Clustering Ensemble via Diffusion on Adaptive Multiplex

Peng Zhou, Boao Hu, Dengcheng Yan, and Liang Du

Abstract—Existing clustering ensemble methods often directly integrate multiple weak base results to obtain a consensus one which can improve the clustering performance. However, since the base results are weak and the clustering ensemble can improve the performance, why not refine the weak base results via the clustering ensemble, and then boost the clustering ensemble with the refined base results? To fulfill this idea, in this paper, we propose a novel clustering ensemble method with an adaptive multiplex. We first use the multiplex to represent the multiple weak base results. Then, we learn an updated representation by diffusing the representation on the multiplex with a manifold ranking model. Since the multiplex characterizes the structure information of all base results, the learned representation can ensemble such structure information during diffusion. Next, the multiplex is refined by such representation, which is a process of refining base results via ensemble. We iteratively learn the representation (i.e., do ensemble) and update the multiplex (i.e., do refinement), which can make the ensemble and refinement be boosted by each other. At last, the final consensus result is obtained from the refined multiplex. The extensive experiments demonstrate the effectiveness and superiority of the proposed framework. The codes of this paper are released in https://Doctor-Nobody.github.io/codes/CEAM.zip.

Index Terms—Clustering ensemble, multiplex, multiplex diffusion.

1 Introduction

Clustering ensemble has attracted much attention since it can provide a more stable and robust clustering result than the conventional single clustering methods [1], [2]. The setting of clustering ensemble is that it takes multiple weak base clustering results without any original features of data as inputs and applies consensus learning to aggregate the multiple weak base results to a consensus one which often achieves a better clustering performance. Since it does not access the original data, it can protect the privacy of data to some extent, which is also one advantage of the clustering ensemble.

Existing works of clustering ensemble often focus on how to learn a consensus result by integrating the information in all base results. For example, Huang et al. applied factor graph to aggregate base results [3]; Li et al. aligned the label information of base results based on Dempster-Shafer evidence theory to obtain the consensus result [4]; Zhou et al. constructed a dynamical hypergraph for the ensemble [5]. Among these methods, one kind of widely studied method is the multiple graph based method [6], [7], [8], [9], [10], [11]. These methods first construct multiple graphs from base results, where one graph corresponds to one base result. In the graph, each vertex represents an instance, and if two instances belong to the same cluster, there is an edge between them in the graph. Then these methods integrate these multiple graphs into a consensus graph and obtain the final clustering result from the consensus graph.

Although these multiple graph based methods have demonstrated promising performance, since the multiple graphs are constructed from weak base results directly and never updated, they are still unreliable and may deteriorate the ensemble learning. To see this, let us take a closer look at the base graphs. According to the construction of the base graphs, we find that each base graph has a special structure, i.e., it consists of multiple *cliques*, where each clique represents a cluster. Notice that a clique is a complete subgraph, where each instance is linked to all other instances in the same clique. For example, considering any two instances x_i and x_j in one clique, given any other instance x_k, according to the definition of the complete subgraph, either x_k links to both x_i and x_j or x_k links to neither of them. It means that x_i and x_j link to the same instances in the graph, and thus x_i and x_j are indistinguishable in the graph. Therefore, if we do not access the original features of instances, all instances in one clique are indistinguishable. However, in a

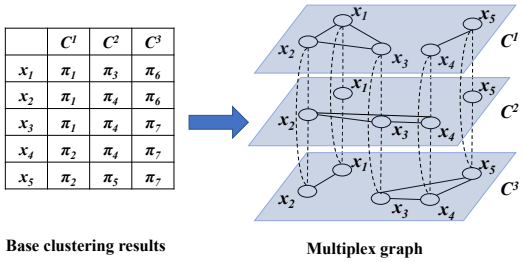


Fig. 1. A simple example of base clustering results and the corresponding multiplex graph. There are five instances $x_1, \cdots, x_5$, and we generate three base clustering results $\mathcal{C}^1, \mathcal{C}^2, \mathcal{C}^3$. The left side shows the three base clustering results. π_i is a cluster in a base clustering. For example, x_1, x_2, and x_3 belong to cluster π_1 in the first base clustering $\mathcal{C}^1$. The right side shows its multiplex graph, where each layer indicates a base clustering $\mathcal{C}^p$ $(p = 1, 2, 3)$.

