natural idea is to minimize \( \sum_{i=1}^{m} \| S - S^{(i)} \|_F^2 \). 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 \( \sum_{i=1}^{m} \alpha_i \| S - S^{(i)} \|_F^2 \), where \( \alpha_i \) is the weight of the \( i \)th 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 \( \alpha_i = (1/(\| S - S^{(i)} \|_F)) \), which means the closer \( S^{(i)} \) to the consensus matrix, the larger the weight of the \( i \)th clustering result is. Thus, we obtain the following objective function:

\[
\min_S \sum_{i=1}^{m} \| S - S^{(i)} \|_F^2
\]

s.t. \( S \odot \Omega = \hat{S} \odot \Omega \), \( 0 \leq S_{pq} \leq 1 \) \( \forall p, q \)  \( (3) \)

where \( \Omega \in \mathbb{R}^{n \times n} \) is an indicator matrix, whose element \( \Omega_{pq} = 1 \) if \( \hat{S}_{pq} = 0 \) or 1 and \( \Omega_{pq} = 0 \) otherwise. \( \odot \) is the Hadamard product, which means the elementwise product of two matrices. The constraint is to make sure that the consensus matrix \( S \) 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 \( x_p \) and \( x_q \), if most \( S^{(i)} \) 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 \( (x_p, x_q) \) is. To this end, we introduce a weight matrix \( W \) whose element \( 0 \leq W_{pq} \leq 1 \) indicates the weight of the pair \( (x_p, x_q) \). 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 \( f(W, \lambda) = -\lambda \| W \|_1 \) and obtain the following objective:

\[
\min_{S, W} \sum_{i=1}^{m} \| (S - S^{(i)}) \odot W \|_F - \lambda \| W \|_1
\]

s.t. \( S \odot \Omega = \hat{S} \odot \Omega \), \( 0 \leq S_{pq} \leq 1 \) \( \forall p, q \) \( \forall \lambda \geq 0 \) \( \forall \alpha_i \geq 0 \) \( (4) \)

where \( \lambda \) is the age parameter and becomes increasingly larger in the process of optimization.

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

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

\[
\min_{S, W} \sum_{i=1}^{m} \| (S - S^{(i)}) \odot W \|_F - \lambda \| W \|_1
\]

s.t. \( S \odot \Omega = \hat{S} \odot \Omega \), \( 0 \leq S_{pq} \leq 1 \) \( \forall p, q \), \( \text{rank}(L) = n - c \), \( 0 \leq W_{pq} \leq 1 \) \( \forall p, q \) \( (5) \)

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

\[
\min_{S, W} \sum_{i=1}^{m} \| (S - S^{(i)}) \odot W \|_F - \lambda \| W \|_1 + \gamma \| S \|_0
\]

s.t. \( S \odot \Omega = \hat{S} \odot \Omega \), \( 0 \leq S_{pq} \leq 1 \) \( \forall p, q \), \( \text{rank}(L) = n - c \), \( 0 \leq W_{pq} \leq 1 \) \( \forall p, q \) \( (6) \)

where \( \gamma \) is a balancing hyperparameter that can adjust the sparsity of \( S \). Note that we use \( \ell_0 \)-norm here instead of \( \ell_1 \)-norm or any other convex or nonconvex approximation, which can make \( S \) 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 \( \alpha \), where \( 0 \leq \alpha_i \leq 1 \) and \( \sum_{i=1}^{m} \alpha_i = 1 \), we have the following theorem.

\[ \text{Theorem 2: min}_{S, W} \sum_{i=1}^{m} \| (S - S^{(i)}) \odot W \|_F' \text{ is equivalent to min}_{S, W, \alpha} \sum_{i=1}^{m} (1/\alpha_i) \| (S - S^{(i)}) \odot W \|_F^2'. \]

Proof: Let \( S^* \), \( W^* \) and \( \alpha^* \) denote the optima of \( \text{min}_{S, W} \sum_{i=1}^{m} \| (S - S^{(i)}) \odot W \|_F \). To prove that \( S^* \) and \( W^* \) are also the optima of \( \text{min}_{S, W} \sum_{i=1}^{m} \| (S - S^{(i)}) \odot W \|_F \), we need to prove that given any \( S \) and \( W \), we have

\[
\sum_{i=1}^{m} \| (S^{(i)} - S) \odot W \|_F \leq \sum_{i=1}^{m} \| (S^{(i)} - S') \odot W' \|_F.
\]

Consider that

\[
\left( \sum_{i=1}^{m} \| (S^{(i)} - S) \odot W \|_F \right)^2
\]

\[
\leq \left( \sum_{i=1}^{m} \frac{1}{\alpha_i} \| (S^{(i)} - S) \odot W \|_F \right)^2 \left( \sum_{i=1}^{m} \alpha_i \right)
\]

\[
= \sum_{i=1}^{m} \frac{1}{\alpha_i} \| (S^{(i)} - S) \odot W \|_F^2
\]

\[
\leq \sum_{i=1}^{m} \frac{1}{\alpha_i} \| (S^{(i)} - S') \odot W' \|_F^2.
\]
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 $S^*, W^*$ and $\alpha^*$ are the optima of \( \min_{S,W,\alpha} \sum_{i=1}^{m} (1/\alpha_i) \| (S - S(i)) \odot W \|_F^2 \), given any $\tilde{\alpha}_i$, $\tilde{S}$ and $W$, we have $\sum_{i=1}^{m} (1/\alpha_i) \| (S^* - S(i)) \odot W^* \|_F^2 \leq \sum_{i=1}^{m} (1/\alpha_i) \| (\tilde{S} - S(i)) \odot \tilde{W} \|_F^2$.

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

$$\left( \sum_{i=1}^{m} (\| (S^* - S(i)) \odot W^* \|_F)^2 \right) \leq \left( \sum_{i=1}^{m} \frac{\sum_{i=1}^{m} (\| (S - S(i)) \odot W \|_F)^2}{\| (S^* - S(i)) \odot W^* \|_F} \right) \leq \left( \sum_{i=1}^{m} (\| (S - S(i)) \odot \tilde{W} \|_F)^2 \right).$$

which leads to that

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

This completes the proof.

