Active Clustering Ensemble With Self-Paced Learning

Peng Zhou, *Member, IEEE*, Bicheng Sun, Xinwang Liu, *Senior Member, IEEE*, Liang Du, and Xuejun Li, *Member, IEEE*

***Abstract*— A clustering ensemble provides an elegant framework to learn a consensus result from multiple prespecified clustering partitions. Though conventional clustering ensemble methods achieve promising performance in various applications, we observe that they may usually be misled by some unreliable instances due to the absence of labels. To tackle this issue, we propose a novel active clustering ensemble method, which selects the uncertain or unreliable data for querying the annotations in the process of the ensemble. To fulfill this idea, we seamlessly integrate the active clustering ensemble method into a self-paced learning framework, leading to a novel self-paced active clustering ensemble (SPACE) method. The proposed SPACE can jointly select unreliable data to label via automatically evaluating their difficulty and applying easy data to ensemble the clusterings. In this way, these two tasks can be boosted by each other, with the aim to achieve better clustering performance. The experimental results on benchmark datasets demonstrate the significant effectiveness of our method. The codes of this article are released in http://Doctor-Nobody.github.io/codes/space.zip.**

***Index Terms*— Active learning, clustering ensemble, self-paced learning.**

I. Introduction

CLUSTERING ensemble provides an elegant framework for integrating multiple weak base clusterings to obtain a consensus and stable result [1]. Many clustering ensemble methods have been proposed in recent decades [2], [3], [4], [5] and applied to various applications. For example, Liu et al. [3] developed an ensemble method to handle incomplete data; Li et al. [4] combined multiple clustering results by analyzing the stability of each instance; and Bai et al. [5] ensembled multiple k-means results to tackle the nonlinear data. By integrating multiple weak clustering results, clustering ensemble can alleviate the robustness and stability problems in single-clustering methods to some extent.

Despite demonstrating promising clustering performance in various applications, the above-mentioned ensemble methods may be usually misled by some unreliable data due to the absence of labels. Since in clustering ensemble tasks, we often do not access the original features and the labels of data, these ensemble methods are often based on the majority voting of the base results. However, due to the limitation of the base clustering method, e.g., k-means can hardly handle nonsphere-shape data, it happens that the majority of base clustering may be unreliable and often make mistakes, and thus, it may mislead the ensemble methods. Fig. 1 shows an example of the two-moon data. We run k-means on the two-moon data four times with different initializations to obtain four base clustering results. Notice that the instances $\mathbf{x}_1$ and $\mathbf{x}_2$ (shown in the dotted line circle in Fig. 1) are put in the same cluster in all four base results. Since conventional clustering ensemble methods are based on the majority voting and do not access the original feature and the labels of data, they often mistakenly believe that $\mathbf{x}_1$ and $\mathbf{x}_2$ should be in the same cluster. To address this issue, some semisupervised clustering ensemble methods are proposed, such as [6], [7], and [8]. These methods used the pregiven must-link (i.e., the two instances belong to the same cluster) and cannot-link (i.e., the two instances belong to different clusters) pairwise constraints to guide the ensemble. However, in real-world applications, the performance of these semisupervised methods highly depends on the selection of the supervised information. Unfortunately, how to choose the supervised information itself is still a tough task.

To tackle this problem, in this article, we propose a novel active clustering ensemble method. We observe that self-paced learning can learn the difficulty of each instance and use easy ones for training, whereas active learning often selects the difficult ones for querying the annotations. As a result, we can automatically select data for querying the human annotations and apply these annotations to guide the ensemble by utilizing this complementarity of self-paced learning and active learning. To do so, we propose to seamlessly integrate the active clustering ensemble into a self-paced learning framework, leading to a novel self-paced active clustering ensemble (SPACE) method. In this method, we use self-paced learning to automatically learn the difficulty of the data and use the easy data for ensemble learning, and meanwhile, select the difficult data for querying to label and apply the human annotations as constraint information to guide the clustering ensemble.

To this end, we carefully design a unified objective function to integrate the clustering ensemble, self-paced learning, and active learning seamlessly. To optimize the introduced objective function, we propose an effective iterative algorithm that involves two steps: S-step for selecting queries and E-step

Manuscript received 3 May 2022; revised 5 January 2023; accepted 1 March 2023. This work was supported by the National Natural Science Foundation of China under Grant 62176001, Grant 61806003, Grant 61976129, Grant 61922088, and Grant 61972001. *(Corresponding author: Xuejun Li.)*

Peng Zhou is with the Anhui Provincial International Joint Research Center for Advanced Technology in Medical Imaging, 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).

Bicheng Sun and Xuejun Li are with the Anhui Provincial International Joint Research Center for Advanced Technology in Medical Imaging, School of Computer Science and Technology, Anhui University, Hefei 230601, China (e-mail: e19301144@stu.ahu.edu.cn; xjli@ahu.edu.cn).

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

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

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2023.3252586.

Digital Object Identifier 10.1109/TNNLS.2023.3252586

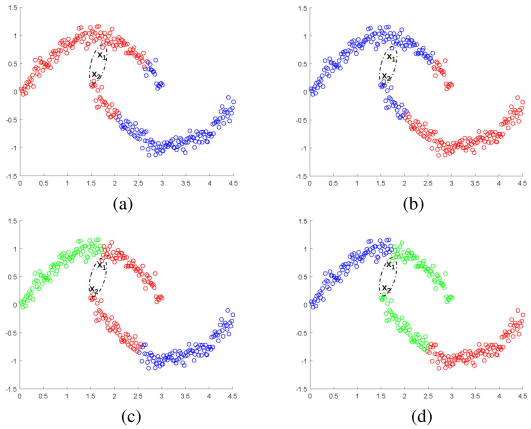

Fig. 1. Four base results of k-means on two-moon data. (a) First base result of k-means (c = 2). (b) Second base result of k-means (c = 2). (c) Third base result of k-means (c = 3). (d) Fourth base result of k-means (c = 3).

for *e*nsemble. In the *S*-step, we automatically estimate the difficulty of each data and select difficult ones for querying; in the *E*-step, we ensemble multiple base partitions to preserve the human annotations. Although the subproblems in both steps contain some complex constraints and regularized terms, we can find the closed-form solution for each subproblem. Note that due to the carefully designed objective function, SPACE can directly obtain the clustering result without any uncertain postprocedure. Thus, together with a reasonable initialization, it can always provide the clustering result without uncertainness and randomness. Furthermore, we theoretically analyze the setting of the parameters and hyperparameters to make the model easy to use. At last, the extensive experiments on benchmark datasets demonstrate the effectiveness and superiority of the proposed algorithm. Notice that compared with our previous self-paced clustering ensemble work [9], there are three significant differences: 1) [9] is an unsupervised clustering ensemble. However, the proposed work focuses on the active clustering ensemble setting. To the best of our knowledge, the active clustering ensemble without accessing the original features of data is new and quite underexplored. Compared with the unsupervised setting or even semisupervised setting, it can leverage a few human annotations to obtain much better performance. 2) Reference [9] only uses the easy data for ensemble whereas ignoring the difficult data like many other conventional self-paced learning methods. The proposed SPACE extends self-paced learning to the active learning setting. It can fully use both the easy data and difficult data, i.e., it uses easy data for ensemble learning and queries difficult data for annotations and then applies the annotations of difficult data to guide the ensemble learning. 3) To fully use the limited number of annotations, we propose a schema for propagating both the must-link and cannot-link constraints. By the propagation methods, despite the limited number of annotations, the proposed SPACE can leverage as much information of annotations as possible.

The main contributions are summarized as follows.

1) We propose a novel SPACE framework. Different from conventional unsupervised or semisupervised clustering ensemble methods, our framework can automatically select important data for annotation and ensemble the data by propagating both the must-link and cannot-link constraints. In this way, the proposed framework can effectively adopt a few annotations to improve the clustering ensemble.
2) We develop an effective algorithm to optimize the objective function. The optimization can be divided into two explainable iterative steps. Moreover, the optimization will not introduce any uncertainness and randomness. We also provide some guidance on the setting of parameters and hyperparameters, which makes it easy to use.
3) We compare our method with state-of-the-art unsupervised and semisupervised clustering ensemble methods in experiments, and the results show that our algorithm can significantly outperform the unsupervised and semisupervised methods, which demonstrates its effectiveness and superiority.

II. Related Work

We first introduce some notations of this article. Throughout this article, we use boldface lowercase and uppercase letters to denote vectors and matrices, 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. We use $\mathbf{M}_{i\cdot}$ and $\mathbf{M}_{\cdot i}$ to denote the ith row and ith column of $\mathbf{M}$, respectively. Then, we will review some related work about clustering ensemble, self-paced learning, and active learning, respectively, in Sections II-A–II-C.

A. Clustering Ensemble

A clustering ensemble, also known as consensus clustering, was first introduced by Strehl and Ghosh [10], which aimed to integrate multiple weak clustering results to obtain a consensus and robust one. Since it can alleviate the stability and robustness problems, which single-clustering methods may suffer from to some extent, it has attracted increasing attention in recent years. Roughly speaking, a clustering ensemble can be categorized into two classes based on whether it accesses the original data features. In the works which access the original data [11], [12], [13], when they ensembled multiple clusterings, they often used the original data to guide the ensemble, and thus, could further improve clustering performance.

However, in the second class, i.e., ones without accessing the original features of data, the problem becomes more challenging, because we just have the multiple clustering results as inputs. Despite this, this kind of clustering ensemble could be applied in more fields, such as in the scenario of distributed data or attributes. Moreover, since this kind of ensemble method does not take the original data as input, it can protect the privacy of data to some extent. Therefore, it has attracted more attention [10], [14], [15].

This article focuses on the class without accessing the original data. We take multiple base partitions as inputs to learn a consensus partition. To achieve this, many unsupervised learning techniques have been extended to combine base results. For example, Zhou and Tang [14] proposed a k-means-based clustering ensemble method with the alignment technique. Besides k-means, since spectral clustering was also a widely studied clustering technique, it has also often been extended in ensemble learning [16], [17], [18]. Some methods took the quality and diversity of base clusterings into consideration to guide the ensemble learning [19], [20]. Some works ensembled multiple clustering results by graph-based methods [21], [22].

As introduced earlier, although these ensemble methods can alleviate the robustness and stability problems to some extent, since they are still unsupervised methods without the guidance of the labels, they may still be misled by the unreliable base results. To address this issue, some works use the supervised information to guide the ensemble, leading to the semisupervised clustering ensemble [6], [7], [8], [23], [24]. In these methods, they use the pairwise constraints as the supervised information, i.e., if two instances belong to the same cluster, there is a must-link constraint on the pairs; if they belong to the different clusters, the constraint between them is a cannot-link constraint. Based on these pairwise constraints, these methods propagate the constraints in the process of the ensemble. For example, Lai et al. [8] assigned a weight on each cluster according to the pairwise constraints, and then emsembled the multiple clustering results to obtain a robust result; Yu et al. [24] applied constraint weighting and ensemble member weighting to guide the ensemble. These semisupervised methods often use the pregiven constraints for learning, while ignoring how to select the informative constraints.

In this article, we will make use of the complementarity of self-paced learning and active learning to automatically select the most uncertain pairs for annotation, which will further improve the performance of the ensemble. Notice that there exist some active clustering ensemble methods, such as [25] and [26]. For example, Barr et al. [25] first computed the pairwise distance of all data, then selected the instances for annotation according to the rank of the distance, and last ensembled the base clustering results with the annotation; Shi et al. [26] designed a fast and effective active clustering ensemble method by an active density peak clustering method, which also needed the distance between all data. Therefore, their methods need to access the original data to decide which data should be queried. This is different from our setting, which does not use the original data. As introduced earlier, our setting is more applicable and also more challenging.

B. Self-Paced Learning

The key idea of self-paced learning is to automatically and incrementally use data for learning, where easy data are used first and difficult ones are then involved gradually [27]. Since it is in line with the learning process of humans, it has been widely used in various machine learning tasks [28], [29], [30], [31], [32], [33]. For example, Bengio et al. [28] applied it to tackle the local optimum problem in nonconvex optimization; in [34] and [30], it is plugged in the multitask learning; Huang et al. [33], [35] applied the self-paced learning to the multiview learning; Shao et al. [36] adopted the self-paced learning to the label distribution learning; and Soviany et al. [37] proposed the curriculum self-paced learning method for the cross-domain object detection.

More formally, given a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$ with n instances, where $\mathbf{x}_i \in \mathbb{R}^d$ is the ith instance and y_i is its label, we denote $h(\mathbf{x}_i, \boldsymbol{\theta})$ as the decision function of a model and $\boldsymbol{\theta}$ as the model parameters. Then, in a machine learning model, we need to minimize the loss $\mathcal{L}(h(\mathbf{x}_i, \boldsymbol{\theta}), y_i)$ between the decision function and the true label. According to [38] and [39], self-paced learning imposes a weight on the loss of each instance to represent the difficulty of the instance and introduces a regularization term on the weights. More formally, we obtain the following objective function:

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

where λ is an adaptive age parameter to control the learning pace, which will grow in the process of learning, and $\Omega(\lambda, \mathbf{w})$ is the regularization term on weights. The self-paced learning optimizes (1) via alternating minimization. Solving $\mathbf{w}$ by fixing $\boldsymbol{\theta}$ is to assign the weight on each data; solving $\boldsymbol{\theta}$ by fixing $\mathbf{w}$ is to train the model with easy data. Intuitively, the easier instance will have a larger weight w_i. Moreover, with the process of learning, more and more instances will become easy and involved in learning. In this work, we will integrate self-paced learning into an active learning framework for clustering ensemble.