Peng Zhou, Boao Hu, and Dengcheng Yan are with Anhui Provincial International Joint Research Center for Advanced Technology in Medical Imaging, the School of Computer Science and Technology, Anhui University, Hefei, 230601, China, Email: zhoupeng@ahu.edu.cn, e21201093@stu.ahu.edu.cn and yanzhou@ahu.edu.cn
Liang Du is with the School of Computer and Information Technology, Shanxi University, Taiyuan, 030006, China. Email: duliang@sxu.edu.cn

real cluster, different instances in a cluster should have different roles and should be treated differently. For example, some instances in the center of the cluster are more reliable and thus can be easily clustered; but some data which stay in the boundary of a cluster are unreliable and should be handled carefully because they may be mis-clustered. The base graphs constructed from the base results do not contain such information.

Therefore, if the ensemble can improve the clustering performance, *why not refine the base graphs via the ensemble and then apply the refined base graphs to boost the ensemble in turn?* To this end, in this paper, instead of focusing on how to ensemble the *fixed pre-defined* unreliable base graphs, we try to answer an alternative question which is how to refine the base graphs with the ensemble. Intuitively, given multiple base results, refining is possible because we can find which data in a base result are unreliable and can further correct them. For example, we generate 10 base results with a single clustering method. Considering data x_i and x_j, in the first base result, x_i and x_j belong to different clusters, i.e., there is no edge between x_i and x_j in the first graph. However, the other nine base results all agree that x_i and x_j should belong to the same cluster, i.e., there is an edge between x_i and x_j in the other nine graphs. In this case, with a high probability, we can say that the relationship of x_i and x_j in the first base result is unreliable, and we can correct it by adding a weighted edge between x_i and x_j in the first graph. That is a basic idea to refine the base results.

To fulfill this, we propose a novel Clustering Ensemble with Adaptive Multiplex (CEAM) method. The key idea is that we construct initial multiple base graphs as other clustering ensemble methods did, but before ensembling them directly, we refine them by fully considering the connection of data inter- and intra- multiple graphs. Notice that, given any two base graphs, their vertices have a one-to-one correspondence. Therefore, to better propagate information from one base graph to other base graphs, we apply the *multiplex* graph to represent the multiple base graphs. A multiplex graph[1] [12] is a multi-layer graph, where each layer is a traditional graph, and the vertex set of all layers is the same. The only inter-layer connections are the edges which link a vertex and all its counterpart vertices in all other layers. Figure 1 shows a simple example of 3 base clustering results and their corresponding multiplex graph. The left side of Figure 1 shows the 3 base clustering results (i.e., $\mathcal{C}^1, \cdots, \mathcal{C}^3$) containing 5 instances ($x_1, \cdots, x_5$). For example, in the first base result, instances x_1, x_2, and x_3 belong to cluster π_1, and the other two instances belong to cluster π_2. The right side shows the corresponding multiplex. Since there are 3 base results, the multiplex contains 3 layers where each layer is a base graph constructed from a base result. The connections of data intra-graph mean the edges in one layer of the multiplex, which are denoted by solid lines in Figure 1, and the connections of data inter-graph mean the edges link two instances which belong to different layers, which are denoted by the dotted lines in Figure 1. Therefore, multiplex can well characterize the connections of data inter- and intra- graphs. Moreover, the node set in different layers of the multiplex is all the same. x_i ($i = 1, 2, \cdots, 5$) in the p-th layer ($p = 1, 2, 3$) corresponds to x_i in the q-th layer ($q = 1, 2, 3$ and $q \neq p$) because there is an inter-layer edge between x_i in the p-th layer and x_i in the q-th layer. It characterizes the one-to-one correspondence in base clustering results.