According to Theorem 2, we relax (6) as

$$\min_{S,W,\alpha} \sum_{i=1}^{m} \frac{1}{\alpha_i} \| (S - S(i)) \odot W \|_F^2 - \lambda \| W \|_1 + \gamma \| S \|_0$$

s.t. $S \odot \Omega = \hat{S} \odot \Omega$, $0 \leq S_{pq} \leq 1 \forall p,q$,

rank($L$) = $n-c$,

$0 \leq W_{pq} \leq 1 \forall p,q$,

$0 \leq \alpha_i \leq 1$, $\sum_{i=1}^{m} \alpha_i = 1$. \hfill (9)

Then, we handle the rank function. According to [56], since we wish the rank of $L$ is $n-c$, i.e., the $c$ smallest eigenvalues of $L$ are 0s, we try to minimize $\sum_{i=c+1}^{n} \sigma_i(L)$, where $\sigma_i(L)$ denotes the $i$th smallest eigenvalue of $L$. According to Ky Fan’s theorem [57], we have $\sum_{i=c+1}^{n} \sigma_i(L) = \min_{Y \in \mathbb{R}^{n \times c}} \| Y^T L Y \|_F$.

Thus, by introducing orthogonal matrix $Y \in \mathbb{R}^{n \times c}$, we can reformulate (9) as follows:

$$\min_{S,W,\alpha,Y} \sum_{i=1}^{m} \frac{1}{\alpha_i} \| (S - S(i)) \odot W \|_F^2 - \lambda \| W \|_1 + \gamma \| S \|_0 + \rho \| Y^T L Y \|_F$$

s.t. $S \odot \Omega = \hat{S} \odot \Omega$, $0 \leq S_{pq} \leq 1 \forall p,q$,

$0 \leq W_{pq} \leq 1 \forall p,q$,

$0 \leq \alpha_i \leq 1$, $\sum_{i=1}^{m} \alpha_i = 1$,

$Y^T Y = I$. \hfill (10)

where $\rho$ is a large enough parameter to make sure that the rank of $L$ is $n-c$. Note that different from some conventional ensemble methods that linearly combine all base connective matrices, our $S$ 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 $W$: We remove the terms that are relative to $W$ and obtain

$$\min_{W} \sum_{i=1}^{m} \frac{1}{\alpha_i} \| A(i) \odot W \|_F^2 - \lambda \| W \|_1$$

s.t. $0 \leq W_{pq} \leq 1 \forall p,q$ \hfill (11)

where $A(i) = S - S(i)$.

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

$$\min_{W_{pq}} W_{pq}^2 - \lambda W_{pq}$$

s.t. $0 \leq W_{pq} \leq 1$ \hfill (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 $B_{pq} \geq 0$, $W_{pq} \geq 0$. If $(\lambda/2B_{pq}) > 1$, i.e., $B_{pq} W_{pq} - \lambda W_{pq}$ decreases monotonically in the range [0, 1], so the optima is 1. To sum up, we optimize $W_{pq}$ as

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

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

2) Optimize $S$: When other variables are fixed, we rewrite (10) as

$$\min_{S} \sum_{i=1}^{m} \frac{1}{\alpha_i} \| (S - S(i)) \odot W \|_F^2 + \gamma \| S \|_0$$

$$+ \rho \| \| y_p - y_q \|_2^2 S_{pq}$$

s.t. $S \odot \Omega = \hat{S} \odot \Omega$, $0 \leq S_{pq} \leq 1 \forall p,q$ \hfill (14)

where $y_p$ and $y_q$ are the $p$th and $q$th row vectors in $Y$, respectively.

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

$$\min_{S} \sum_{i=1}^{m} \frac{1}{\alpha_i} (S_{pq} - S_{pq}^2) W_{pq}^2 + \gamma g(S_{pq}) + \rho \| y_p - y_q \|_2^2 S_{pq}$$

s.t. $0 \leq S_{pq} \leq 1$. \hfill (15)
Equation (15) can be simplified further as
\[
\min_{S_{pq}} \frac{(S_{pq} - C_{pq})^2 + \tau g(S_{pq})}{\alpha_i} \\
\text{s.t. } 0 \leq S_{pq} \leq 1
\]  
(16)
where
\[
C_{pq} = \frac{\sum_{i=1}^{m} S_{pq}^{(i)}}{\alpha_i} - \frac{\rho \| y_p - y_q \|^2}{2 W_{pq}}
\]
and \(\tau = (\gamma / \sum_{i=1}^{m} (W_{pq}^2 / (\alpha_i)))\). Equation (16) has a closed-form solution
\[
S_{pq} = \begin{cases} 
1, & \text{if } C_{pq} \geq 1 \\
C_{pq}, & \text{if } \sqrt{\tau} \leq C_{pq} < 1 \\
0, & \text{if } C_{pq} < \sqrt{\tau}.
\end{cases}
\]  
(17)

3) Optimize \(Y\): When optimizing \(Y\), we have
\[
\min_Y \text{tr}(Y^T L Y) \\
\text{s.t. } Y^T Y = I.
\]  
(18)
According to Ky Fan’s theorem [57], the solution of \(Y\) is formed by the \(c\) eigenvectors of \(L\) corresponding to the \(c\) smallest eigenvalues.

4) Optimize \(\alpha\): Let \(d_i\) denote \(\| (S - S^{(i)}) \odot W \|_F^2\), and we have
\[
\min_{\alpha} \sum_{i=1}^{m} d_i/\alpha_i \\
\text{s.t. } 0 \leq \alpha_i \leq 1, \sum_{i=1}^{m} \alpha_i = 1.
\]  
(19)
According to the Cauchy–Schwarz inequality, we have
\[
\sum_{i=1}^{m} d_i/\alpha_i \geq \left( \sum_{i=1}^{m} \sqrt{d_i} \right)^2.
\]  
(20)
The equality in (20) holds when \(\alpha_i \propto \sqrt{d_i}\). Thus, the closed-form solution of (19) is
\[
\alpha_i = \frac{\sqrt{d_i}}{\sum_{j=1}^{m} \sqrt{d_j}}.
\]  
(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 \(S = \sum_{i=1}^{m} (1/m) S^{(i)}\) and construct \(L\) from \(S\). Then, we initialize \(Y\) by solving (18). We set \(\rho = 1\) at first and adjust it automatically by observing the rank of \(L\). We initialize \(\alpha_i = (1/m)\).

We initialize \(W\) by (13). However, in (13), we need to decide \(\lambda\) first. In the initialization, we have set \(S\) as mean of \(S^{(i)}\) and \(\alpha_i = (1/m)\), and then, we take a closer look at \(B_{pq}\). Suppose that for the pair \((x_p, x_q)\), there are \(k\) clustering results agrees that they belong to a cluster and the other \(m-k\) results agrees that they belong to different clusters. Then, we have
\[
B_{pq} = \sum_{i=1}^{m} \frac{(S_{pq} - S_{pq}^{(i)})^2}{\alpha_i} = \left( \frac{k}{m} - 1 \right)^2 k + \left( \frac{k}{m} \right)^2 (m-k) m.
\]  
(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 \(\lambda = 2 r (r - 1)^2 r + r^2 (1 - r)) m^2\) \(23\)
which means for \(x_p\) and \(x_q\), 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 \(\lambda\) by decreasing \(r\) from 0.9 to 0.5.