C. Active Learning

Active learning is a machine learning methodology to automatically selects informative instances for annotation when handling a large amount of unlabeled data [40], [41]. Its goal is to train a classifier that has good generalization performance with only the selected labeled instances. In this article, we focus on the batch mode active learning, which selects a batch of instances for annotation in each iteration [42], [43], [44], [45], [46], [47], [48]. For example, Hoi et al. [42] applied the Fisher information matrix to select a batch of informative instances for annotation and used the annotated data to train a classifier; different from [42] which adopted Fisher information, Wang et al. [44] used the α-relative Pearson divergence for batch selection; Yang et al. [49] proposed an uncertainty sampling-based active learning method, which maximized the diversity of selected data; and Liu et al. [50], [51] designed the pairwise active learning, which considers both the uncertainty and diversity, and applied it to the person reidentification task.

More formally, considering the dataset $\mathcal{X}$ as defined earlier, we divide it into two sets: labeled set $\mathcal{L}$ and unlabeled set $\mathcal{U}$ such that $\mathcal{L} \cup \mathcal{U} = \mathcal{X}$ and $\mathcal{L} \cap \mathcal{U} = \varnothing$. If $\mathbf{x}_i \in \mathcal{L}$, its label y_i is revealed by a human labeler; otherwise, y_i is unknown. Batch mode active learning methods iteratively select a batch of instances $\mathcal{S} \subset \mathcal{U}$ with a given batch size k (a predefined constant) for labeling until there is no budget.

In this article, since we focus on the clustering task, we select a pair of instances $(\mathbf{x}_p, \mathbf{x}_q)$ for querying whether these two instances belong to the same cluster or not, instead of directly labeling which class should $\mathbf{x}_p$ or $\mathbf{x}_q$ belong to. In the clustering task, since the label space is often unknown, our scheme to label two instances with must-link and cannot-link relations is simpler and more practical.

III. Self-Paced Active Clustering Ensemble

In this section, we introduce our SPACE method. First, we show some notations and their descriptions in Table I. Then, we provide the framework.

A. Framework

Following [15], [52], [53], and [54], we first construct a *connective matrix* $\mathbf{S}^{(i)} \in \mathbb{R}^{n \times n}$ for a base partition, whose (p, q)th element $S_{pq} = 1$, if $\mathbf{x}_p$ and $\mathbf{x}_q$ belong to the same cluster, and $S_{pq} = 0$, otherwise. Then, we will ensemble

TABLE I
NOTATIONS AND DESCRIPTIONS USED IN OUR METHOD

Notation	Description
n, c	Number of instances and clusters, respectively.
m	Number of base clusterings.
$\mathbf{S}^{(k)} \in \{0,1\}^{n\times n}$	The connective matrix of the base clustering.
$\mathbf{S} \in [0,1]^{n\times n}$	The consensus matrix.
$\boldsymbol{\alpha} \in [0,1]^m$	The weight vector of base clusterings.
$\mathbf{W} \in [0,1]^{n\times n}$	The weight matrix.
$\mathcal{M}, \quad \mathcal{C}$	The must-link and cannot-link sets, respectively.

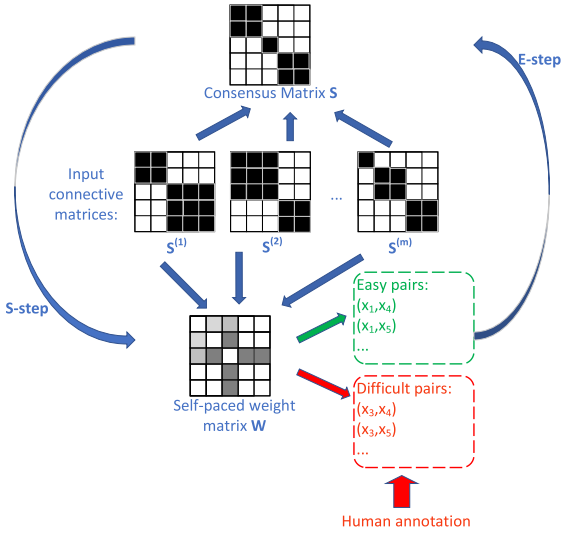

Fig. 2. Framework of SPACE. It contains two iterative steps: S-step and E-step. In the S-step, we learn the self-paced weight matrix $\mathbf{W}$ with input connective matrices $\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(m)}$ and consensus matrix $\mathbf{S}$, and then select easy pairs for self-paced learning and difficult pairs for annotation; in E-step, we apply the easy pairs together with the labeled difficult pairs to learn the consensus matrix $\mathbf{S}$.

the m base connective matrices to learn a consensus matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$. Note that our method just takes multiple base partitions as inputs without accessing the original data features $\mathbf{x}_1, \ldots, \mathbf{x}_n$.

We seamlessly integrate the clustering ensemble and constraints active selection into a unified self-paced framework. Our method can be basically divided into two iterative steps: S-step for *s*electing pairs to label and E-step for *e*nsemble. Fig. 2 illustrates the whole framework of our method. In Sections III-B–III-D, we will introduce it in more detail.

B. Objective Function

A natural way to learn the consensus matrix $\mathbf{S}$ is to minimize the disagreement between it and all connective matrices. To this end, we can minimize the following objective function:

$$\begin{aligned} \min_{\mathbf{S},\boldsymbol{\alpha}} \quad & \sum_{i=1}^{m} \alpha_i^2 \left\| \mathbf{S} - \mathbf{S}^{(i)} \right\|_F^2 \\ \text{s.t.} \quad & \sum_{i=1}^{m} \alpha_i = 1, \quad \forall i : 0 \le \alpha_i \le 1 \\ & \mathbf{S} = \mathbf{S}^T \quad \forall p, q : 0 \le S_{pq} \le 1 \\ & \forall (\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{M} : S_{pq} = 1 \quad \forall (\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{C} : S_{pq} = 0 \end{aligned} \tag{2}$$

where α_i is the weight of the ith clustering result. Intuitively, if $\|\mathbf{S}-\mathbf{S}^{(i)}\|_F^2$ is large, i.e., the quality of $\mathbf{S}^{(i)}$ is low, to minimize (2), α_i should be small. Since all connective matrices are symmetric and their elements are either 0 or 1, we wish the learned consensus matrix $\mathbf{S}$ also be symmetric and its elements are in the range $[0, 1]$. $\mathcal{M}$ and $\mathcal{C}$ indicate the set of must-link and cannot-link, respectively. In the beginning, the two sets are empty, since we do not have any labeled information. Then, in the process of learning, we actively select some pairs for annotation. For any selected pair $(\mathbf{x}_p, \mathbf{x}_q)$, if human labels that $\mathbf{x}_p$ and $\mathbf{x}_q$ belong to the same cluster, then we add it to the must-link set $\mathcal{M}$, or otherwise, we add it to the cannot-link set $\mathcal{C}$. In Section III-C, we will introduce how to select such pairs for annotation.

Note that $\mathbf{S}$ in (2) contains $n \times n$ variables to be learned, and most of them are difficult to learn since the base results $\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(m)}$ are unreliable. To tackle this problem, we plug (2) into the self-paced learning framework. We first use easy data or reliable data to learn the model and then involve more and more difficult data in learning. To this end, we need to determine the difficulty of each data pair first. Intuitively, given a data pair $\mathbf{x}_p$ and $\mathbf{x}_q$, if most $\mathbf{S}^{(i)}$'s agree with each other, we believe that this is an easy pair. We use $\sum_{i=1}^{m}(S_{pq}-S_{pq}^{(i)})^2$ to represent such agreement, i.e., small $\sum_{i=1}^{m}(S_{pq}-S_{pq}^{(i)})^2$ means $(\mathbf{x}_p, \mathbf{x}_q)$ is an easy pair. Based on this, we impose a weight matrix $\mathbf{W} \in [0,1]^{n\times n}$ on all the pairs, whose element W_{pq} indicates the weight of the pair $(\mathbf{x}_p, \mathbf{x}_q)$. The easy pair $(\mathbf{x}_p, \mathbf{x}_q)$ will have large W_{pq}. Following [39], we simply set $\Omega(\lambda, \mathbf{w})$ in (1) as $\Omega(\lambda, \mathbf{w}) = -\lambda\|\mathbf{W}\|_1$ and obtain the following problem:

$$\begin{aligned} \min_{\mathbf{S},\mathbf{W},\boldsymbol{\alpha}} \quad & \sum_{i=1}^{m} \alpha_i^2 \left\| \mathbf{W} \odot \left(\mathbf{S} - \mathbf{S}^{(i)}\right) \right\|_F^2 - \lambda\|\mathbf{W}\|_1 \\ \text{s.t.} \quad & \sum_{i=1}^{m} \alpha_i = 1 \quad \forall i : 0 \le \alpha_i \le 1 \\ & \mathbf{S} = \mathbf{S}^T \quad \forall p, q : 0 \le S_{pq} \le 1, \quad 0 \le W_{pq} \le 1 \\ & \forall (\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{M} : S_{pq} = 1 \quad \forall (\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{C} : S_{pq} = 0 \end{aligned} \tag{3}$$

where λ is an adaptive parameter and grows in the process of optimization, and $\odot$ is the Hadamard product, i.e., the elementwise production of two matrices.

After obtaining $\mathbf{S}$, conventional ensemble methods use some postprocessing methods such as spectral clustering to generate the final clustering result. In our method, we wish to obtain the final clustering result in an end-to-end way, i.e., we directly obtain the clustering result when learning $\mathbf{S}$. A natural way is to make sure that $\mathbf{S}$ contains just c connected components, and the two data in the pairs in $\mathcal{M}$ are in the same connected components and the two data in the pairs in $\mathcal{C}$ are in the different connective components. Then, we just need to put the instances in the same connected components into a cluster.

To achieve this, we first make sure that $\mathbf{S}$ has c connective components. We define the Laplacian matrix $\mathbf{L}_S$ of $\mathbf{S}$ as $\mathbf{L}_S = \mathbf{D} - \mathbf{S}$, where $\mathbf{D}$ is a diagonal matrix whose pth diagonal element is $D_{pp} = \sum_q S_{pq}$. If $\mathbf{S}$ is nonnegative and symmetric, then according to [55], the rank of the Laplacian matrix of a graph, which contains c connective components is $n-c$. Thus, we impose the constraint $\text{rank}(\mathbf{L}_S) = n - c$ on (3).

Then, we make sure that the connective components do not violate the constraints. It is easy to make sure that the must-link data are in the same connective component by taking the transitive closure operation. We will introduce this in more detail in Section III-C. However, if $(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{C}$, it is difficult to guarantee that $\mathbf{x}_p$ and $\mathbf{x}_q$ are in different connective components. Notice that, even though $S_{pq} = 0$, it is also possible that $\mathbf{x}_p$ and $\mathbf{x}_q$ are in the same connective component, because there may exist another instance $\mathbf{x}_r$ such that $S_{pr} > 0$ and $S_{rq} > 0$. Fortunately, Nie et al. [56]

proposed the following theorem which can address this problem.

Theorem 1: [56] Given a graph $\mathbf{S}$ and a cannot-link pair $(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{C}$ any vector $\mathbf{f} \in \mathbb{R}^n$ such that $f_p = 1$ and $f_q = -1$, if $\mathbf{f}^T \mathbf{L}_S \mathbf{f} = 0$, where $\mathbf{L}_S$ is the Laplacain matrix of $\mathbf{S}$, and then $\mathbf{x}_p$ and $\mathbf{x}_q$ are in two different connective components.

According to Theorem 1, we can define an auxiliary matrix $\mathbf{F} \in \mathbb{R}^{n \times C}$ where $C = |\mathcal{C}|$ is the number of constraints in $\mathcal{C}$, and for the ith constraint (name $(\mathbf{x}_{p_i}, \mathbf{x}_{q_i})$) in $\mathcal{C}$, $F_{p_i i} = 1$ and $F_{q_i i} = -1$. Then, we add the constraint $tr(\mathbf{F}^T \mathbf{L}_S \mathbf{F}) = 0$ on (3).