As discussed before, since the base results are weak, the pre-defined multiplex is also unreliable and may mislead the clustering ensemble. To address this issue, in this paper, we refine the multiplex via ensemble. In more detail, after constructing the initial multiplex graph, we adaptively refine such multiplex by learning a representation of the data via ensemble. Since we do not access the original features of the data, we can only obtain an initial representation of each instance by its cluster indicator, which is also indistinguishable from the data in the same cluster. To tackle this problem, we refine such representations by diffusing them on the whole multiplex, which can ensemble the structure information under multiple base results. After that, we update the multiplex with the refined representation. We iteratively update the representation (i.e., do ensemble) and the multiplex (i.e., do refinement), so that they can be boosted by each other. At last, we obtain the final clustering result from the refined multiplex. Notice that the proposed mechanism can be plugged into any off-the-shelf multiple graph based clustering ensemble methods, such as [7], [10], [13]. In this paper, since our motivation is to show the importance of base refinement, we use the most straightforward and simplest way to do the ensemble, i.e., we directly average the multiple layers in the final multiplex and run spectral clustering on that averaged graph. The extensive experimental results show that, even though we use this most straightforward way to generate the consensus result, it can still outperform most state-of-the-art ensemble methods, which well demonstrates the effectiveness of our base refinement.

The main contributions of this paper are summarized as follows:

- We provide an alternative idea for clustering ensemble, i.e., apply the ensemble to refine the base results and in turn boost the ensemble by the refined base results.
- We propose a novel clustering ensemble method with an adaptive multiplex. To the best of our knowledge, this is the first work to do clustering ensemble on a dynamic multiplex.
- The extensive experiments show the effectiveness and superiority of the proposed method.

2 Related Work and Preliminaries

In this section, we introduce some related work of clustering ensemble and preliminaries of the multiplex graph.

2.1 Clustering Ensemble

Clustering ensemble is first proposed in [1] to integrate multiple weak base clustering results to obtain a strong consensus one. The benefits of clustering ensemble are mainly twofold. First, it can improve the robustness and stableness

1. In [12] and some other literature, it is often called a multiplex network. However, in the machine learning domain, to avoid confusion with the widely-known neural network, we call it a multiplex graph.

of the single clustering methods and thus further improve the clustering performance [2]. Second, since clustering ensemble does not access the original features of data, it can be easily used in many scenarios, such as the scenario of distributed data, and can also protect the privacy of data to some extent [14].

Conventional works often focus on proposing new ensemble methods. For example, some methods construct new representations for data based on the cluster indicators of each data and do the ensemble by the clustering on the categorical data [15], [16], [17]. Topchy et al. applied an expectation maximization clustering method on the categorical data for ensemble [15]; Bai et al. proposed an information theoretical based ensemble method [17].

Some methods applied label alignment methods for the ensemble. For example, Zhou et al. aligned the cluster indicators of multiple base kmeans results for ensemble [18]; Hore et al. proposed a scalable relabel method for the ensemble on big data [19].

Among these ensemble methods, one kind of the most popular methods is the multiple graph based method. These methods construct a graph for each base result and ensemble the base results by fusing the multiple graphs. For example, Iam-on et al. constructed a similarity metric based on the edges of graphs and did the ensemble with such similarity [13], [20]; Zhou et al. constructed multiple graphs from base results and proposed robust multiple graph learning methods to do the ensemble [10], [21], [22]; Zhou et al. constructed bipartite graphs from base results for ensemble and obtain the final consensus result by the partition of the bipartite graphs [23], [24]; Tao et al. proposed an adversarial learning method on the graphs for clustering ensemble [9]. Some methods integrate the multiple graphs to construct a co-association matrix and do the ensemble on that co-association matrix. For example, Tao et al. proposed a robust spectral clustering method on the co-association matrix [7], [8], [25]; Huang et al. designed a scalable spectral clustering ensemble method on the co-association matrix [26].

In this paper, we also focus on the multiple graph based clustering ensemble methods. Notice that all the above-mentioned methods, including the multiple graph based methods, focus on how to do the ensemble but ignore the base results refinement. Our method provides another way for improving the clustering performance, which means instead of proposing a sophisticated ensemble method, we refine the base results with the ensemble and can easily obtain a comparable or even better clustering result with even the simplest ensemble method.