Then, we discuss how to choose the hyperparameter \(\gamma\). As we know, \(\gamma\) controls the sparsity of \(S\). From (17), we find that if \(C_{pq} < \sqrt{\tau}\), we have \(S_{pq} = 0\) and \(\gamma\) is proportional to \(\tau\); thus, \(\gamma\) plays a role as a threshold. More specifically, \(S_{pq} = 0\) when
\[
C_{pq} < \sqrt{\tau} = \sqrt{\frac{\gamma}{\sum_{i=1}^{m} W_{pq}^2 / \alpha_i}} \approx \sqrt{\frac{\gamma}{\sum_{i=1}^{m} \frac{1}{\alpha_i}}}.
\]  
(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
\[
C_{pq} < \sqrt{\frac{\gamma}{\sum_{i=1}^{m} \frac{1}{\alpha_i}}} \leq \sqrt{\frac{\gamma \sum_{i=1}^{m} \alpha_i}{\sum_{i=1}^{m} \frac{1}{\alpha_i}}} = \sqrt{\frac{\gamma}{m}}.
\]  
(25)
Denote \(\theta = (\sqrt{\gamma}/m)\). \(\theta\) can be viewed as a threshold, i.e., \(S_{pq}\) is nonzero when \(C_{pq} > \theta\). Therefore, we can easily give a threshold \(\theta\) and compute \(\gamma\) by \(\gamma = m^2 \theta^2\) instead of directly setting \(\gamma\). For example, if we wish to keep \(S_{pq}\) nonzero when \(S_{pq} > \theta\), 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 \(W\) automatically in the process of the ensemble. From (13), we observe that \(W\) is proportional to \(\lambda\), i.e., when time goes on (\(\lambda\) increases), \(W\) will also increases until it reaches 1. Therefore, this property can be well characterized by our method.

\section*{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 \(S\), \(Y\), and \(\alpha\). 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.

\textbf{Algorithm 1} SPCE

\textbf{Input:} \(m\) connective matrices \(S^{(1)}, \ldots, S^{(m)}\), number \(c\) of clusters, threshold \(\theta\).

\textbf{Output:} Final clusters.

1: Construct \(\hat{S}\) by Eq.(2) and construct \(\Omega\) from \(\hat{S}\).

2: Initialize the parameters as introduced in Section III-E.

3: \textbf{for} \(r = 0.9, 0.8, \ldots, 0.5\) \textbf{do}

4: Compute \(\rho\) by Eq.(23).

5: Compute \(W\) by Eq.(13).

6: \textbf{while} not converge \textbf{do}

7: Compute \(S\) by Eq.(17).

8: Compute \(Y\) by solving Eq.(18).

9: Compute \(\alpha\) by Eq.(21).

10: \textbf{if} The rank of \(L\) is larger than \(n - c\) \textbf{then}

11: \hspace{1em} \(\rho = 2\rho\).

12: \textbf{else if} The rank of \(L\) is smaller than \(n - c\) \textbf{then}

13: \hspace{1em} \(\rho = \rho/2\).

14: \textbf{else}

15: Break.

16: \textbf{end if}

17: \textbf{end while}

18: \textbf{end for}

19: Obtain the final clusters from the \(c\) connective component in \(S\).

\section*{G. Time and Space Complexity}
Since we need to save and handle \(m\) connective matrices \(S^{(1)}, \ldots, S^{(m)}\), the space complexity is \(O(mn^2)\).

When computing \(W\), we need to compute \(A^{(i)} (i = 1, \ldots, m)\); thus, the time complexity is \(O(mn^2)\). Computing \(S\) costs \(O(n^2c + n^3m)\) since we need to compute \(C\) first. Computing \(Y\) involves an eigenvector decomposition that costs \(O(n^3c)\) time. When computing \(\alpha\), we need to compute \(d\) in \(O(n^2m)\) time. Therefore, the whole time complexity is \(O((n^2m + n^3c)t)\), where \(t\) is the number of iterations.

\begin{table}[h]
\centering
\caption{Description of the Data Sets}
\begin{tabular}{|c|c|c|c|}
\hline
Data Set & \#instances & \#features & \#classes \\
\hline
AR & 840 & 768 & 120 \\
Coil20 & 1440 & 1024 & 20 \\
GILIOMA & 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 \\
\hline
\end{tabular}
\end{table}

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.

\section*{IV. EXPERIMENTS}

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

\subsection*{A. Data Sets}
We use totally 12 data sets to evaluate the effectiveness of our proposed SPCE, including AR [64], Coil20 [65], GILIOMA [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.

\subsection*{B. Compared Methods}
We compare our SPCE with the following algorithms.

1) \textit{KM/SC}: These are the average of all the base k-means and spectral clustering results, respectively.

2) \textit{KM-best/SC-best}: These are the best result of all the base k-means and spectral clustering results, respectively.

3) \textit{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) \textit{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) \textit{Hypergraph Partitioning Algorithm (HGPA)} [3]: It approximates the maximum mutual information objective with a constrained minimum cut objective.

6) \textit{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)