Since we wish $\mathbf{S}$ contains c connective components, many elements in $\mathbf{S}$ should be exactly zeros because the nonzero value in $\mathbf{S}$ means the corresponding two instances are connected with an edge. However, in practice, it often happens that $\mathbf{S}$ is not sparse, i.e., some elements in $\mathbf{S}$ are very small but not zeros due to the numerical computing. To address this issue, we add a sparse regularized term $\|\mathbf{S}\|_0$ to learn a clearer clustering structure. The final objective function is as follows:

$$\begin{aligned} \min_{\mathbf{S},\mathbf{W},\boldsymbol{\alpha},\mathbf{F}} & \sum_{i=1}^{m} \alpha_i^2 \left\| \mathbf{W} \odot \left(\mathbf{S} - \mathbf{S}^{(i)} \right) \right\|_F^2 - \lambda \|\mathbf{W}\|_1 + \gamma \|\mathbf{S}\|_0 \\ \text{s.t.} \quad & \sum_{i=1}^{m} \alpha_i = 1 \; \forall i : 0 \le \alpha_i \le 1, \; rank(\mathbf{L}_S) = n - c \\ & \mathbf{S} = \mathbf{S}^T \; \forall p, q : 0 \le S_{pq} \le 1, \; 0 \le W_{pq} \le 1 \\ & \forall (\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{M} : S_{pq} = 1 \quad \forall (\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{C} : S_{pq} = 0 \\ & \forall (\mathbf{x}_{p_i}, \mathbf{x}_{q_i}) \in \mathcal{C} : F_{p_i i} = 1, F_{q_i i} = -1, \; tr(\mathbf{F}^T \mathbf{L}_S \mathbf{F}) = 0 \end{aligned} \tag{4}$$

where γ is a hyperparameter to control the sparsity of $\mathbf{S}$. Note that by optimizing (4), we can directly obtain the c clusters without any uncertain discretization postprocessing like spectral clustering and k-means. Different from the traditional two-stage way, where the ensemble and postprocessing are separated, our one-stage way can make them be boosted by each other to achieve the optimal goal.

C. Optimization

Now, we introduce how to optimize the objective function (4). In our framework, since we need to actively select pairs to label, or equivalently speaking, to determine $\mathcal{M}$ and $\mathcal{C}$ in (4), we divide the whole optimization process into two iterative steps: S-step for selecting pairs to label and E-step for the ensemble. In the following, we will introduce these two steps, respectively.

1) S-Step: In the S-step, we wish to select the most uncertain or difficult pairs to label. Fortunately, in our self-paced learning framework, the pair weight matrix $\mathbf{W}$ exactly indicates the difficulty or uncertainty of all pairs. Therefore, in this step, we optimize $\mathbf{W}$ while fixing the other variables, and then select pairs for annotation according to $\mathbf{W}$.

When other variables are fixed, we get the subproblem w.r.t. $\mathbf{W}$ as follows:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \sum_{i=1}^{m} \alpha_i^2 \left\| \mathbf{W} \odot \left(\mathbf{S} - \mathbf{S}^{(i)} \right) \right\|_F^2 - \lambda \|\mathbf{W}\|_1 \\ \text{s.t.} \quad & \forall p, q : 0 \le W_{pq} \le 1. \end{aligned} \tag{5}$$

Equation (5) can be decoupled into $n \times n$ independent subproblems. Considering the (p, q)th element of $\mathbf{W}$, we have

$$\min_{0 \le W_{pq} \le 1} \quad A_{pq} W_{pq}^2 - \lambda W_{pq} \tag{6}$$

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

Setting the partial derivative of (6) w.r.t. W_{pq} to zero, we obtain that $W_{pq} = (\lambda / 2A_{pq})$. Since $A_{pq} \ge 0$, $W_{pq} \ge 0$. If $(\lambda / 2A_{pq}) > 1$, we find that in the range $[0, 1]$, $A_{pq} W_{pq}^2 - \lambda W_{pq}$ is a monotonically decreasing function, and thus, the minimum is obtained when $W_{pq} = 1$. So, its closed-form solution is

$$W_{pq} = \min\left(\frac{\lambda}{2A_{pq}}, 1 \right). \tag{7}$$

As introduced earlier, A_{pq} can be regarded as one evaluation of the difficulty of the pairs, i.e., easy pairs may have small A_{pq}. From (7), small A_{pq} leads to large W_{pq}, and thus, easy pairs (with large W_{pq}) will contribute much to the model. Moreover, W_{pq} monotonically increases with λ, which means with the growth of λ, more and more pairs will have large weight and, thus, will be involved in learning.

In conventional self-paced learning, they often pay more attention to the pairs with large W_{pq}. However, in our framework, we also consider the pairs with small weights. Note that the smaller W_{pq} is, the more difficult the pair $(\mathbf{x}_p, \mathbf{x}_q)$ is. So, given a batch size k, we can choose the k pairs corresponding to the smallest k unlabeled elements in $\mathbf{W}$ for annotation. Since $\mathbf{W}$ is symmetric, we only consider W_{pq} with $p > q$. For the pair $(\mathbf{x}_p, \mathbf{x}_q)$, if it is labeled to belong to the same cluster, we add it into the must-link set $\mathcal{M}$, or otherwise, we add it into the cannot-link set $\mathcal{C}$.

Note that if $n \gg m$, it often happens that many elements on $\mathbf{W}$ have the same value. To address this issue, we need to reorder the W_{pq}'s which have the same value. In our method, we consider the degree of the instances. As we know, in a graph, if an instance has a larger degree, this instance may be connected with more other instances, and thus, the constraints about this instance may be easier to be propagated on the graph. Therefore, for fully propagating the pairwise constraints, we select the pairs whose two instances have large degrees. More formally, we denote $d(\mathbf{x}_p) = \sum_{i=1}^{n} S_{pi}$ as the degree of the pth instance, and then for a pair $(\mathbf{x}_p, \mathbf{x}_q)$, we define the function $\phi(\mathbf{x}_p, \mathbf{x}_q)$ as the degree score of this pair

$$\phi(\mathbf{x}_p, \mathbf{x}_q) = d(\mathbf{x}_p) * d(\mathbf{x}_q) = \sum_{i,j=1}^{n} S_{pi} S_{qj}. \tag{8}$$

For the pairs who have the same values, we further reorder them as the descending order of $\phi(\cdot, \cdot)$.

After selecting the pairs for annotation, we will further expand the pairwise constraints to propagate the must-link and cannot-link constraints. Consider that the must-link relation should be an equivalence relation, i.e., $\mathcal{M}$ should satisfy the following properties.

Property 1: Reflexive—$(\mathbf{x}_p, \mathbf{x}_p) \in \mathcal{M}$.

Property 2: Symmetric—if $(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{M}$, then $(\mathbf{x}_q, \mathbf{x}_p) \in \mathcal{M}$.

Property 3: Transitive—if $(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{M}$ and $(\mathbf{x}_q, \mathbf{x}_r) \in \mathcal{M}$, then $(\mathbf{x}_p, \mathbf{x}_r) \in \mathcal{M}$.

Obviously, the cannot-link relation should satisfy the symmetric property.

Property 4: Symmetric—if $(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{C}$, then $(\mathbf{x}_q, \mathbf{x}_p) \in \mathcal{C}$.

In addition, the must-link and cannot-link relation also have a transitive property.

Property 5: Transitive—if $(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{M}$ and $(\mathbf{x}_q, \mathbf{x}_r) \in \mathcal{C}$, then $(\mathbf{x}_p, \mathbf{x}_r) \in \mathcal{C}$.

At last, $\mathcal{M}$ and $\mathcal{C}$ should be self-consistent, i.e., as shown in the following.

Property 6: Self-consistent: if $(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{M}$, then $(\mathbf{x}_p, \mathbf{x}_q) \notin \mathcal{C}$, and vice versa.

To satisfy the above-mentioned properties, after adding pairs into $\mathcal{M}$ and $\mathcal{C}$, we apply the transitive closure operator to expand $\mathcal{M}$ and $\mathcal{C}$ and learn the ensemble results in the expanded $\mathcal{M}$ and $\mathcal{C}$. In more detail, when expanding the must-link set, we first add all $(\mathbf{x}_p, \mathbf{x}_p)$ in $\mathcal{M}$ if they are not in $\mathcal{M}$ originally. Then, for any $(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{M}$, if $(\mathbf{x}_q, \mathbf{x}_p) \notin \mathcal{M}$, we add it into $\mathcal{M}$. At last, we obtain the transitive closure of $\mathcal{M}$ by a standard algorithm, such as the Warshall Algorithm [57].

For expanding $\mathcal{C}$ more easily, we first partite the whole instance set into several equivalence classes according to $\mathcal{M}$, such that if $\mathbf{x}_p$ and $\mathbf{x}_q$ has a must-link relation, then they are in the same equivalence class. If $\mathbf{x}_p$ does not have any must-link relation with other instances, then $\mathbf{x}_p$ itself forms an equivalence class. After the equivalence class partition, we expand $\mathcal{C}$. For any $(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{C}$, we add all pairs $(\mathbf{x}_r, \mathbf{x}_s)$, where $\mathbf{x}_r \sim \mathbf{x}_p$ and $\mathbf{x}_s \sim \mathbf{x}_q$, and $\sim$ denotes that the two instances are in the same equivalence class, into $\mathcal{C}$. At last, for any $(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{C}$, if $(\mathbf{x}_q, \mathbf{x}_p) \notin \mathcal{C}$, we add it into $\mathcal{C}$.

Algorithm 1 summarizes the process of expanding. According to the following theorem, the expanded $\mathcal{M}$ and $\mathcal{C}$ satisfy the Properties 1–6.

Algorithm 1 Expanding $\mathcal{M}$ and $\mathcal{C}$

Input: The initial must-link sets $\mathcal{M}$ and cannot-link set $\mathcal{C}$.
Output: The expanded $\mathcal{M}$ and $\mathcal{C}$

1: For any p, if $(\mathbf{x}_p, \mathbf{x}_p) \notin \mathcal{M}$, add it into $\mathcal{M}$.
2: For any p, q, if $(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{M}$ and $(\mathbf{x}_q, \mathbf{x}_p) \notin \mathcal{M}$, add $(\mathbf{x}_q, \mathbf{x}_p)$ into $\mathcal{M}$.
3: Use Warshall Algorithm [57] algorithm to expand $\mathcal{M}$.
4: Partite the instance set into several equivalence classes according to $\mathcal{M}$.
5: For any $(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{C}$, $\mathbf{x}_r \sim \mathbf{x}_p$, $\mathbf{x}_s \sim \mathbf{x}_q$ and $(\mathbf{x}_r, \mathbf{x}_s) \notin \mathcal{C}$, add $(\mathbf{x}_r, \mathbf{x}_s)$ into $\mathcal{C}$.
6: For any p, q, if $(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{C}$ and $(\mathbf{x}_q, \mathbf{x}_p) \notin \mathcal{C}$, add $(\mathbf{x}_q, \mathbf{x}_p)$ into $\mathcal{C}$.

Theorem 2: For any self-consistent initial must-link set $\mathcal{M}$ and cannot-link set $\mathcal{C}$, after expanding them by Algorithm 1, the expanded sets $\mathcal{M}$ and $\mathcal{C}$ satisfy Properties 1–6.

Proof: See Appendix A. □

2) E-Step: After the S-step, we do the ensemble in the E-step. In this step, we will optimize (4) while fixing $\mathbf{W}$, $\mathcal{M}$, and $\mathcal{C}$. Since (4) contains the rank function, which is difficult to be optimized, we first relax it. According to the Ky Fan theorem [58], to eliminate the rank constraint, we introduce an orthogonal auxiliary matrix $\mathbf{Y} \in \mathbb{R}^{n\times c}$, and rewrite it as

$$\begin{aligned}
\min_{\mathbf{S},\boldsymbol{\alpha},\mathbf{F},\mathbf{Y}} & \sum_{i=1}^{m} \alpha_i^2 \left\| \mathbf{W} \odot \left(\mathbf{S} - \mathbf{S}^{(i)}\right) \right\|_F^2 + \gamma \|\mathbf{S}\|_0 \\
& + 2\rho\left(tr\left(\mathbf{Y}^T \mathbf{L}_S \mathbf{Y}\right) + tr\left(\mathbf{F}^T \mathbf{L}_S \mathbf{F}\right)\right) \\
\text{s.t.} \quad & \sum_{i=1}^{m} \alpha_i = 1 \quad \forall i : 0 \le \alpha_i \le 1, \quad \mathbf{Y}^T\mathbf{Y} = \mathbf{I} \\
& \mathbf{S} = \mathbf{S}^T \quad \forall p, q : 0 \le S_{pq} \le 1 \\
& \forall (\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{M} : S_{pq} = 1 \quad \forall (\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{C} : S_{pq} = 0 \\
& \forall (\mathbf{x}_{p_i}, \mathbf{x}_{q_i}) \in \mathcal{C} : F_{p_i i} = 1, \quad F_{q_i i} = -1
\end{aligned} \tag{9}$$

where ρ is a large enough parameter to make sure that the rank of $\mathbf{L}_S$ is $n - c$. Since (9) involves multiple variables, we will use a block coordinate descent to optimize it.

Optimizing S. Although (9) contains complex constraints and ℓ_0-norm on $\mathbf{S}$, which is nonconvex and discontinuous, fortunately, according to the following theorem, it also has a closed-form solution.