2.2 Multiplex Graph

A multiplex graph is a kind of graph with multiple layers, whose definition is shown as follows:

Definition 1. *(Multiplex graph) [12] A multiplex graph or multiplex network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}_1, \cdots, \mathcal{E}_m, \hat{\mathcal{E}}\}$ is a multi-layer graph with m layers $\mathcal{G}_1, \cdots, \mathcal{G}_m$, where each layer is an undirected graph $\mathcal{G}_k = \{\mathcal{V}, \mathcal{E}_k\}$. $\mathcal{V} = \{x_1, \cdots, x_n\}$ is a vertex set which contains n vertices, and $\mathcal{E}_k$ is the intra-layer edge set of the k-th layer, where if x_i and x_j are linked in the k-th layer, $(x_i, x_j) \in \mathcal{E}_k$. Besides, for any vertex $x_i \in \mathcal{V}$, denoting x_i^p and x_i^q $(p \neq q)$ as the vertices of x_i in the p-th and q-th layer respectively, we have $(x_i^p, x_i^q) \in \hat{\mathcal{E}}$ where $\hat{\mathcal{E}}$ is the edge set which contains the inter-layer edges.*

Figure 1 shows an example of a multiplex graph, where $\mathcal{V} = \{x_1, \cdots, x_5\}$, and $\mathcal{E}_1, \cdots, \mathcal{E}_3$ contains all the edges denoted by the solid lines which are the intra-layer edges, and $\hat{\mathcal{E}}$ contains the edges denoted by the dashed lines which are the inter-layer edges. To show the dynamics of the multiplex, we introduce the projection graph of a multiplex:

Definition 2. *(Projection graph) [12] Given a multiplex $\mathcal{G} = \{\mathcal{V} = \{x_1, \cdots, x_n\}, \mathcal{E}_1, \cdots, \mathcal{E}_m, \hat{\mathcal{E}}\}$ which contains m layers where each layer contains n vertices, its projection graph is a graph $\tilde{\mathcal{G}} = \{\tilde{\mathcal{V}}, \tilde{\mathcal{E}}\}$. $\tilde{\mathcal{V}} = \{x_1^1, \cdots, x_n^1, x_1^2, \cdots, x_n^2, \cdots, x_1^m, \cdots, x_n^m\}$, where x_i^k denotes the vertex of x_i in the k-th layer. $\tilde{\mathcal{E}}$ contains all inter-layer and intra-layer edges in $\mathcal{G}$, i.e.,*

$$\tilde{\mathcal{E}} = (\cup_{k=1}^{m} \mathcal{E}_k) \cup \hat{\mathcal{E}}. \tag{1}$$

In this paper, we will use the weighted multiplex, and thus we need to construct the weighted adjacency matrix of the projection graph. Given the k-th layer $\mathcal{G}_k$ of a multiplex, we define $\mathbf{W}^{(k)} \in [0, 1]^{n \times n}$ as its adjacency matrix, i.e., if $(x_i, x_j) \in \mathcal{E}_k$, then the weight of this edge is $W_{ij}^{(k)}$; and if $(x_i, x_j) \notin \mathcal{E}_k$, the weight is 0. Then, we consider the weights of inter-layer edges. For simplicity, in this paper, we fix the weights of all inter-layer edges as 1, i.e., the weights of (x_i^p, x_i^q) $(p \neq q)$ are all 1's, and the weights of (x_i^p, x_j^q) $(i \neq j)$ are all 0's. To sum up, we can define the adjacency matrix of the projection graph of a multiplex $\mathbf{W} \in \mathbb{R}^{mn \times mn}$ as:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}^{(1)} & \mathbf{I}_n & \cdots & \mathbf{I}_n \\ \mathbf{I}_n & \mathbf{W}^{(2)} & \cdots & \mathbf{I}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_n & \mathbf{I}_n & \cdots & \mathbf{W}^{(m)} \end{bmatrix}, \tag{2}$$

where $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ is an Identity matrix, which characterizes the weights of the inter-layer edges. In this paper, we will adaptively refine this multiplex structure for the clustering ensemble task.

Multiplex has been applied in many tasks, such as community detection [27], [28], social network analysis [29], [30], and bioinformatics [31]. Among them, few works applied multiplex for clustering ensemble [32], [33]. For example, Rastin et al. applied the multiplex to clustering ensemble selection problem [32]. They used multiplex to select the informative base clusterings for the ensemble. Lyu et al. proposed an evolutionary clustering ensemble method with the multiplex graph and applied this method to the community detection task [33]. Notice that they both pre-defined a *fixed* multiplex and used some partition methods to obtain the final results on the fixed multiplex. As discussed before, since the base results are unreliable, the multiplex constructed from these unreliable base results is also unreliable. In this paper, we adaptively update the multiplex with the ensemble to address this problem.