<table>
<thead>
<tr>
<th>Methods</th>
<th>AR</th>
<th>Coil20</th>
<th>GLOMA</th>
<th>Kib</th>
<th>Lang</th>
<th>Medical</th>
<th>Tre1</th>
<th>Test2</th>
<th>Tox</th>
<th>UMNIST</th>
<th>WebACB</th>
<th>WarpAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>KM</td>
<td>0.390</td>
<td>0.537</td>
<td>0.923</td>
<td>0.703</td>
<td>0.375</td>
<td>0.410</td>
<td>0.369</td>
<td>0.410</td>
<td>0.415</td>
<td>0.381</td>
<td>0.315</td>
<td></td>
</tr>
<tr>
<td>KM-best</td>
<td>0.310</td>
<td>0.635</td>
<td>0.488</td>
<td>0.856</td>
<td>0.587</td>
<td>0.471</td>
<td>0.695</td>
<td>0.446</td>
<td>0.482</td>
<td>0.436</td>
<td>0.419</td>
<td>0.319</td>
</tr>
<tr>
<td>KC</td>
<td>0.317</td>
<td>0.580</td>
<td>0.410</td>
<td>0.561</td>
<td>0.593</td>
<td>0.361</td>
<td>0.644</td>
<td>0.320</td>
<td>0.425</td>
<td>0.375</td>
<td>0.333</td>
<td>0.230</td>
</tr>
<tr>
<td>CSPA [3]</td>
<td>0.331</td>
<td>0.638</td>
<td>0.410</td>
<td>0.453</td>
<td>0.414</td>
<td>0.350</td>
<td>0.321</td>
<td>0.285</td>
<td>0.425</td>
<td>0.407</td>
<td>0.277</td>
<td>0.238</td>
</tr>
<tr>
<td>HGPA [3]</td>
<td>0.331</td>
<td>0.378</td>
<td>0.092</td>
<td>0.473</td>
<td>0.036</td>
<td>0.295</td>
<td>0.439</td>
<td>0.296</td>
<td>0.385</td>
<td>0.403</td>
<td>0.252</td>
<td></td>
</tr>
<tr>
<td>MCLA [3]</td>
<td>0.334</td>
<td>0.663</td>
<td>0.400</td>
<td>0.738</td>
<td>0.048</td>
<td>0.402</td>
<td>0.570</td>
<td>0.400</td>
<td>0.415</td>
<td>0.409</td>
<td>0.281</td>
<td>0.232</td>
</tr>
<tr>
<td>NMFC [24]</td>
<td>0.332</td>
<td>0.646</td>
<td>0.414</td>
<td>0.586</td>
<td>0.676</td>
<td>0.379</td>
<td>0.632</td>
<td>0.372</td>
<td>0.427</td>
<td>0.398</td>
<td>0.346</td>
<td>0.223</td>
</tr>
<tr>
<td>BCE [28]</td>
<td>0.008</td>
<td>0.629</td>
<td>0.428</td>
<td>0.355</td>
<td>0.670</td>
<td>0.397</td>
<td>0.621</td>
<td>0.181</td>
<td>0.414</td>
<td>0.393</td>
<td>0.338</td>
<td>0.229</td>
</tr>
<tr>
<td>RCE [39]</td>
<td>0.331</td>
<td>0.630</td>
<td>0.426</td>
<td>0.689</td>
<td>0.714</td>
<td>0.385</td>
<td>0.639</td>
<td>0.411</td>
<td>0.406</td>
<td>0.345</td>
<td>0.305</td>
<td>0.205</td>
</tr>
<tr>
<td>MEC [41]</td>
<td>0.280</td>
<td>0.573</td>
<td>0.394</td>
<td>0.819</td>
<td>0.738</td>
<td>0.363</td>
<td>0.656</td>
<td>-</td>
<td>0.430</td>
<td>0.413</td>
<td>0.376</td>
<td>0.259</td>
</tr>
<tr>
<td>LWEA [72]</td>
<td>0.313</td>
<td>0.694</td>
<td>0.423</td>
<td>0.828</td>
<td>0.746</td>
<td>0.421</td>
<td>0.672</td>
<td>0.574</td>
<td>0.423</td>
<td>0.405</td>
<td>0.332</td>
<td>0.213</td>
</tr>
<tr>
<td>LWGP [72]</td>
<td>0.332</td>
<td>0.655</td>
<td>0.432</td>
<td>0.717</td>
<td>0.650</td>
<td>0.405</td>
<td>0.648</td>
<td>0.429</td>
<td>0.419</td>
<td>0.418</td>
<td>0.346</td>
<td>0.259</td>
</tr>
<tr>
<td>RSEC [61]</td>
<td>0.286</td>
<td>0.536</td>
<td>0.418</td>
<td>0.841</td>
<td>0.522</td>
<td>0.349</td>
<td>0.637</td>
<td>0.422</td>
<td>0.404</td>
<td>0.313</td>
<td>0.310</td>
<td>0.242</td>
</tr>
<tr>
<td>DREC [73]</td>
<td>0.331</td>
<td>0.546</td>
<td>0.428</td>
<td>0.646</td>
<td>0.638</td>
<td>0.393</td>
<td>0.624</td>
<td>0.566</td>
<td>0.421</td>
<td>0.417</td>
<td>0.343</td>
<td>0.206</td>
</tr>
<tr>
<td>SPCE-W</td>
<td>0.262</td>
<td>0.362</td>
<td>0.412</td>
<td>0.845</td>
<td>0.909</td>
<td>0.431</td>
<td>0.663</td>
<td>0.580</td>
<td>0.384</td>
<td>0.375</td>
<td>0.331</td>
<td>0.218</td>
</tr>
<tr>
<td>SPCE-fixW</td>
<td>0.348</td>
<td>0.689</td>
<td>0.456</td>
<td>0.856</td>
<td>0.907</td>
<td>0.436</td>
<td>0.691</td>
<td>0.603</td>
<td>0.425</td>
<td>0.412</td>
<td>0.345</td>
<td>0.262</td>
</tr>
<tr>
<td>SPCE</td>
<td>0.360</td>
<td>0.700</td>
<td>0.442</td>
<td>0.866</td>
<td>0.913</td>
<td>0.453</td>
<td>0.735</td>
<td>0.665</td>
<td>0.449</td>
<td>0.434</td>
<td>0.393</td>
<td>0.283</td>
</tr>
</tbody>
</table>

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 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 W are fixed to 1s.
16) SPCE-fixW: It is our method with fixed 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 W, i.e., in each iteration, 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 $k(x_i, x_j) = e^{-((||x_i-x_j||^2)/2\sigma^2)}$ to construct the Laplacian matrix for spectral clustering. $\sigma$ is the bandwidth parameter, and in our experiments, we set $\sigma = d \ast [0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100]$ for ensemble, i.e., we
TABLE IV
NMI RESULTS ON ALL THE DATA SETS (K-MEANS BASED)