Theorem 3: The (p, q)th element of $\mathbf{S}$ has the following closed-form solution:

$$S_{pq} = \begin{cases} 1, & \text{if } (\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{M} \\ 0, & \text{if } (\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{C} \\ 1, & \text{if } (\mathbf{x}_p, \mathbf{x}_q) \notin \mathcal{M} \cup \mathcal{C} \text{ and } B_{pq} \ge 1 \\ B_{pq}, & \text{if } (\mathbf{x}_p, \mathbf{x}_q) \notin \mathcal{M} \cup \mathcal{C} \text{ and } \tau_{pq} \le B_{pq} < 1 \\ 0, & \text{if } (\mathbf{x}_p, \mathbf{x}_q) \notin \mathcal{M} \cup \mathcal{C} \text{ and } B_{pq} < \tau_{pq}. \end{cases} \tag{10}$$

where $B_{pq} = \left(\sum_{i=1}^m \alpha_i^2 S_{pq}^{(i)} - (\rho(\|\mathbf{Y}_{p\cdot} - \mathbf{Y}_{q\cdot}\|_2^2 + \|\mathbf{F}_{p\cdot} - \mathbf{F}_{q\cdot}\|_2^2)/2W_{pq}^2\right)/\sum_{i=1}^m \alpha_i^2)$ and $\tau_{pq} = ((\gamma)^{1/2}/(\sum_{i=1}^m \alpha_i^2 W_{pq}^2)^{1/2})$.

Proof: See Appendix B. □

Optimizing F. When optimizing $\mathbf{F}$, we find that it can be decoupled into C independent subproblems according to the C constraints in $\mathcal{C}$. Considering the kth constraint $(\mathbf{x}_{p_k}, \mathbf{x}_{q_k})$ in $\mathcal{C}$, the corresponding subproblem is that

$$\min_{\mathbf{F}_{\cdot k}} \quad \mathbf{F}_{\cdot k}^T \mathbf{L}_S \mathbf{F}_{\cdot k}, \quad \text{s.t.} \quad F_{p_k k} = 1, \quad F_{q_k k} = -1. \tag{11}$$

It seems that (11) can be solved by the label propagation process [56], [59]. However, this method is inappropriate for our problem. Notice that, in the label propagation process methods, they need to construct a matrix $\mathbf{L}_{uu} \in \mathbb{R}^{(n-2)\times(n-2)}$, which is a submatrix of $\mathbf{L}_S$ by removing the p_kth and q_kth rows and columns of $\mathbf{L}_S$, and then they need to compute the inverse of $\mathbf{L}_{uu}$. However, in our problem, since the rank of $\mathbf{L}_S$ is c, which is often much smaller than n, $\mathbf{L}_{uu}$ is often noninvertible.

To address this issue, we propose a modified label propagation method by taking a closer look at the current graph $\mathbf{S}$. We consider two cases. The first case is that $\mathbf{x}_{p_k}$ and $\mathbf{x}_{q_k}$ are in different connective components in current $\mathbf{S}$. We find all instances in the connective component which $\mathbf{x}_{p_k}$ belongs to, put them in a set $\mathcal{P}$, and put all instances in the connective component of $\mathbf{x}_{q_k}$ into a set $\mathcal{Q}$. Then, for any $\mathbf{x}_p \in \mathcal{P}$, we set $F_{pk} = 1$ and for any $\mathbf{x}_q \in \mathcal{Q}$, we set $F_{qk} = -1$. For all remaining instances $\mathbf{x}_r$, we set $F_{rk} = 0$. It is easy to verify that $\mathbf{F}_{\cdot k}^T \mathbf{L}_S \mathbf{F}_{\cdot k} = 0$, and notice that since $\mathbf{L}_S$ is positive semidefinite, for any vector $\mathbf{x}$, we have $\mathbf{x}^T \mathbf{L}_S \mathbf{x} \ge 0$. Therefore, this $\mathbf{F}_{\cdot k}$ is the optima of this subproblem.

Now, consider the second case that $\mathbf{x}_{p_k}$ and $\mathbf{x}_{q_k}$ are in the same connective component in current $\mathbf{S}$. We put all instances in such connective component into a set $\mathcal{R} = \{\mathbf{x}_p, \mathbf{x}_q, \mathbf{x}_{k_1}, \ldots, \mathbf{x}_{k_{r-2}}\}$, where r is the number of instances in the set $\mathcal{R}$. Then, we extract the submatrix of $\mathbf{L}_S$ corresponding to the instances in $\mathcal{R}$, called $\mathbf{L}_R \in \mathbb{R}^{r\times r}$. We define a vector $\mathbf{f} = (\mathbf{f}_l, \mathbf{f}_u)^T \in \mathbb{R}^r$, where $\mathbf{f}_l = (1, -1)^T$. Then, we rearrange $\mathbf{L}_R$ as

$$\mathbf{L}_R = \begin{bmatrix} \mathbf{L}_R^{ll} \in \mathbb{R}^{2\times 2} & \mathbf{L}_R^{lu} \in \mathbb{R}^{2\times(r-2)} \\ \mathbf{L}_R^{luT} \in \mathbb{R}^{(r-2)\times 2} & \mathbf{L}_R^{uu} \in \mathbb{R}^{(r-2)\times(r-2)} \end{bmatrix}.$$

Then, we have

$$\mathcal{L} = \mathbf{f}^T \mathbf{L}_R \mathbf{f} = \mathbf{f}_l^T \mathbf{L}_R^{ll} \mathbf{f}_l + 2\mathbf{f}_l^T \mathbf{L}_R^{lu} \mathbf{f}_u + \mathbf{f}_u^T \mathbf{L}_R^{uu} \mathbf{f}_u. \tag{12}$$

By setting $(\partial\mathcal{L}/\partial\mathbf{f}_u) = 0$, we have

$$\mathbf{f}_u = -(\mathbf{L}_R^{uu})^{-1} \mathbf{L}_R^{luT} \mathbf{f}_l. \tag{13}$$

Notice that since $\mathbf{L}_R$ is a Laplacian matrix of a connective graph, $\text{rank}(\mathbf{L}_R) = r - 1$ and $\mathbf{L}_R^{uu}$ is invertible in most cases. Then, for all instances in $\mathcal{R}$, we set the corresponding elements in $\mathbf{F}_{.k}$ as the value in $\mathbf{f}$ and set all remaining elements in $\mathbf{F}_{.k}$ as 0. This is the optima of this subproblem.

Optimizing Y: When optimizing $\mathbf{Y}$, we have

$$\min_{\mathbf{Y}} \quad tr\left(\mathbf{Y}^T \mathbf{L}_S \mathbf{Y}\right), \quad \text{s.t.} \quad \mathbf{Y}^T \mathbf{Y} = \mathbf{I}. \tag{14}$$

The closed-form solution of (14) can be obtained by the Ky Fan theorem [58]. In more detail, the columns of $\mathbf{Y}$ are the c eigenvectors of $\mathbf{L}_S$ corresponding to its c smallest eigenvalues.

Optimizing $\boldsymbol{\alpha}$: When optimizing $\boldsymbol{\alpha}$, we have

$$\min_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{m} \alpha_i^2 \left\| \mathbf{W} \odot \left(\mathbf{S} - \mathbf{S}^{(i)}\right) \right\|_F^2$$
$$\text{s.t.} \quad 0 \le \alpha_i \le 1, \quad \sum_{i=1}^{m} \alpha_i = 1. \tag{15}$$

According to the Cauchy–Schwarz Inequality, we obtain its closed-form solution

$$\alpha_i = \frac{\left\| \mathbf{W} \odot \left(\mathbf{S} - \mathbf{S}^{(i)}\right) \right\|_F^{-2}}{\sum_{j=1}^{m} \left\| \mathbf{W} \odot \left(\mathbf{S} - \mathbf{S}^{(i)}\right) \right\|_F^{-2}}. \tag{16}$$

Note that $\left\| \mathbf{W} \odot \left(\mathbf{S} - \mathbf{S}^{(i)}\right) \right\|_F^2$ indicates the difference between the ith clustering result and the consensus result, which can be regarded as the quality of the ith clustering result. The smaller it is, the better the ith clustering result is. Since $\alpha_i \propto 1/\|\mathbf{W} \odot (\mathbf{S} - \mathbf{S}^{(i)})\|_F^2$, α_i indicates quality of the ith clustering result. If the ith clustering result is better, then its weight α_i should be larger and the ith clustering result plays a more important role in ensemble learning.

In this E-step, we can find the closed-form solution for each subproblem, which monotonically decreases the objective function. In addition, the objective function always has a lower bound, and thus, the E-step can always converge. In fact, it converges very fast in practice (often within 20 iterations in our experiments).

D. Discussion

In this section, we first discuss the initialization of the parameters and the selection strategy of the hyperparameter, and then we provide the algorithm and its time complexity.

We initialize $\mathbf{S} = (1/m)\sum_{i=1}^{m} \mathbf{S}^{(i)}$ and construct $\mathbf{L}_S = \mathbf{D} - \mathbf{S}$. Then, we obtain the initial $\mathbf{Y}$ by solving (14). We set $\rho = 1$ at first and adjust it automatically by observing the rank of $\mathbf{L}_S$. In more detail, if the rank of $\mathbf{L}_S$ is greater than $n - c$, i.e., the rank regularization is not strong enough, we update $\rho \leftarrow 2\rho$; if its rank is smaller than it, we update $\rho \leftarrow \rho/2$. We initialize $\alpha_i = 1/m$.

Now, we consider the initialization of the self-paced parameter λ. Since λ directly influences $\mathbf{W}$, we need to set λ based on $\mathbf{W}$. In (7), we have that if $\lambda > 2A_{pq}$, then $W_{pq} = 1$, which means we use the pair $(\mathbf{x}_p, \mathbf{x}_q)$ completely. For $(\mathbf{x}_p, \mathbf{x}_q)$, supposing m_{pq} base results agree that they should belong to a cluster and the remainder $m - m_{pq}$ results agree that they belong to different clusters, we can compute A_{pq} as follows:

$$A_{pq} = \sum_{i=1}^{m} \alpha_i^2 \left(S_{pq} - S_{pq}^{(i)}\right)^2 = \left(m_{pq}/m - \left(m_{pq}/m\right)^2\right)/m.$$

Define $\psi = m_{pq}/m$, which indicates the ratio of the results which agree that they belong to the same cluster. Therefore, when $\psi \ge 0.5$, larger ψ indicates the easier the pair.

Algorithm 2 SPACE Algorithm

Input: m base partition $\mathcal{C}^1, \ldots, \mathcal{C}^m$, number c of clusters, threshold θ, batch size k and the number of iterations T.
Output: Final c clusters.

```
Construct m base connective matrices S^1, ..., S^(m).
Initialize the parameters as introduced in Section III-D. Set M = ∅ and C = ∅.
for iter = 1, 2, ..., T do
    //S-step:
    Compute λ by Eq. (17), and then compute W by Eq. (7).
    Select the pairs corresponding to the k smallest elements in W to label and add them into M or C.
    Expand M and C by Algorithm 1.
    //E-step:
    while not converge do
        Compute S by Eq. (19).
        Compute F by the modified label propagation method.
        Compute Y by solving Eq. (14).
        Compute α by Eq. (16).
        Adjust ρ as introduced in Section III-D.
    end while
    Update ψ = max(r − δ, 0.5).
end for
Obtain the final clusters from the c connective component in S.
```

TABLE II
DESCRIPTION OF THE DATASETS

	#instances	#features	#classes	batch size
ALLAML	72	7129	2	10
AR	840	768	120	50
GLIOMA	50	4434	4	10
Lung	203	3312	5	50
Tr41	878	7454	10	50
Tdt	10212	36771	96	50
TOX	171	5748	4	50
WebACE	2340	1000	20	50

In SPACE, we initialize $\psi = 0.9$, and set λ as follows:

$$\lambda = 2(\psi - \psi^2)/m. \tag{17}$$

Taking it back to (7), we find that, for $\mathbf{x}_p$ and $\mathbf{x}_q$, if more than 90% base results reach a consensus, then $W_{pq} = 1$, which means the pair will be involved in learning completely. In the subsequent iterations, we increase λ by decreasing ψ from 0.9 to 0.5 with a step size $\delta = 0.1$.

Then, we discuss the selection of the hyperparameter γ. Since the range of γ is $[0, +\infty)$, it is difficult to directly set γ. To select an appropriate γ, we need to figure out how γ influences the sparsity of $\mathbf{S}$. As can be seen from (19), γ plays a role as a threshold to determine whether S_{pq} should be zero. More specifically, $S_{pq} = 0$ when

$$B_{pq} < \tau_{pq} = \sqrt{\frac{\gamma}{\sum_{i=1}^{m} \alpha_i^2 W_{pq}^2}} \approx \sqrt{\frac{\gamma}{\sum_{i=1}^{m} \alpha_i^2}}$$
$$\le \sqrt{\frac{m\gamma}{(\sum_{i=1}^{m} \alpha_i)^2}} = \sqrt{m\gamma}. \tag{18}$$

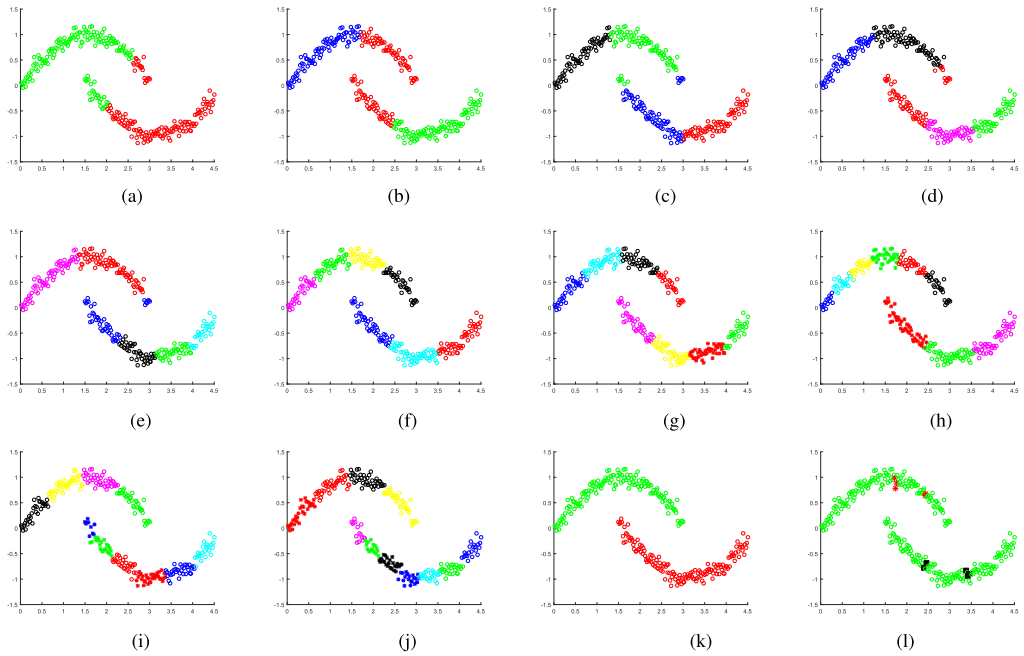

Fig. 3. Toy example results on the two-moon data. (a)–(j) Ten k-means base clustering results. (k) Clustering result of our SPACE method. (l) Selected data for annotation in the first two batches of SPACE.