3 Clustering Ensemble with Adaptive Multiplex

In this section, we will introduce our CEAM in more detail. First, we introduce some notations. We use the bold uppercase letter to denote the matrix and the bold lowercase letter to denote the vector. Given a matrix $\mathbf{M}$, we denote $\mathbf{M}_{i.}$ and $\mathbf{M}_{.i}$ as the i-th row and i-th column of matrix $\mathbf{M}$, respectively. M_{ij} denotes the (i,j)-th element in matrix $\mathbf{M}$.

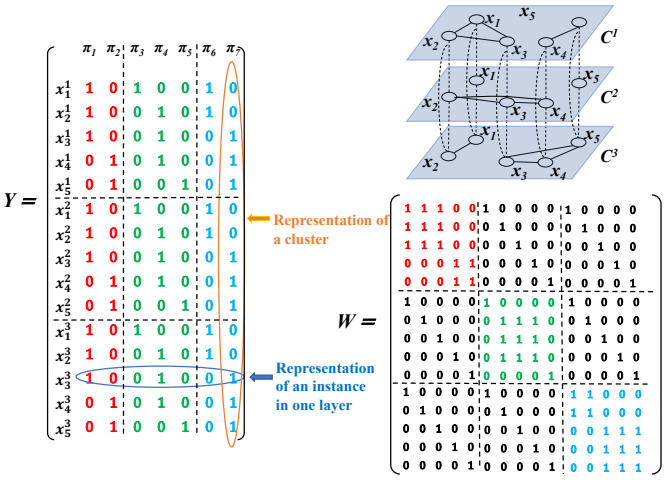


Fig. 2. An example of the representation matrix $\mathbf{Y}$ and the adjacency matrix of the multiplex $\mathbf{W}$. The three base clustering results are denoted by red, green, and blue font, respectively. In the representation matrix $\mathbf{Y}$, each row denotes a representation of an instance in one layer and each column denotes a representation of a cluster.

3.1 Initial Representation and Multiplex Construction

Given m base clustering results $\mathcal{C}^1, \cdots, \mathcal{C}^m$ with n instances, clustering ensemble wishes to learn a consensus clustering result $\mathcal{C}^*$ from $\mathcal{C}^1, \cdots, \mathcal{C}^m$ without accessing the original features of data. To achieve this, we first construct an initial representation of each data and an initial multiplex. Given the p-th base result $\mathcal{C}^p$, we first construct the p-th representation matrix $\mathbf{Y}^{(p)} \in \{0,1\}^{n \times c_p}$, where c_p is the number of clusters in the p-th base result. If x_i belongs to the q-th cluster in the p-th result, then $Y^{(p)}_{iq} = 1$ and otherwise $Y^{(p)}_{iq} = 0$. Then, denote $c = \sum_{k=1}^{m} c_k$, which is the total number of clusters in all base results. We concatenate $\mathbf{Y}^{(p)}$'s to obtain the representation of all instances as $\mathbf{Y}' = [\mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \cdots, \mathbf{Y}^{(m)}] \in \mathbb{R}^{n \times c}$.

Then, for the k-th base result, we construct an initial base graph $\mathcal{G}^{(k)}$ whose adjacency matrix is $\mathbf{W}^{(k)} = \mathbf{Y}^{(k)}\mathbf{Y}^{(k)T}$. We stack $\mathcal{G}^{(1)}, \cdots, \mathcal{G}^{(m)}$ into an initial multiplex $\mathcal{G}$ whose projection graph has the adjacency matrix with $\mathbf{W} \in \mathbb{R}^{mn \times mn}$ which is defined as Eq.(2). Notice that the multiplex $\mathcal{G}$ has m layers and the k-th layer represents the base graph constructed by the k-th base result $\mathbf{Y}^{(k)}$.