<table>
<thead>
<tr>
<th>Methods</th>
<th>AR</th>
<th>Coil20</th>
<th>GLIOMA</th>
<th>K1b</th>
<th>Lung</th>
<th>Medical</th>
<th>Tr41</th>
<th>Tdt2</th>
<th>tox</th>
<th>UMIST</th>
<th>WebACE</th>
<th>WarpAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>KM</td>
<td>0.648±0.008</td>
<td>0.708±0.025</td>
<td>0.806±0.039</td>
<td>0.549±0.061</td>
<td>0.528±0.060</td>
<td>0.421±0.029</td>
<td>0.584±0.051</td>
<td>0.611±0.007</td>
<td>0.137±0.004</td>
<td>0.383±0.019</td>
<td>0.057±0.027</td>
<td>0.104±0.046</td>
</tr>
<tr>
<td>KM-best</td>
<td>0.561±0.004</td>
<td>0.753±0.009</td>
<td>0.235±0.023</td>
<td>0.685±0.030</td>
<td>0.656±0.062</td>
<td>0.481±0.019</td>
<td>0.671±0.025</td>
<td>0.624±0.006</td>
<td>0.216±0.009</td>
<td>0.409±0.013</td>
<td>0.296±0.018</td>
<td></td>
</tr>
<tr>
<td>KC</td>
<td>0.677±0.006</td>
<td>0.732±0.031</td>
<td>0.152±0.029</td>
<td>0.500±0.023</td>
<td>0.454±0.072</td>
<td>0.426±0.020</td>
<td>0.654±0.028</td>
<td>0.580±0.009</td>
<td>0.134±0.007</td>
<td>0.376±0.019</td>
<td>0.195±0.030</td>
<td></td>
</tr>
<tr>
<td>CSP [3]</td>
<td>0.698±0.022</td>
<td>0.751±0.018</td>
<td>0.172±0.028</td>
<td>0.407±0.007</td>
<td>0.371±0.019</td>
<td>0.399±0.013</td>
<td>0.592±0.015</td>
<td>0.559±0.003</td>
<td>0.144±0.005</td>
<td>0.347±0.016</td>
<td>0.204±0.023</td>
<td></td>
</tr>
<tr>
<td>HGPA [3]</td>
<td>0.693±0.004</td>
<td>0.671±0.024</td>
<td>0.151±0.036</td>
<td>0.392±0.074</td>
<td>0.337±0.048</td>
<td>0.361±0.033</td>
<td>0.508±0.035</td>
<td>0.539±0.008</td>
<td>0.108±0.021</td>
<td>0.382±0.026</td>
<td>0.218±0.038</td>
<td></td>
</tr>
<tr>
<td>MCLA [3]</td>
<td>0.690±0.007</td>
<td>0.761±0.017</td>
<td>0.133±0.029</td>
<td>0.584±0.070</td>
<td>0.326±0.017</td>
<td>0.430±0.019</td>
<td>0.604±0.024</td>
<td>0.607±0.004</td>
<td>0.133±0.017</td>
<td>0.368±0.016</td>
<td>0.186±0.029</td>
<td></td>
</tr>
<tr>
<td>NMFC [24]</td>
<td>0.685±0.004</td>
<td>0.780±0.030</td>
<td>0.155±0.027</td>
<td>0.500±0.021</td>
<td>0.320±0.066</td>
<td>0.426±0.019</td>
<td>0.651±0.039</td>
<td>0.593±0.004</td>
<td>0.143±0.018</td>
<td>0.398±0.022</td>
<td>0.197±0.031</td>
<td></td>
</tr>
<tr>
<td>BCE [28]</td>
<td>0.000±0.000</td>
<td>0.751±0.024</td>
<td>0.166±0.030</td>
<td>0.541±0.067</td>
<td>0.498±0.025</td>
<td>0.450±0.036</td>
<td>0.640±0.000</td>
<td>0.617±0.003</td>
<td>0.137±0.008</td>
<td>0.381±0.005</td>
<td>0.200±0.030</td>
<td></td>
</tr>
<tr>
<td>RCE [39]</td>
<td>0.676±0.004</td>
<td>0.759±0.012</td>
<td>0.162±0.016</td>
<td>0.607±0.010</td>
<td>0.325±0.032</td>
<td>0.448±0.019</td>
<td>0.650±0.024</td>
<td>0.667±0.011</td>
<td>0.134±0.017</td>
<td>0.371±0.014</td>
<td>0.178±0.026</td>
<td></td>
</tr>
<tr>
<td>MEC [41]</td>
<td>0.692±0.017</td>
<td>0.736±0.026</td>
<td>0.131±0.043</td>
<td>0.682±0.071</td>
<td>0.362±0.087</td>
<td>0.409±0.029</td>
<td>0.676±0.027</td>
<td>0.676±0.031</td>
<td>0.131±0.024</td>
<td>0.397±0.030</td>
<td>0.206±0.030</td>
<td></td>
</tr>
<tr>
<td>LWHA [72]</td>
<td>0.664±0.006</td>
<td>0.738±0.011</td>
<td>0.169±0.021</td>
<td>0.695±0.065</td>
<td>0.336±0.030</td>
<td>0.419±0.015</td>
<td>0.678±0.039</td>
<td>0.718±0.009</td>
<td>0.124±0.006</td>
<td>0.330±0.008</td>
<td>0.184±0.028</td>
<td></td>
</tr>
<tr>
<td>LWGP [72]</td>
<td>0.678±0.006</td>
<td>0.764±0.013</td>
<td>0.168±0.018</td>
<td>0.612±0.014</td>
<td>0.499±0.023</td>
<td>0.427±0.011</td>
<td>0.654±0.005</td>
<td>0.627±0.013</td>
<td>0.133±0.005</td>
<td>0.369±0.014</td>
<td>0.195±0.025</td>
<td></td>
</tr>
<tr>
<td>RSEC [61]</td>
<td>0.590±0.015</td>
<td>0.702±0.014</td>
<td>0.154±0.054</td>
<td>0.682±0.063</td>
<td>0.404±0.049</td>
<td>0.404±0.048</td>
<td>0.634±0.034</td>
<td>0.592±0.014</td>
<td>0.118±0.014</td>
<td>0.320±0.021</td>
<td>0.188±0.028</td>
<td></td>
</tr>
<tr>
<td>DREC [73]</td>
<td>0.677±0.004</td>
<td>0.722±0.017</td>
<td>0.164±0.019</td>
<td>0.577±0.041</td>
<td>0.465±0.039</td>
<td>0.481±0.020</td>
<td>0.651±0.017</td>
<td>0.597±0.001</td>
<td>0.119±0.008</td>
<td>0.404±0.008</td>
<td>0.184±0.019</td>
<td></td>
</tr>
<tr>
<td>SPCE-W</td>
<td>0.556±0.019</td>
<td>0.711±0.050</td>
<td>0.141±0.024</td>
<td>0.586±0.115</td>
<td>0.753±0.016</td>
<td>0.333±0.040</td>
<td>0.609±0.135</td>
<td>0.693±0.005</td>
<td>0.104±0.014</td>
<td>0.542±0.014</td>
<td>0.192±0.050</td>
<td>0.259±0.014</td>
</tr>
<tr>
<td>SPCE-bxW</td>
<td>0.737±0.010</td>
<td>0.798±0.016</td>
<td>0.178±0.022</td>
<td>0.671±0.073</td>
<td>0.730±0.035</td>
<td>0.355±0.023</td>
<td>0.683±0.051</td>
<td>0.661±0.012</td>
<td>0.129±0.022</td>
<td>0.645±0.023</td>
<td>0.232±0.019</td>
<td>0.416±0.014</td>
</tr>
<tr>
<td>SPCE</td>
<td>0.740±0.008</td>
<td>0.805±0.010</td>
<td>0.301±0.015</td>
<td>0.699±0.060</td>
<td>0.742±0.013</td>
<td>0.455±0.013</td>
<td>0.685±0.059</td>
<td>0.713±0.016</td>
<td>0.196±0.014</td>
<td>0.663±0.048</td>
<td>0.408±0.043</td>
<td>0.436±0.013</td>
</tr>
</tbody>
</table>