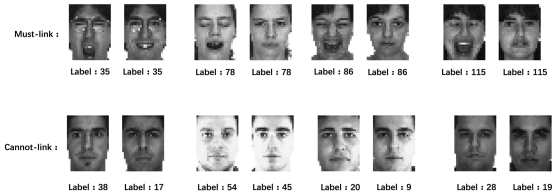

Fig. 4. Examples of selected pairs of SPACE in AR dataset. The first line shows four pairs with must-link and the second line shows four pairs with cannot-link.

The approximate equals sign is due to that at last almost all pairs are used for learning, and thus, all $W_{pq} \approx 1$. The no-greater-than sign is due to the Cauchy–Schwarz inequality. Denote $\theta = (m\gamma)^{1/2}$. θ can be regarded as a threshold, i.e., according to Theorem 3, S_{pq} is nonzero when $S_{pq} > \theta$. Therefore, to control the sparsity of $\mathbf{S}$, we do not need to directly set γ whose range is $[0, +\infty)$, but just set θ whose range is $[0, 1)$. Moreover, setting θ is more explainable. For instance, if we want to keep S_{pq} nonzero when $S_{pq} > 0.3$, we first set $\theta = 0.3$, and compute $\gamma = \theta^2/\mathrm{m} = 0.09/\mathrm{m}$.

Algorithm 2 summarizes the SPACE method. Since we need to store the m connective matrices, the space complexity is $O(mn^2)$. In the S-step, the time complexity of computing $\mathbf{W}$ is $O(mn^2)$ since we need to compute $\mathbf{A}$ first. The time complexity of the transitive closure operator is $O((kT)^3)$. Note that k is the batch size and often no more than 100 and T is the number of batches for annotation and at most several tens. So, $k, T \ll n$ in practice.

In E-step, computing $\mathbf{S}$ costs $O(n^2c + n^2m)$ according to Theorem 3. When optimizing $\mathbf{F}$, in the worst case, we need to calculate the k equation system of $r - 2$ variables, which costs $O((r-2)^2k)$ time. When learning $\mathbf{Y}$, we need to find the smallest c eigenvalues of $\mathbf{L}_S$, whose complexity is $O(n^2c)$. It takes $O(n^2m)$ to obtain $\boldsymbol{\alpha}$. Therefore, the time complexity is $O((n^2m + n^2c + (r-2)^2k)T_1T)$, where T is the number of outer iterations (lines 3–17), which is equal to the number of batches, and T_1 is the number of inner iterations (lines 9–15). The time complexity is comparable with the existing connective matrix-based methods [15], [53], [54]. Despite this, in the future, we will study how to further reduce the time complexity.

IV. Experiments

In this section, we conduct extensive experiments by comparing our SPACE with several state-of-the-art unsupervised and semisupervised consensus clustering methods on benchmark datasets.

A. Toy Example

Before comparing with state-of-the-art methods on benchmark datasets, we use a toy example to show the effectiveness of our method. Here, we use the two-moon data, as shown in Fig. 3. We run k-means with different numbers of clusters ten times and obtain the ten base results plotted in Fig. 3(a)–(j), respectively. As can be seen, k-means does not perform well on this nonlinear manifold data. Then, we apply SPACE to ensemble the ten base results, with batch size $k = 5$ and the number of batches $T = 10$. The result of our method is shown in Fig. 3(k). Although k-means often fails on nonlinear data, by ensembling multiple k-means results, our method can handle these nonlinear manifold data.

Moreover, although in the experiments we use ten batches, we find that after the second batch, our method can already discover the two-moon cluster structure. We show the selected data for annotation in Fig. 3(l). The red stars represent the data selected in the first batch and the black squares represent the ones selected in the second batch. We can see that these data are ones of the most ambiguous data because they are in the boundary of clusters in many base clustering results. After obtaining the relationship of these difficult data, our method can easily propagate this supervised information on all data and obtain accurate clustering results.

TABLE III
AVERAGE ACC AND STANDARD DEVIATION ON ALL DATASETS

Methods	ALLAML	AR	GLIOMA	Lung	Tr41	Tdt	Tox	WebACE
Base	0.6545 ±0.0644	0.2897 ±0.0113	0.4239 ±0.0347	0.7114 ±0.0944	0.3377 ±0.0733	0.4104 ±0.0188	0.4229 ±0.0322	0.3613 ±0.0366
Base-best	0.7292 ±0.0118	0.3100 ±0.0052	0.4880 ±0.0193	0.8675 ±0.0368	0.5034 ±0.0240	0.4460 ±0.0081	0.4825 ±0.0152	0.4192 ±0.0129
CSPA [10]	0.6583 ±0.0134	0.3310 ±0.0033	0.4100 ±0.0271	0.4138 ±0.0145	0.3151 ±0.0286	0.2850 ±0.0047	0.4246 ±0.0373	0.2773 ±0.0092
HGPA [10]	0.5444 ±0.0403	0.3314 ±0.0069	0.4180 ±0.0394	0.5025 ±0.0334	0.2421 ±0.0206	0.2959 ±0.0041	0.3854 ±0.0286	0.2593 ±0.0192
MCLA [10]	0.6722 ±0.0149	0.3336 ±0.0066	0.4000 ±0.0133	0.7084 ±0.0477	0.2530 ±0.0612	0.4000 ±0.0088	0.4152 ±0.0242	0.2809 ±0.0627
NMFC [52]	0.6722 ±0.0149	0.3315 ±0.0088	0.4140 ±0.0212	0.6764 ±0.1083	0.4198 ±0.0381	0.3716 ±0.0169	0.4269 ±0.0226	0.3458 ±0.0279
RCE [15]	0.6708 ±0.0161	0.3311 ±0.0098	0.4260 ±0.0097	0.7143 ±0.0710	0.3910 ±0.0757	-	0.4105 ±0.0264	0.3542 ±0.0219
MEC [54]	0.6056 ±0.0360	0.2795 ±0.0138	0.3940 ±0.0366	0.7379 ±0.1239	0.4690 ±0.0536	-	0.4304 ±0.0310	0.3762 ±0.0337
LWEA [67]	0.6736 ±0.0210	0.3130 ±0.0101	0.4320 ±0.0140	0.7458 ±0.0960	0.2919 ±0.0067	0.5744 ±0.0273	0.4234 ±0.0127	0.3319 ±0.0393
LWGP [67]	0.6750 ±0.0176	0.3320 ±0.0083	0.4320 ±0.0103	0.6498 ±0.0498	0.4464 ±0.0771	0.4288 ±0.0103	0.4193 ±0.0259	0.3456 ±0.0308
RSEC [68]	0.5917 ±0.0908	0.2862 ±0.0098	0.4180 ±0.0503	0.8217 ±0.0516	0.3779 ±0.0357	0.3029 ±0.0772	0.4041 ±0.0243	0.3098 ±0.0523
DREC [69]	0.6819 ±0.0249	0.3314 ±0.0085	0.4280 ±0.0103	0.6379 ±0.0724	0.4555 ±0.0346	0.3657 ±0.0036	0.4205 ±0.0408	0.3429 ±0.0262
SPCE [9]	0.6861 ±0.0238	0.3596 ±0.0067	0.4420 ±0.0199	0.9133 ±0.0053	0.3958 ±0.0767	0.6653 ±0.0741	0.4485 ±0.0185	**0.3934** ±0.0174
TRCE [70]	0.6917 ±0.0333	0.3531 ±0.0063	0.4400 ±0.0000	0.7901 ±0.0214	0.4698 ±0.0298	0.6273 ±0.0457	0.4491 ±0.0189	**0.3968** ±0.0170
CESHL [71]	0.6750 ±0.0134	0.3229 ±0.0101	0.4500 ±0.0170	0.8995 ±0.0256	0.2859 ±0.0034	0.4636 ±0.0821	0.4351 ±0.0275	0.3664 ±0.0374
E2CP [72]	0.7194 ±0.0909	0.3298 ±0.0117	0.4200 ±0.0094	0.8734 ±0.0461	0.4547 ±0.0575	0.5258 ±0.0233	0.4573 ±0.0328	**0.3930** ±0.0379
RSEMICE [7]	0.6972 ±0.0234	0.3363 ±0.0109	0.4600 ±0.0267	0.7483 ±0.0628	0.4490 ±0.0443	0.3892 ±0.0088	0.4684 ±0.0276	0.3456 ±0.0307
WECR [8]	0.6778 ±0.0171	0.3390 ±0.0042	0.4420 ±0.0148	0.7581 ±0.0418	0.4579 ±0.0330	0.4019 ±0.0143	0.4444 ±0.0490	0.3268 ±0.0379
SPACE-R	0.6806 ±0.0131	0.3268 ±0.0058	0.5340 ±0.0472	0.8557 ±0.0219	0.3103 ±0.0413	0.6225 ±0.0604	0.3117 ±0.0431	0.3498 ±0.0358
SPACE-R-P	0.7500 ±0.0940	0.3279 ±0.0052	0.5920 ±0.0598	0.9650 ±0.0188	0.3393 ±0.0563	0.6329 ±0.0691	0.6111 ±0.1510	0.3681 ±0.0239
SPACE	**0.8958** ±0.0639	**0.4057** ±0.0060	**0.6960** ±0.0853	**0.9906** ±0.0364	**0.4770** ±0.0370	**0.7029** ±0.0448	**0.7316** ±0.1568	**0.3979** ±0.0250

B. Real Datasets

We conduct experiments on benchmark datasets, including ALLAML [60], AR [61], Glioma [62], Lung [63], Tr41 [64], Tdt [65] TOX [62], and WebACE [66]. The details of these datasets are summarized in Table II.

C. Experimental Setup

Following the experimental setup of [15], to generate the base clusterings, we run k-means 200 times with different initializations to obtain 200 base results. Then, we divide the 200 base results into ten subsets, with 20 in each one. Next, we ensemble the 20 base results in each subset and report the average results over the ten subsets.

We compare SPACE with the following unsupervised clustering ensemble methods.

1) **Base:** It is the average result of all base clustering.
2) **Base-Best:** It is the best result of all base results.
3) **CSPA [10]:** It constructs a relationship between instances in the same cluster to obtain a measurement of pairwise similarity for the ensemble.

TABLE IV
AVERAGE NMI AND STANDARD DEVIATION ON ALL DATASETS

Methods	ALLAML	AR	GLIOMA	Lung	Tr41	Tdt	Tox	WebACE
Base	0.0882 ±0.0490	0.6483 ±0.0076	0.1629 ±0.0391	0.5284 ±0.0600	0.1066 ±0.0812	0.6111 ±0.0072	0.1374 ±0.0397	0.3567 ±0.0265
Base-best	0.1772 ±0.0547	0.6614 ±0.0042	0.2347 ±0.0227	0.6558 ±0.0620	0.2805 ±0.0392	0.6240 ±0.0055	0.2164 ±0.0269	0.4008 ±0.0127
CSPA [10]	0.0815 ±0.0137	0.6979 ±0.0022	0.1716 ±0.0281	0.3712 ±0.0194	0.2453 ±0.0329	0.5589 ±0.0028	0.1436 ±0.0446	0.3470 ±0.0164
HGPA [10]	0.0110 ±0.0141	0.6932 ±0.0041	0.1509 ±0.0362	0.3372 ±0.0481	0.1027 ±0.0165	0.5385 ±0.0088	0.1083 ±0.0211	0.2660 ±0.0219
MCLA [10]	0.0909 ±0.0117	0.6895 ±0.0067	0.1327 ±0.0291	0.5258 ±0.0166	0.0655 ±0.0529	0.6070 ±0.0044	0.1329 ±0.0165	0.1864 ±0.1616
NMFC [52]	0.0909 ±0.0117	0.6846 ±0.0037	0.1550 ±0.0270	0.5202 ±0.0656	0.2975 ±0.0370	0.5930 ±0.0042	0.1434 ±0.0286	0.3976 ±0.0223
RCE [15]	0.0899 ±0.0125	0.6755 ±0.0044	0.1624 ±0.0163	0.5248 ±0.0315	0.1865 ±0.0733	-	0.1344 ±0.0204	0.4009 ±0.0126
MEC [54]	0.0485 ±0.0429	0.6017 ±0.0174	0.1312 ±0.0433	0.5617 ±0.0868	0.3023 ±0.0357	-	0.1313 ±0.0308	0.3974 ±0.0298
LWEA [67]	0.0935 ±0.0192	0.6636 ±0.0055	0.1686 ±0.0207	0.5364 ±0.0624	0.0381 ±0.0102	0.7183 ±0.0091	0.1236 ±0.0289	0.3302 ±0.0089
LWGP [67]	0.0932 ±0.0142	0.6780 ±0.0060	0.1682 ±0.0177	0.4993 ±0.0230	0.2769 ±0.0783	0.6266 ±0.0053	0.1333 ±0.0280	0.3869 ±0.0137
RSEC [68]	0.0495 ±0.0491	0.5899 ±0.0148	0.1544 ±0.0455	0.6027 ±0.0621	0.1976 ±0.0487	0.4670 ±0.0342	0.1184 ±0.0137	0.3199 ±0.1095
DREC [69]	0.1006 ±0.0218	0.67672 ±0.0036	0.1641 ±0.0189	0.4647 ±0.0392	0.2785 ±0.0266	0.5985 ±0.0013	0.1394 ±0.0276	0.4284 ±0.0114
SPCE [9]	0.1237 ±0.0126	0.7399 ±0.0009	0.3010 ±0.0152	0.7415 ±0.0099	0.2338 ±0.0897	0.7125 ±0.0427	0.1961 ±0.0139	0.4075 ±0.0492
TRCE [70]	0.1150 ±0.0276	0.7390 ±0.0016	0.2289 ±0.0193	0.5130 ±0.0486	0.2513 ±0.0286	0.7275 ±0.0236	0.1541 ±0.0298	0.4060 ±0.0054
CESHL [71]	0.0929 ±0.0106	0.6157 ±0.0087	0.1939 ±0.0144	0.7133 ±0.0531	0.0175 ±0.0059	0.5325 ±0.1296	0.1445 ±0.0268	0.2702 ±0.0715
E2CP [72]	0.1719 ±0.0424	0.6763 ±0.0100	0.1551 ±0.0213	0.6569 ±0.0866	0.3336 ±0.0541	0.6624 ±0.0123	0.1893 ±0.0413	0.4025 ±0.0381
RSEMICE [7]	0.1294 ±0.0200	0.6743 ±0.0049	0.2338 ±0.0298	0.5521 ±0.0283	0.2710 ±0.0260	0.5929 ±0.0065	0.1801 ±0.0436	0.3682 ±0.0147
WECR [8]	0.0954 ±0.0101	0.7021 ±0.0018	0.1788 ±0.0157	0.5616 ±0.0250	0.2682 ±0.0271	0.5984 ±0.0054	0.1667 ±0.0566	0.3684 ±0.0167
SPACE-R	0.1150 ±0.0410	0.7350 ±0.0011	0.4393 ±0.0620	0.6174 ±0.0506	0.2434 ±0.0410	0.6424 ±0.0134	0.1948 ±0.0531	0.3627 ±0.0585
SPACE-R-P	0.2877 ±0.2059	0.7349 ±0.0013	0.4656 ±0.0693	0.8949 ±0.0592	0.3198 ±0.0359	0.6491 ±0.0286	0.4567 ±0.1441	0.4206 ±0.0297
SPACE	**0.6990** ±0.1704	**0.7579** ±0.0018	**0.5814** ±0.0505	**0.9740** ±0.0488	**0.3867** ±0.0362	**0.7429** ±0.0261	**0.6071** ±0.1707	**0.4642** ±0.0124

4) **HGPA [10]:** It ensembles base results with a constrained minimum cut objective.
5) **MCLA [10]:** It transforms the clustering ensemble problem into a cluster correspondence problem.
6) **NMFC [52]:** It applies the nonnegative matrix factorization to integrate base results.
7) **RCE [15]:** It minimizes the Kullback-Leibler (KL) divergence among each base result to learn a robust consensus result.
8) **MEC [54]:** It is a robust multiview consensus clustering method that uses low-rank and sparse decomposition to ensemble base results.
9) **LWEA [67]:** It applies a local weighting strategy to an agglomerative consensus clustering method.
10) **LWGP [67]:** It applies a local weighting strategy to a graph partition consensus clustering method.
11) **RSEC [68]:** It is a spectral-based robust consensus clustering method.
12) **DREC [69]:** It learns a dense embedding from base results for the ensemble.
13) **SPCE [9]:** It is a self-paced clustering ensemble method.
14) **TRCE [70]:** It is a trilevel robust clustering ensemble method.

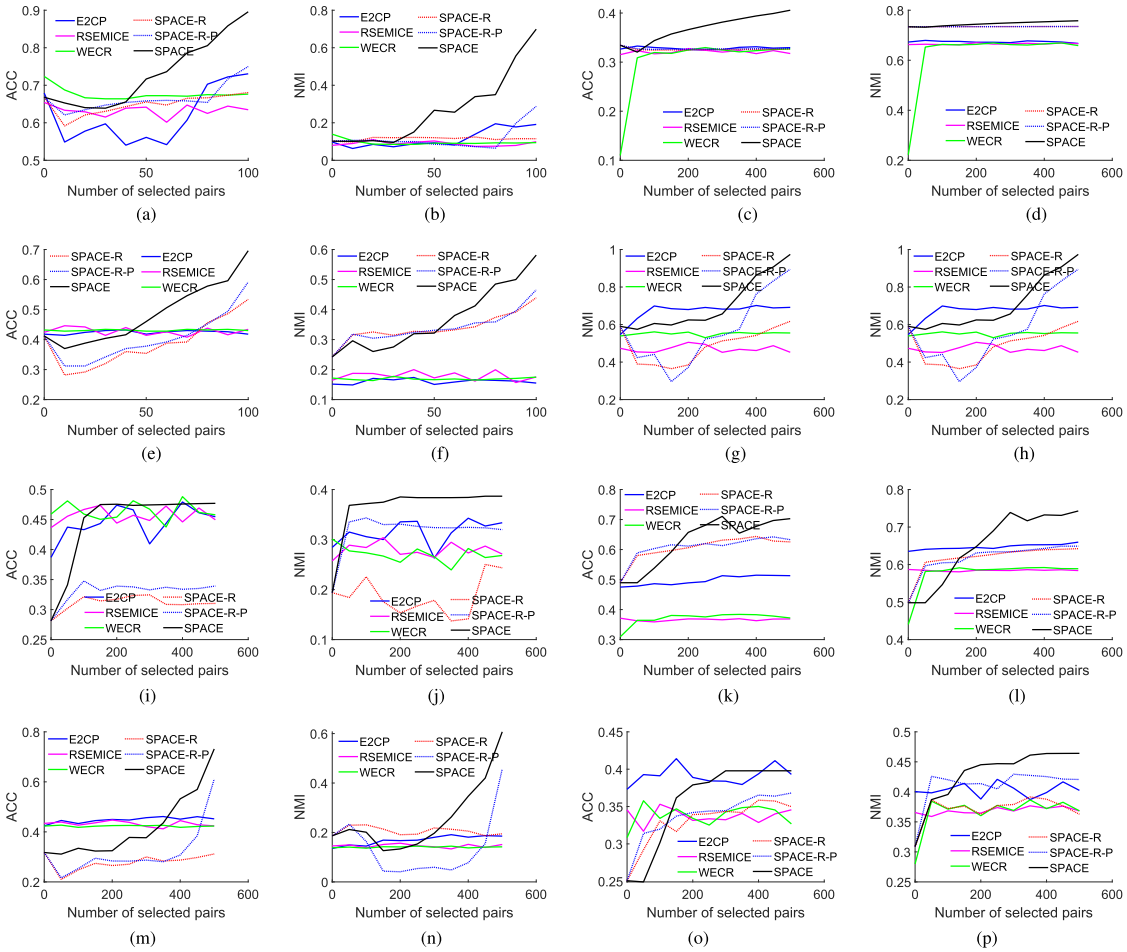

Fig. 5. Clustering results of semisupervised methods on different numbers of selected pairs. (a) ACC on ALLAML. (b) NMI on ALLAML. (c) ACC on AR. (d) NMI on AR. (e) ACC on Glioma. (f) NMI on Glioma. (g) ACC on Lung. (h) NMI on Lung. (i) ACC on Tr41. (j) NMI on Tr41. (k) ACC on Tdt. (l) NMI on Tdt. (m) ACC on TOX. (n) NMI on TOX. (o) ACC on WebACE. (p) NMI on WebACE.

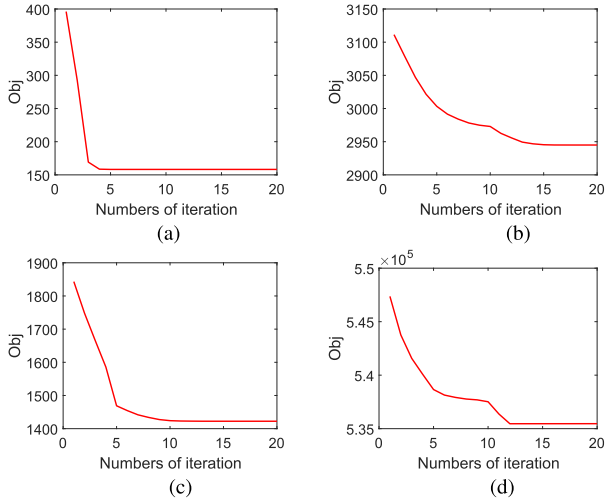

Fig. 6. Convergence curves on all datasets. (a) ALLAML. (b) AR. (c) TOX. (d) Tdt.

15) **CESHL [71]:** It is a clustering ensemble method with structured hypergraph learning.

Besides these, we also compare with the following semisupervised clustering ensemble methods.

1) **E2CP [72]:** It is a semisupervised clustering ensemble method to propagate the pairwise constraints.
2) **RSEMICE [7]:** It is a random subspace-based semisupervised clustering ensemble method.
3) **WECR [8]:** It is a semisupervised clustering ensemble method to learn the consensus result by assigning the weight to each cluster.
4) **SPACE-R:** It is our method without active learning, i.e., we randomly select pairwise constraints in S-step without propagation, and thus, SPACE degenerates to a semisupervised clustering ensemble method.
5) **SPACE-R-P:** It randomly selects the constraints and then expands the constraints with propagation. The difference between SPACE and SPACE-R-P lies in how they select the pairwise constraints.

In our method, the number of batches T is set to 10. Since ALLAML and Glioma are small-size datasets, on these data, we set the batch size $k = 10$, and on other datasets, we set the batch size $k = 50$. For all the compared semisupervised clustering ensemble methods, they use the same randomly selected $k \times T$ link constraints. Notice that the total number (i.e., $k \times T$) of constraints used in compared methods are the same as ours for a fair comparison. As discussed in Section III-D, we set $\gamma = \theta^2/20$, due to we ensemble 20 base results, and tune θ in $\{0, 0.1, \ldots, 0.9\}$. Following [73], we use accuracy (ACC) and normalized mutual information (NMI) to evaluate the clustering results.

D. Experimental Results

Tables III and IV show the average results and standard deviation of our method and other compared unsupervised

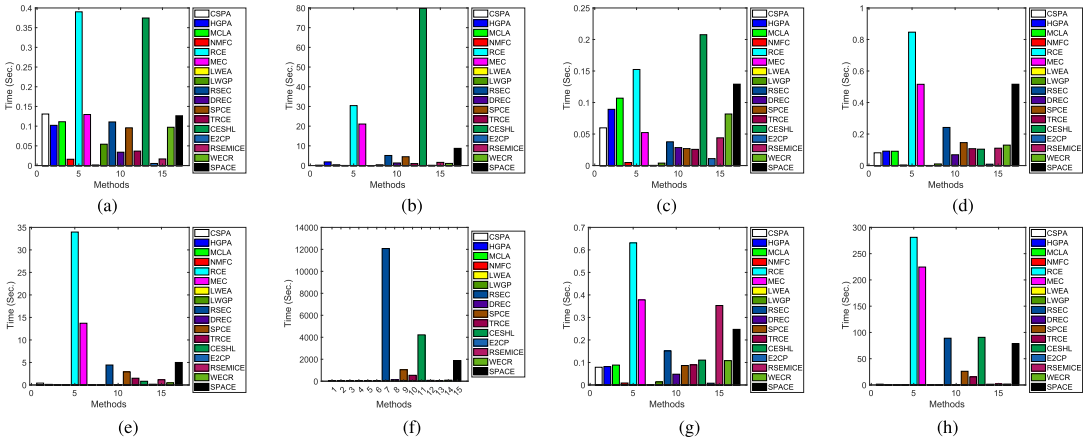

Fig. 7. Running time (seconds) of all methods. (a) ALLAML. (b) AR. (c) GLIOMA. (d) Lung. (e) Tr41. (f) Tdt. (g) TOX. (h) WebACE.