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) 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 S. (b) RCE. (c) MEC. (d) RSEC. (e) SPCE.

ensemble nine spectral clustering results, where $d$ is the mean distance between all pairs $(x_i, x_j)$. 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.
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 \( \gamma \) as \( \gamma = \frac{m^2}{\theta^2} \) (where \( m \) is the number of base clustering results), as introduced in Section III-E, and tune \( \theta = \{0, 0.1, 0.2, \ldots, 0.9\} \) to control the sparsity. Note that \( \theta = 0 \) means we drop the regularized term \( S_0 \). We tune

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 \( \gamma \) as \( \gamma = \frac{m^2}{\theta^2} \) (where \( m \) is the number of base clustering results), as introduced in Section III-E, and tune \( \theta = \{0, 0.1, 0.2, \ldots, 0.9\} \) to control the sparsity. Note that \( \theta = 0 \) means we drop the regularized term \( S_0 \). We tune...
### TABLE VII

<table>
<thead>
<tr>
<th>Methods</th>
<th>AR</th>
<th>Coil20</th>
<th>GLIOMA</th>
<th>Kib</th>
<th>Lung</th>
<th>Medical</th>
<th>Tr41</th>
<th>Tdt2</th>
<th>Toy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC</td>
<td>0.562 ± 0.156</td>
<td>0.618 ± 0.020</td>
<td>0.147 ± 0.043</td>
<td>0.523 ± 0.330</td>
<td>0.383 ± 0.221</td>
<td>0.321 ± 0.171</td>
<td>0.444 ± 0.296</td>
<td>0.416 ± 0.270</td>
<td>0.128 ± 0.073</td>
</tr>
<tr>
<td>SC-best</td>
<td>0.674 ± 0.005</td>
<td>0.792 ± 0.013</td>
<td>0.190 ± 0.012</td>
<td>0.814 ± 0.009</td>
<td>0.622 ± 0.015</td>
<td>0.464 ± 0.006</td>
<td>0.684 ± 0.020</td>
<td>0.620 ± 0.008</td>
<td>0.202 ± 0.005</td>
</tr>
<tr>
<td>KC</td>
<td>0.547 ± 0.018</td>
<td>0.765 ± 0.039</td>
<td>0.170 ± 0.020</td>
<td>0.665 ± 0.055</td>
<td>0.447 ± 0.064</td>
<td>0.392 ± 0.028</td>
<td>0.528 ± 0.011</td>
<td>0.525 ± 0.011</td>
<td>0.362 ± 0.002</td>
</tr>
<tr>
<td>CPA [3]</td>
<td>0.006 ± 0.006</td>
<td>0.003 ± 0.004</td>
<td>0.001 ± 0.005</td>
<td>0.019 ± 0.008</td>
<td>0.011 ± 0.017</td>
<td>0.012 ± 0.016</td>
<td>0.000 ± 0.000</td>
<td>0.014 ± 0.009</td>
<td>0.000 ± 0.000</td>
</tr>
<tr>
<td>HGPA [3]</td>
<td>0.681 ± 0.004</td>
<td>0.687 ± 0.045</td>
<td>0.149 ± 0.061</td>
<td>0.348 ± 0.058</td>
<td>0.287 ± 0.042</td>
<td>0.389 ± 0.019</td>
<td>0.320 ± 0.057</td>
<td>0.377 ± 0.012</td>
<td>0.047 ± 0.003</td>
</tr>
<tr>
<td>MCLA [3]</td>
<td>0.001 ± 0.001</td>
<td>0.009 ± 0.015</td>
<td>0.003 ± 0.015</td>
<td>0.014 ± 0.013</td>
<td>0.012 ± 0.019</td>
<td>0.021 ± 0.021</td>
<td>0.030 ± 0.045</td>
<td>0.036 ± 0.006</td>
<td>0.000 ± 0.000</td>
</tr>
<tr>
<td>NMFC [24]</td>
<td>0.535 ± 0.020</td>
<td>0.793 ± 0.010</td>
<td>0.139 ± 0.020</td>
<td>0.677 ± 0.054</td>
<td>0.454 ± 0.077</td>
<td>0.411 ± 0.032</td>
<td>0.623 ± 0.037</td>
<td>0.538 ± 0.037</td>
<td>0.171 ± 0.005</td>
</tr>
<tr>
<td>BCI [28]</td>
<td>0.000 ± 0.000</td>
<td>0.605 ± 0.018</td>
<td>0.181 ± 0.060</td>
<td>0.714 ± 0.086</td>
<td>0.486 ± 0.011</td>
<td>0.431 ± 0.020</td>
<td>0.574 ± 0.001</td>
<td>0.000 ± 0.034</td>
<td>0.166 ± 0.013</td>
</tr>
<tr>
<td>RCE [39]</td>
<td>0.658 ± 0.006</td>
<td>0.824 ± 0.017</td>
<td>0.781 ± 0.060</td>
<td>0.560 ± 0.454</td>
<td>0.670 ± 0.015</td>
<td>0.649 ± 0.025</td>
<td>0.325 ± 0.025</td>
<td>0.219 ± 0.011</td>
<td></td>
</tr>
<tr>
<td>MEC [41]</td>
<td>0.499 ± 0.011</td>
<td>0.768 ± 0.017</td>
<td>0.356 ± 0.454</td>
<td>0.410 ± 0.500</td>
<td>0.660 ± 0.167</td>
<td>0.389 ± 0.224</td>
<td>0.176 ± 0.011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LWEA [72]</td>
<td>0.681 ± 0.004</td>
<td>0.824 ± 0.173</td>
<td>0.795 ± 0.558</td>
<td>0.497 ± 0.635</td>
<td>0.651 ± 0.173</td>
<td>0.658 ± 0.277</td>
<td>0.215 ± 0.011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LWGP [72]</td>
<td>0.677 ± 0.003</td>
<td>0.825 ± 0.173</td>
<td>0.788 ± 0.560</td>
<td>0.456 ± 0.687</td>
<td>0.638 ± 0.173</td>
<td>0.657 ± 0.315</td>
<td>0.218 ± 0.011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSEC [61]</td>
<td>0.536 ± 0.023</td>
<td>0.575 ± 0.014</td>
<td>0.319 ± 0.384</td>
<td>0.264 ± 0.500</td>
<td>0.472 ± 0.093</td>
<td>0.329 ± 0.216</td>
<td>0.167 ± 0.011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DREC [73]</td>
<td>0.683 ± 0.004</td>
<td>0.827 ± 0.172</td>
<td>0.798 ± 0.559</td>
<td>0.435 ± 0.658</td>
<td>0.607 ± 0.173</td>
<td>0.654 ± 0.318</td>
<td>0.219 ± 0.011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPCE-W [22]</td>
<td>0.623 ± 0.054</td>
<td>0.720 ± 0.164</td>
<td>0.334 ± 0.474</td>
<td>0.325 ± 0.524</td>
<td>0.405 ± 0.177</td>
<td>0.581 ± 0.124</td>
<td>0.210 ± 0.031</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPCE-fxW</td>
<td>0.654 ± 0.042</td>
<td>0.810 ± 0.155</td>
<td>0.696 ± 0.521</td>
<td>0.454 ± 0.667</td>
<td>0.602 ± 0.161</td>
<td>0.641 ± 0.230</td>
<td>0.214 ± 0.031</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPCE</td>
<td>0.716 ± 0.008</td>
<td>0.830 ± 0.294</td>
<td>0.808 ± 0.611</td>
<td>0.473 ± 0.685</td>
<td>0.679 ± 0.227</td>
<td>0.673 ± 0.348</td>
<td>0.278 ± 0.043</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 $S$ and $S^{(i)}$. It is equivalent to add the weight on each base clustering, which can reduce the side effect of the bad base clusterings. Moreover,
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, $\theta$ plays a role as a threshold that controls the sparsity of the consensus matrix $S$, i.e., the larger $\theta$ is, the sparser $S$ is. We tune $\theta$ from $\{0.0, 0.1, \ldots, 0.9\}$ 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 $S^0$ 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 $\theta$ from $[0.2, 0.6]$, which can often obtain a relatively good performance.