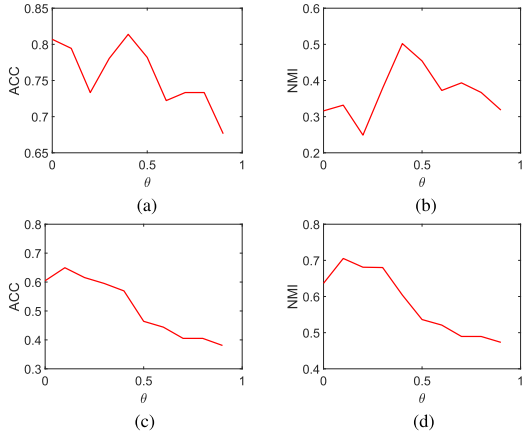

Fig. 8. Clustering results on different values of θ. (a) ACC on ALLAML. (b) NMI on ALLAML. (c) ACC on Tdt. (d) NMI on Tdt.

and semisupervised methods. From Tables III and IV, we can find that our proposed method can significantly outperform all compared unsupervised and semisupervised methods, which demonstrates the superiority of our self-paced active learning framework. Notice that, on the largest data Tdt, RCE and MEC suffer the out-of-memory error because of their high space complexity. Due to the carefully selected pairwise constraints, our method can even outperform the best base result (Base-best). Since our method is an active learning method, which uses supervised information (i.e., pairwise annotation), it can easily outperform the unsupervised methods. Although other semisupervised methods also use pairwise annotations, since they do not consider how to select the supervised information, their pairwise annotation is selected randomly. Different from these methods, our method actively selects informative or important pairs for annotation, which are more helpful for the clustering ensemble task. That is why our method can also outperform the semisupervised methods.

As an ablation study, we compare SPACE with SPACE-R and SPACE-R-P. It can be seen that SPACE-R-P outperforms SPACE-R on most datasets, which shows that the propagation operation is useful most time. Notice that on some datasets, e.g., AR and Tdt, the propagation seems to have no effect. It is because, these datasets contain too many clusters, and if we randomly select pairs, most pairs are cannot-link constraints and can hardly be propagated by Algorithm 1. SPACE often outperforms SPACE-R-P, which means the carefully selected annotations are much more useful than randomly selected annotations for the ensemble, which can demonstrate the motivation of active learning.

Fig. 4 shows some examples of selected data in the AR dataset. AR is a face image dataset, which contains the face images of 120 people, and each people has seven images. The first line in Fig. 4 shows some examples of must-link pairs selected by our algorithm, and the second line shows some examples of cannot-link pairs selected by SPACE. The label under each image is the ID of the people in the dataset. Although the two images of the selected must-link pairs belong to the same person, they have very different facial expressions, and thus, SPACE regards them as difficult data and selects them for the query. In the selected cannot-link pairs, the two persons in the pair are very alike, albeit different persons. The examples show that the selected pairs are indeed the difficult ones, which is consistent with our motivation.

To further demonstrate the effectiveness of the active learning process, we show the ACC and NMI with different numbers of selected pairs in our method compared with the semisupervised clustering ensemble methods in Fig. 5. From Fig. 5, we can find that our method outperforms other semisupervised methods at most times. Moreover, since our method integrates active selection into the ensemble learning, the selected constraints are suitable for our model, and thus, the performance of SPACE is improved very fast with the growth of the number of batches. For other compared methods, since they have no mechanism to select the informative constraints, their performance is improved more slowly than ours.

We conduct experiments to show the convergence curves of E-step in our method. We show the results on the ALLAML, AR, TOX, and Tdt datasets in Fig. 6. The results on other datasets are similar. From Fig. 6, we find that it converges very fast (often within 20 iterations), which demonstrates the claim in Section IV-C. We also show the running time of the proposed method and compared methods in Fig. 7. There are some methods faster than ours. The reasons are twofold. First, our method is a connective matrix based, or in other words, graph-based method, which needs to construct multiple connective matrices or multiple graphs and learn a consensus matrix for an ensemble, whose time complexity is square with the number of instances. Second, since our method is an active clustering method, we need some time for selecting key data for annotation. Despite this, compared with other connective matrix-based or graph-based methods, e.g., RCE, MEC, RSEC, and CESHL, our method is comparable with or even faster than theirs. Especially on the largest dataset Tdt, our method is nearly six times faster than RSEC.

E. Hyperparameter Study

The only hyperparameter needed to be tuned manually in our method is θ ($0 \leq \theta < 1$). Note that $\theta = 0$ means we remove the sparse regularized term $\|\mathbf{S}\|_0$. We tune θ in the range $\{0, 0.1, \ldots, 0.9\}$. The detailed results on the ALLAML and Tdt datasets can be found in Fig. 8. The results on other datasets are similar. Experimental results show that the performance of our method is stable across a wide range of parameters, and θ can be set in the range $[0.1, 0.4]$ to obtain a good performance. This is in line with intuition. Since θ is the truncation threshold that keeps S_{ij} nonzero when $S_{ij} > \theta$, if θ is too large, the graph will be too sparse, so that the number of the connective components may be far greater than c. Moreover, we can find that when $\theta = 0$ our method does not perform well, which also demonstrates the necessity of this regularized term.

V. Conclusion

In this article, we proposed a novel SPACE method, which jointly selected data to query for labeling and ensembled the clustering results. To select the data for annotation, we integrated the clustering ensemble into a self-paced framework, which can automatically find the difficult data for annotation and use the easy data for the ensemble. Then, we provided an iterative algorithm, i.e., jointly doing S-step and E-step, to optimize the introduced objective function. At last, extensive experimental results show that the proposed SPACE can outperform the state-of-the-art unsupervised and semisupervised clustering ensemble algorithms.

Despite the strengths of the proposed method, there still exist some limitations. The first one is the scalable issue. Since it needs to construct m $n \times n$ connective matrices, the time complexity is square with the number of instances. In the future, we will study how to further speed up the active clustering ensemble process. The second one is the effectiveness of the selection. Although the proposed method achieves good performance compared with other methods, it still has space to improve. The proposed one selects data for annotation by considering the uncertainty and transitivity, and there are other selection criteria, such as representativity and large margin, which may further improve the effectiveness of selection.

Appendix A: Proof of Theorem 2

Theorem 1: For any self-consistent initial must-link set $\mathcal{M}$ and cannot-link set $\mathcal{C}$, after expanding them by Algorithm 1, the expanded sets $\mathcal{M}$ and $\mathcal{C}$ satisfy Properties 1–6.

Proof: For Property 1, after line 1 of Algorithm 1, $\mathcal{M}$ satisfies the reflexive property. According to Theorem 1 in [57], after line 3 of Algorithm 1, $\mathcal{M}$ satisfies Property 3. Note that in the following steps, we do not add any pairs into $\mathcal{M}$, and therefore, $\mathbf{M}$ expanded by the whole Algorithm 1 satisfies Property 3.

Now, we show that after Line 3 $\mathcal{M}$ satisfies Property 2. Due to line 2 of Algorithm 1, for any $(\mathbf{x}_p, \mathbf{x}_q)$ in the initial $\mathcal{M}$, $(\mathbf{x}_q, \mathbf{x}_p)$ is also in $\mathcal{M}$. We just need to show that for any pair $(\mathbf{x}_p, \mathbf{x}_q)$ which is added into $\mathcal{M}$ by the Warshall algorithm, $(\mathbf{x}_q, \mathbf{x}_p)$ is also in $\mathcal{M}$. According to [57, Th. 1], if $(\mathbf{x}_p, \mathbf{x}_q)$ is in the expanded $\mathcal{M}$, then there exist $r_1, \ldots, r_l$, where $(\mathbf{x}_p, \mathbf{x}_{r_1}), (\mathbf{x}_{r_1}, \mathbf{x}_{r_2}), \ldots, (\mathbf{x}_{r_{l-1}}, \mathbf{x}_{r_l}), (\mathbf{x}_{r_l}, \mathbf{x}_q)$ are in the initial $\mathcal{M}$. After line 2, we have $(\mathbf{x}_q, \mathbf{x}_{r_l}), (\mathbf{x}_{r_l}, \mathbf{x}_{r_{l-1}}), \ldots, (\mathbf{x}_{r_2}, \mathbf{x}_{r_1}), (\mathbf{x}_{r_1}, \mathbf{x}_p)$ are in $\mathcal{M}$. Then, according to [57, Th, 1] again, we have $(\mathbf{x}_q, \mathbf{x}_p)$ is also in $\mathcal{M}$. Thus, Property 2 is satisfied.

Due to line 6 in Algorithm 1, $\mathcal{C}$ satisfies Property 4. After line 4, considering $(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{M}$ (i.e., $\mathbf{x}_p \sim \mathbf{x}_q$) and $(\mathbf{x}_q, \mathbf{x}_r) \in \mathcal{C}$, according to line 5, $(\mathbf{x}_p, \mathbf{x}_r) \in \mathcal{C}$, which means Property 5 is satisfied.

At last, we prove Property 6. Since the initial $\mathcal{M}$ and $\mathcal{C}$ are self-consistent, the instance set can be partitioned into several equivalence classes according to the initial $\mathcal{M}$, so that there are no cannot-links in any equivalence classes. Lines 1–3 preserve the Properties 1–3 of the equivalence class, and thus, in line 4, we can also successfully divide the instances into several self-consistent equivalence classes. Note that in lines 5 and 6, we do not add any cannot-link relationship into one equivalence class, i.e., we do not violate the self-consistent property in lines 5 and 6. Therefore, the expanded $\mathcal{M}$ and $\mathcal{C}$ also satisfy Property 6. □

Appendix B: Proof of Theorem 3

Theorem 2: The (p, q)th element of $\mathbf{S}$ has the following closed-form solution:

$$S_{pq} = \begin{cases} 1, & \text{if } (\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{M} \\ 0, & \text{if } (\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{C} \\ 1, & \text{if } (\mathbf{x}_p, \mathbf{x}_q) \notin \mathcal{M} \cup \mathcal{C} \text{ and } B_{pq} \geq 1 \\ B_{pq}, & \text{if } (\mathbf{x}_p, \mathbf{x}_q) \notin \mathcal{M} \cup \mathcal{C} \text{ and } \tau_{pq} \leq B_{pq} < 1 \\ 0, & \text{if } (\mathbf{x}_p, \mathbf{x}_q) \notin \mathcal{M} \cup \mathcal{C} \text{ and } B_{pq} < \tau_{pq}. \end{cases} \tag{19}$$

where $B_{pq} = (\sum_{i=1}^m \alpha_i^2 S_{pq}^{(i)} - (\rho(\|\mathbf{Y}_{p\cdot} - \mathbf{Y}_{q\cdot}\|_2^2 + \|\mathbf{F}_{p\cdot} - \mathbf{F}_{q\cdot}\|_2^2)/2W_{pq}^2)/\sum_{i=1}^m \alpha_i^2)$ and $\tau_{pq} = ((\gamma)^{1/2}/(\sum_{i=1}^m \alpha_i^2 W_{pq}^2)^{1/2})$.

Proof: We first drop the constraint $\mathbf{S} = \mathbf{S}^T$ for simplicity and then prove that the learned $\mathbf{S}$ satisfies the symmetric constraint. Note that $\mathbf{L}_S$ is relative with $\mathbf{S}$, and thus, we first handle the terms $2\rho tr(\mathbf{Y}^T\mathbf{L}_S\mathbf{Y})$ and $2\rho tr(\mathbf{F}^T\mathbf{L}_S\mathbf{F})$ as $2\rho tr(\mathbf{Y}^T\mathbf{L}_S\mathbf{Y}) = \rho\sum_{j,k=1}^n S_{jk}\|\mathbf{Y}_{j\cdot} - \mathbf{Y}_{k\cdot}\|_2^2$ and $2\rho tr(\mathbf{F}^T\mathbf{L}_S\mathbf{F}) = \rho\sum_{j,k=1}^n S_{jk}\|\mathbf{F}_{j\cdot} - \mathbf{F}_{k\cdot}\|_2^2$

Taking them back to the objective function, i.e., (9) in this article, we can decompose it into $n \times n$ independent subproblems. Since S_{pq} is definite when $(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{M} \cup \mathcal{C}$ according to the constraints, we just need to consider S_{pq} whose $(\mathbf{x}_p, \mathbf{x}_q)$ belongs to neither $\mathcal{M}$ nor $\mathcal{C}$. When handling the ℓ_0-norm, we introduce an auxiliary function $f(x)$, whose definition is $f(x) = 1$ if $x \neq 0$, and $f(x) = 0$ otherwise. Then, we can rewrite the objective function as follows:

$$\min_{0 \leq S_{pq} \leq 1} \left(S_{pq} - B_{pq}\right)^2 + \tau_{pq}^2 \, f(S_{pq}) \tag{20}$$

where $B_{pq} = (\sum_{i=1}^m \alpha_i^2 S_{pq}^{(i)} - (\rho(\|\mathbf{Y}_{p\cdot} - \mathbf{Y}_{q\cdot}\|_2^2 + \|\mathbf{F}_{p\cdot} - \mathbf{F}_{q\cdot}\|_2^2)/2W_{pq}^2)/\sum_{i=1}^m \alpha_i^2)$ and $\tau_{pq} = ((\gamma)^{1/2}/(\sum_{i=1}^m \alpha_i^2 W_{pq}^2)^{1/2})$.

The objective function of (20) is a quadratic function w.r.t. S_{pq}, and its closed-form solution is

$$S_{pq} = \begin{cases} 1, & \text{if } B_{pq} \geq 1 \\ B_{pq}, & \text{if } \tau_{pq} \leq B_{pq} < 1 \\ 0, & \text{if } B_{pq} < \tau_{pq}. \end{cases} \tag{21}$$

Taking all S_{pq} into consideration (including those in $\mathcal{M}$ and $\mathcal{C}$), we obtain (19).

To demonstrate the symmetry of $\mathbf{S}$, we use mathematical induction. In the first iteration, we initialize $\mathbf{S} = {}^1/_m \sum_{i=1}^m \mathbf{S}^{(i)}$. Since the input connective matrices are symmetric, $\mathbf{S}$ is also symmetric. In the following iterations, if $\mathbf{S}$ in the last iteration is symmetric, due to the solution of $\mathbf{W}$, i.e., (7) in this article, $\mathbf{W}$ is symmetric. Thus $\mathbf{B}$ and $\boldsymbol{\tau}$ are symmetric.

Moreover, according to Property 2 and Property 5, $\mathcal{M}$ and $\mathcal{C}$ are symmetric. To sum up, in each iteration, $\mathbf{S}$ computed by (19) satisfies the symmetric constraint. □

REFERENCES

[1] A. Topchy, A. K. Jain, and W. F. Punch, "A mixture model for clustering ensembles," in *Proc. SDM*, 2004, pp. 379–390.

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

[3] 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.

[4] F. Li, Y. Qian, J. Wang, C. Dang, and L. Jing, "Clustering ensemble based on sample's stability," *Artif. Intell.*, vol. 273, pp. 37–55, Aug. 2019.

[5] L. Bai, J. Liang, and F. Cao, "A multiple k-means clustering ensemble algorithm to find nonlinearly separable clusters," *Inf. Fusion*, vol. 61, pp. 36–47, Sep. 2020.

[6] Z. Yu et al., "Incremental semi-supervised clustering ensemble for high dimensional data clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 701–714, Mar. 2016.

[7] Z. Yu et al., "Adaptive ensembling of semi-supervised clustering solutions," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 8, pp. 1577–1590, Aug. 2017.

[8] Y. Lai, S. He, Z. Lin, F. Yang, Q.-F. Zhou, and X. Zhou, "An adaptive robust semi-supervised clustering framework using weighted consensus of random k-means ensemble," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 5, pp. 1877–1890, May 2021.

[9] P. Zhou, L. Du, X. Liu, Y.-D. Shen, M. Fan, and X. Li, "Self-paced clustering ensemble," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 4, pp. 1497–1511, Apr. 2021.

[10] 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.

[11] S. Vega-Pons, J. Correa-Morris, and J. Ruiz-Shulcloper, "Weighted partition consensus via kernels," *Pattern Recognit.*, vol. 43, no. 8, pp. 2712–2724, Aug. 2010.

[12] Z. Yu, H.-S. Wong, J. You, G. Yu, and G. Han, "Hybrid cluster ensemble framework based on the random combination of data transformation operators," *Pattern Recognit.*, vol. 45, no. 5, pp. 1826–1837, 2012.

[13] X. Liu et al., "Multiple kernel K-means with incomplete kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1191–1204, Feb. 2022.

[14] Z.-H. Zhou and W. Tang, "Clusterer ensemble," *Knowl.-Based Syst.*, vol. 19, no. 1, pp. 77–83, Mar. 2006.

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

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

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

[18] D. Huang, C.-D. Wang, J.-S. Wu, J.-H. Lai, and C.-K. Kwoh, "Ultra-scalable spectral clustering and ensemble clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1212–1226, Jun. 2020.

[19] 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, Aug. 2019.

[20] 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.

[21] P. Zhou, L. Du, and X. Li, "Self-paced consensus clustering with bipartite graph," in *Proc. 29th Int. Joint Conf. Artif. Intell. (IJCAI)*, C. Bessiere, Ed. Yokohama, Japan: Ijcai.org, 2020, pp. 2133–2139.

[22] P. Zhou, X. Liu, L. Du, and X. Li, "Self-paced adaptive bipartite graph learning for consensus clustering," *ACM Trans. Knowl. Discovery from Data*, vol. 17, no. 5, pp. 1–35, Oct. 2023.

[23] F. Yang, T. Li, Q. Zhou, and H. Xiao, "Cluster ensemble selection with constraints," *Neurocomputing*, vol. 235, pp. 59–70, Apr. 2017.

[24] Z. Yu et al., "Semi-supervised ensemble clustering based on selected constraint projection," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2394–2407, Dec. 2018.

[25] J. R. Barr, K. W. Bowyer, and P. J. Flynn, "Framework for active clustering with ensembles," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 11, pp. 1986–2001, Nov. 2014.

[26] Y. Shi, Z. Yu, W. Cao, C. L. P. Chen, H. S. Wong, and G. Han, "Fast and effective active clustering ensemble based on density peak," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3593–3607, Aug. 2020.

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

[28] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. ICML*, 2009, pp. 41–48.

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

[30] Y. Ren, X. Que, D. Tao, and Z. Xu, "Self-paced multi-task clustering," *Neurocomputing*, vol. 350, no. 1, pp. 212–220, Jul. 2019.

[31] 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.*, Aug. 2017, pp. 2641–2647.

[32] X. Guo et al., "Adaptive self-paced deep clustering with data augmentation," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 9, pp. 1680–1693, Sep. 2020.

[33] Z. Huang, Y. Ren, X. Pu, and L. He, "Non-linear fusion for self-paced multi-view clustering," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 3211–3219.

[34] 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. 1–7.

[35] Z. Huang, Y. Ren, X. Pu, L. Pan, D. Yao, and G. Yu, "Dual self-paced multi-view clustering," *Neural Netw.*, vol. 140, pp. 184–192, Aug. 2021.

[36] J. Shao, Z. Wu, Y. Luo, S. Huang, X. Pu, and Y. Ren, "Self-paced label distribution learning for in-the-wild facial expression recognition," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 161–169.

[37] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe, "Curriculum self-paced learning for cross-domain object detection," *Comput. Vis. Image Understand.*, vol. 204, Mar. 2021, Art. no. 103166.

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

[39] 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 2016.

[40] B. Settles, "Active learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep., 2009.

[41] P. Ren et al., "A survey of deep active learning," *ACM Comput. Surv.*, vol. 54, no. 9, p. 180, 2022.

[42] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Batch mode active learning and its application to medical image classification," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 417–424.

[43] R. Chattopadhyay, Z. Wang, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, "Batch mode active sampling based on marginal probability distribution matching," *ACM Trans. Knowl. Discovery Data*, vol. 7, no. 3, p. 13, 2013.

[44] H. Wang, L. Du, P. Zhou, L. Shi, and Y.-D. Shen, "Convex batch mode active sampling via α-relative Pearson divergence," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 1–7.

[45] H. Wang, L. Du, P. Zhou, L. Shi, Y. Qian, and Y.-D. Shen, "Experimental design with multiple kernels," in *Proc. IEEE Int. Conf. Data Mining*, Atlantic City, NJ, USA, Nov. 2015, pp. 419–428.

[46] Y.-F. Yan and S.-J. Huang, "Cost-effective active learning for hierarchical multi-label classification," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2962–2968.

[47] H. Wang, R. Zhou, and Y.-D. Shen, "Bounding uncertainty for active batch selection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5240–5247.

[48] B. Sun, P. Zhou, L. Du, and X. Li, "Active deep image clustering," *Knowl.-Based Syst.*, vol. 252, Sep. 2022, Art. no. 109346.

[49] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, "Multi-class active learning by uncertainty sampling with diversity maximization," *Int. J. Comput. Vis.*, vol. 113, no. 2, pp. 113–127, 2015.

[50] W. Liu, X. Chang, L. Chen, and Y. Yang, "Early active learning with pairwise constraint for person re-identification," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases (PKDD)* (Lecture Notes in Computer Science), vol. 10534. Skopje, Macedonia: Springer, Sep. 2017, pp. 103–118.

[51] W. Liu et al., "Pair-based uncertainty and diversity promoting early active learning for person re-identification," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 2, p. 21, 2020.

[52] T. Li and C. H. Q. Ding, "Weighted consensus clustering," in *Proc. SDM*, 2008, pp. 798–809.

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

[54] 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.
[55] 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*, Aug. 2014, pp. 977–986.
[56] F. Nie, H. Zhang, R. Wang, and X. Li, "Semi-supervised clustering via pairwise constrained optimal graph," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 3160–3166.
[57] S. Warshall, "A theorem on Boolean matrices," *J. ACM*, vol. 9, no. 1, pp. 11–12, Jan. 1962.
[58] H. Zha, X. He, C. Ding, M. Gu, and H. D. Simon, "Spectral relaxation for *k*-means clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 1057–1064.
[59] X. Zhu and Z. Ghahramain, "Learning from labeled and unlabeled data with label propagation," Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CALD-02-107, 2002.
[60] T. R. Golub et al., "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.
[61] H. Wang, F. Nie, and H. Huang, "Globally and locally consistent unsupervised projection," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1–6.
[62] J. Li et al., "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, p. 94, 2018.
[63] 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.
[64] 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, 2004.
[65] D. Cai, X. He, and J. Han, "Locally consistent concept factorization for document clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 6, pp. 902–913, Jun. 2011, doi: 10.1109/TKDE.2010.165.
[66] 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.
[67] 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.
[68] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "Robust spectral ensemble clustering via rank minimization," *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 1, pp. 1–25, Feb. 2019.
[69] J. Zhou, H. Zheng, and L. Pan, "Ensemble clustering based on dense representation," *Neurocomputing*, vol. 357, pp. 66–76, Sep. 2019.
[70] P. Zhou, L. Du, Y. Shen, and X. Li, "Tri-level robust clustering ensemble with multiple graph learning," in *Proc. 35th AAAI Conf. Artif. Intell. (AAAI)*, 2021, pp. 11125–11133.
[71] P. Zhou, X. Wang, L. Du, and X. Li, "Clustering ensemble via structured hypergraph learning," *Inf. Fusion*, vol. 78, pp. 171–179, Feb. 2022.
[72] Z. Lu and Y. Peng, "Exhaustive and efficient constraint propagation: A graph-based learning approach and its applications," *Int. J. Comput. Vis.*, vol. 103, no. 3, pp. 306–325, Jul. 2013.
[73] F. Nie, D. Xu, and X. Li, "Initialization independent clustering with actively self-training method," *IEEE Trans. Syst., Man, Cybern., B (Cybern.)*, vol. 42, no. 1, pp. 17–27, Feb. 2012.

Peng Zhou (Member, IEEE) 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, Chinese Academy of Sciences, Beijing, China, in 2017.

He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University, Hefei. He has authored or coauthored more than 30 papers in highly regarded conferences and journals, including the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TCYB), *Pattern Recognition*, *Information Fusion*, International Joint Conferences on Artificial Intelligence (IJCAI), AAAI Conference on Artificial Intelligence (AAAI), SIAM International Conference on Data Mining (SDM), and IEEE International Conference on Data Mining (ICDM). His research interests include machine learning, data mining, and artificial intelligence. More publications and codes can be found on his homepage: https://doctor-nobody.github.io/.

Bicheng Sun is currently pursuing the M.S. degree with the School of Computer Science and Technology, Anhui University, Hefei, China.

His research interests include machine learning and data mining.

Xinwang Liu (Senior Member, IEEE) received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China, in 2013.

He is currently a Professor with the School of Computer, NUDT. He has authored or coauthored more than 80 peer-reviewed papers, including those in highly regarded journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON MULTIMEDIA (TMM), IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (TIFS), International Conference on Machine Learning (ICML), Conference on Neural Information Processing Systems (NeurIPS), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE International Conference on Computer Vision (ICCV), AAAI Conference on Artificial Intelligence (AAAI), and International Joint Conferences on Artificial Intelligence (IJCAI). His current research interests include kernel learning, multiview clustering, and unsupervised feature learning.

Dr. Liu is an Associate Editor of the IEEE TNNLS and the *Information Fusion Journal*. More information can be found at https://xinwangliu.github.io/.

Liang Du 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.

From 2013 to 2014, he was a Software Engineer with Alibaba Group, Hangzhou, China. He was also an Assistant Researcher with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing. He is currently a Lecturer with Shanxi University, Taiyuan, China. He has authored or coauthored more than 40 papers in top conferences and journals, including ACM SIGKDD conference on Knowledge Discovery and Data mining (KDD), International Joint Conferences on Artificial Intelligence (IJCAI), AAAI Conference on Artificial Intelligence (AAAI), IEEE International Conference on Data Mining (ICDM), Transactions on Knowledge and Data Engineering (TKDE), SIAM International Conference on Data Mining (SDM), and International Conference on Information and Knowledge Management (CIKM). His research interests include clustering with noise and heterogeneous data, ranking for feature selection, active learning, and document summarization.

Xuejun Li (Member, IEEE) 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 research interests include workflow systems, cloud computing, and intelligent software.