Table VIII

<table>
<thead>
<tr>
<th>Methods</th>
<th>AR</th>
<th>Coil20</th>
<th>GLIOMA</th>
<th>K1b</th>
<th>Lung</th>
<th>Medical</th>
<th>TTr</th>
<th>TTd2</th>
<th>TExp</th>
<th>UMIST</th>
<th>WebACE</th>
<th>WarpAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC</td>
<td>0.083</td>
<td>0.351</td>
<td>0.070</td>
<td>0.602</td>
<td>0.360</td>
<td>0.134</td>
<td>0.349</td>
<td>0.150</td>
<td>0.090</td>
<td>0.226</td>
<td>0.026</td>
<td>0.032</td>
</tr>
<tr>
<td>SC-best</td>
<td>0.134</td>
<td>0.374</td>
<td>0.108</td>
<td>0.885</td>
<td>0.640</td>
<td>0.246</td>
<td>0.575</td>
<td>0.258</td>
<td>0.155</td>
<td>0.249</td>
<td>0.085</td>
<td>0.064</td>
</tr>
<tr>
<td>KC</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CSPA [3]</td>
<td>0.036</td>
<td>0.652</td>
<td>0.097</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>HGPA</td>
<td>0.142</td>
<td>0.482</td>
<td>0.072</td>
<td>0.276</td>
<td>0.128</td>
<td>0.17</td>
<td>0.201</td>
<td>0.072</td>
<td>0.025</td>
<td>0.334</td>
<td>0.069</td>
<td>0.046</td>
</tr>
<tr>
<td>MCLA</td>
<td>0.000</td>
<td>0.014</td>
<td>0.072</td>
<td>0.266</td>
<td>0.176</td>
<td>0.002</td>
<td>0.005</td>
<td>0.000</td>
<td>0.063</td>
<td>0.000</td>
<td>0.000</td>
<td>0.004</td>
</tr>
<tr>
<td>NMFC [24]</td>
<td>0.026</td>
<td>0.606</td>
<td>0.090</td>
<td>0.764</td>
<td>0.339</td>
<td>0.194</td>
<td>0.820</td>
<td>0.162</td>
<td>0.137</td>
<td>0.035</td>
<td>0.067</td>
<td>0.024</td>
</tr>
<tr>
<td>BCE [28]</td>
<td>0.000</td>
<td>0.431</td>
<td>0.110</td>
<td>0.792</td>
<td>0.394</td>
<td>0.201</td>
<td>0.469</td>
<td>0.000</td>
<td>0.132</td>
<td>0.184</td>
<td>0.046</td>
<td>0.041</td>
</tr>
<tr>
<td>RCE [39]</td>
<td>0.010</td>
<td>0.646</td>
<td>0.100</td>
<td>0.855</td>
<td>0.544</td>
<td>0.225</td>
<td>0.348</td>
<td>-</td>
<td>0.139</td>
<td>0.333</td>
<td>0.054</td>
<td>0.041</td>
</tr>
<tr>
<td>MEC [41]</td>
<td>0.008</td>
<td>0.545</td>
<td>0.101</td>
<td>0.326</td>
<td>0.328</td>
<td>0.186</td>
<td>0.440</td>
<td>-</td>
<td>0.153</td>
<td>0.279</td>
<td>0.032</td>
<td>0.024</td>
</tr>
<tr>
<td>LWEA [72]</td>
<td>0.150</td>
<td>0.631</td>
<td>0.102</td>
<td>0.862</td>
<td>0.541</td>
<td>0.231</td>
<td>0.555</td>
<td>0.292</td>
<td>0.136</td>
<td>0.346</td>
<td>0.016</td>
<td>0.037</td>
</tr>
<tr>
<td>LWGP [72]</td>
<td>0.124</td>
<td>0.629</td>
<td>0.102</td>
<td>0.83</td>
<td>0.545</td>
<td>0.229</td>
<td>0.352</td>
<td>0.270</td>
<td>0.136</td>
<td>0.338</td>
<td>0.042</td>
<td>0.039</td>
</tr>
<tr>
<td>RSEC [61]</td>
<td>0.034</td>
<td>0.291</td>
<td>0.060</td>
<td>0.349</td>
<td>0.312</td>
<td>0.085</td>
<td>0.345</td>
<td>0.135</td>
<td>0.056</td>
<td>0.256</td>
<td>0.043</td>
<td>0.014</td>
</tr>
<tr>
<td>DREC [73]</td>
<td>0.145</td>
<td>0.640</td>
<td>0.102</td>
<td>0.861</td>
<td>0.543</td>
<td>0.256</td>
<td>0.529</td>
<td>0.220</td>
<td>0.136</td>
<td>0.339</td>
<td>0.038</td>
<td>0.041</td>
</tr>
<tr>
<td>SPCE-W</td>
<td>0.099</td>
<td>0.462</td>
<td>0.097</td>
<td>0.421</td>
<td>0.500</td>
<td>0.164</td>
<td>0.399</td>
<td>0.066</td>
<td>0.139</td>
<td>0.262</td>
<td>0.010</td>
<td>0.039</td>
</tr>
<tr>
<td>SPCE-fixW</td>
<td>0.108</td>
<td>0.590</td>
<td>0.089</td>
<td>0.791</td>
<td>0.551</td>
<td>0.225</td>
<td>0.328</td>
<td>0.250</td>
<td>0.126</td>
<td>0.315</td>
<td>0.014</td>
<td>0.043</td>
</tr>
<tr>
<td>SPCE</td>
<td>0.154</td>
<td>0.640</td>
<td>0.115</td>
<td>0.881</td>
<td>0.639</td>
<td>0.249</td>
<td>0.566</td>
<td>0.323</td>
<td>0.152</td>
<td>0.355</td>
<td>0.041</td>
<td>0.053</td>
</tr>
</tbody>
</table>
TABLE IX
RUNNING TIME ON ALL THE DATA SETS WITH 20 BASE CLUSTERINGS (s)

<table>
<thead>
<tr>
<th>Methods</th>
<th>AR</th>
<th>Coil20</th>
<th>GLOMA</th>
<th>K1b</th>
<th>Lung</th>
<th>Medical</th>
<th>Tr41</th>
<th>Tid2</th>
<th>Tet</th>
<th>UMIST</th>
<th>WebACE</th>
<th>Warp-AR</th>
</tr>
</thead>
<tbody>
<tr>
<td>KC</td>
<td>0.128</td>
<td>0.171</td>
<td>0.002</td>
<td>0.468</td>
<td>0.005</td>
<td>0.036</td>
<td>0.036</td>
<td>0.048</td>
<td>55.44</td>
<td>0.004</td>
<td>0.024</td>
<td>0.658</td>
</tr>
<tr>
<td>CSPA [3]</td>
<td>0.403</td>
<td>1.076</td>
<td>0.102</td>
<td>2.490</td>
<td>0.197</td>
<td>0.327</td>
<td>0.502</td>
<td>14.22</td>
<td>0.186</td>
<td>0.249</td>
<td>2.735</td>
<td>0.175</td>
</tr>
<tr>
<td>HGPA [3]</td>
<td>2.600</td>
<td>1.206</td>
<td>0.191</td>
<td>0.342</td>
<td>0.196</td>
<td>0.388</td>
<td>0.318</td>
<td>40.91</td>
<td>0.182</td>
<td>1.175</td>
<td>1.230</td>
<td>0.300</td>
</tr>
<tr>
<td>MCLA [3]</td>
<td>0.953</td>
<td>0.257</td>
<td>0.180</td>
<td>0.210</td>
<td>0.179</td>
<td>0.219</td>
<td>0.203</td>
<td>3.878</td>
<td>0.185</td>
<td>0.219</td>
<td>0.289</td>
<td>0.185</td>
</tr>
<tr>
<td>NMFC [24]</td>
<td>0.218</td>
<td>0.318</td>
<td>0.003</td>
<td>4.132</td>
<td>0.007</td>
<td>0.056</td>
<td>0.317</td>
<td>76.63</td>
<td>0.008</td>
<td>0.037</td>
<td>1.048</td>
<td>0.006</td>
</tr>
<tr>
<td>BCE [28]</td>
<td>80.26</td>
<td>6.964</td>
<td>0.092</td>
<td>5.043</td>
<td>0.411</td>
<td>4.373</td>
<td>2.467</td>
<td>983.8</td>
<td>0.338</td>
<td>3.326</td>
<td>15.03</td>
<td>0.438</td>
</tr>
<tr>
<td>RCE [39]</td>
<td>79.87</td>
<td>251.1</td>
<td>0.209</td>
<td>804.2</td>
<td>2.154</td>
<td>53.94</td>
<td>97.20</td>
<td>-</td>
<td>1.577</td>
<td>31.73</td>
<td>802.1</td>
<td>0.955</td>
</tr>
<tr>
<td>MEC [41]</td>
<td>53.95</td>
<td>207.6</td>
<td>0.089</td>
<td>913.8</td>
<td>0.886</td>
<td>20.85</td>
<td>40.05</td>
<td>-</td>
<td>0.634</td>
<td>13.51</td>
<td>967.4</td>
<td>0.408</td>
</tr>
<tr>
<td>LWBA [72]</td>
<td>0.007</td>
<td>0.002</td>
<td>0.001</td>
<td>0.002</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.011</td>
<td>0.001</td>
<td>0.001</td>
<td>0.002</td>
<td>0.001</td>
</tr>
<tr>
<td>LWGP [72]</td>
<td>1.911</td>
<td>0.031</td>
<td>0.005</td>
<td>0.014</td>
<td>0.008</td>
<td>0.021</td>
<td>0.014</td>
<td>1.634</td>
<td>0.007</td>
<td>0.024</td>
<td>0.035</td>
<td>0.010</td>
</tr>
<tr>
<td>RSEC [61]</td>
<td>11.64</td>
<td>75.82</td>
<td>0.030</td>
<td>412.1</td>
<td>0.317</td>
<td>5.631</td>
<td>10.95</td>
<td>30127</td>
<td>0.232</td>
<td>3.504</td>
<td>411.1</td>
<td>0.130</td>
</tr>
<tr>
<td>DREC [73]</td>
<td>8.092</td>
<td>0.809</td>
<td>0.069</td>
<td>1.278</td>
<td>0.123</td>
<td>0.989</td>
<td>0.807</td>
<td>2981</td>
<td>0.101</td>
<td>0.609</td>
<td>2.690</td>
<td>0.028</td>
</tr>
<tr>
<td>SFCE</td>
<td>9.495</td>
<td>31.92</td>
<td>0.016</td>
<td>98.55</td>
<td>0.123</td>
<td>4.973</td>
<td>6.806</td>
<td>3423</td>
<td>0.103</td>
<td>3.852</td>
<td>89.82</td>
<td>0.115</td>
</tr>
</tbody>
</table>

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

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.

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 (TPAMI), the IEEE transactions on knowledge and data engineering (TKDE), 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.

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.

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.

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.