Since the projection matrix of the multiplex $\mathcal{G}$ contains $m \times n$ vertices, we need to extend the instance set to match the vertices in the graph. We define the extended instance set as $\tilde{\mathcal{X}} = \{x_1^1, \cdots, x_n^1, x_1^2, \cdots, x_n^2, \cdots, x_1^m, \cdots, x_n^m\}$, where x_i^p denotes the instance x_i in the p-th base result. Since $x_i^1, x_i^2, \cdots, x_i^m$ represent exactly the same instance (i.e., x_i) in the base results, there are inter-layer edges between any two instances x_i^p and x_i^q in the multiplex. We can similarly extend $\mathbf{Y}'$ by replicating it m times to obtain the representation matrix $\mathbf{Y} \in \{0,1\}^{mn \times c}$ of the extended instance set $\tilde{\mathcal{X}}$:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}^{(1)} & \mathbf{Y}^{(2)} & \cdots & \mathbf{Y}^{(m)} \\ \mathbf{Y}^{(1)} & \mathbf{Y}^{(2)} & \cdots & \mathbf{Y}^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Y}^{(1)} & \mathbf{Y}^{(2)} & \cdots & \mathbf{Y}^{(m)} \end{bmatrix}. \tag{3}$$

Each column of matrix $\mathbf{Y}$ is the representation of a cluster in all base results w.r.t. all extended instances, and each row of matrix $\mathbf{Y}$ is the representation of an instance in the extended set $\tilde{\mathcal{X}}$ w.r.t. all clusters. Figure 2 gives an example of an initial multiplex and the initial extended representation of the 3 base clustering results shown in Figure 1.

Notice that, in the initial multiplex $\mathcal{G}$, since each cluster in a base result forms a complete subgraph, two instances in the same cluster in a base result are indistinguishable. To address this issue, we need to adaptively update $\mathcal{G}$ and $\mathbf{Y}$ to enrich their information. We denote $\mathbf{F} \in \mathbb{R}^{mn \times c}$ and $\mathcal{G}'$ as the representation matrix and the multiplex we wish to learn, respectively. We initialize $\mathcal{G}' = \mathcal{G}$ and $\mathbf{F} = \mathbf{Y}$ first and update $\mathbf{F}$ and $\mathcal{G}'$ jointly. Therefore, our schema is an iterative framework. Figure 3 shows an example of the process in one iteration to handle the first base result. Suppose we need T epochs for updating. In the t-th epoch, we update the representation and the base graph corresponding to each base result in turn, i.e., we update $\mathbf{F}_{.1}, \cdots, \mathbf{F}_{.c_1}$ and $\mathcal{G}'^{(1)}$ first (whose process is illustrated in Figure 3) which are in the first base result $\mathcal{C}^1$, and then update $\mathbf{F}_{.(c_1+1)}, \cdots, \mathbf{F}_{.(c_1+c_2)}$ and $\mathcal{G}'^{(2)}$ which are in the second base result $\mathcal{C}^2$, and so on. At last, we update $\mathbf{F}_{.(c-c_m+1)}, \cdots, \mathbf{F}_{.c}$ and $\mathcal{G}'^{(m)}$ which are in the last base result $\mathcal{C}^m$. Next, we will introduce how to update the representation and multiplex in more detail.

3.2 Update Representation with Multiplex

Now, we consider how to update the representation in the p-th base result $\mathcal{C}^p$, whose procedure is shown in the blue dashed box in Figure 3. As introduced before, we update the representation $\mathbf{F}$ column by column. Take the s-th cluster in the p-th base result as an example. It corresponds to the $(r = \sum_{q=1}^{p-1} c_q + s)$-th column of $\mathbf{F}$. So, here we consider how to update $\mathbf{F}_{.r}$.

$\mathbf{F}_{.r}$ is the representation of the r-th cluster. F_{ir} can be seen as a score of x_i w.r.t. the r-th cluster, which is shown as the number in each vertex in Figure 3. The larger F_{ir} is, the more likely it is that x_i belongs to the r-th cluster. Therefore, we can update $\mathbf{F}_{.r}$ via a diffusion model on the multiplex $\mathcal{G}'$, which means that we set $\mathbf{Y}_{.r}$ as the initial scores of each vertex and propagate them through the inter-layer and intra-layer edges in the multiplex $\mathcal{G}'$ to obtain $\mathbf{F}_{.r}$. Due to the inter-layer edges, the information can be propagated among different base clusterings. Take Figure 3 as an example. In the first base result, x_4^1 is not in the cluster π_1 and thus its initial score w.r.t. π_1 is 0. However, in the second base result, x_4^2 and x_2^2 are in the same cluster, and there exists an inter-layer edge between x_2^2 and x_2^1, which is in π_1, and another inter-layer edge between x_4^1 and x_4^2.

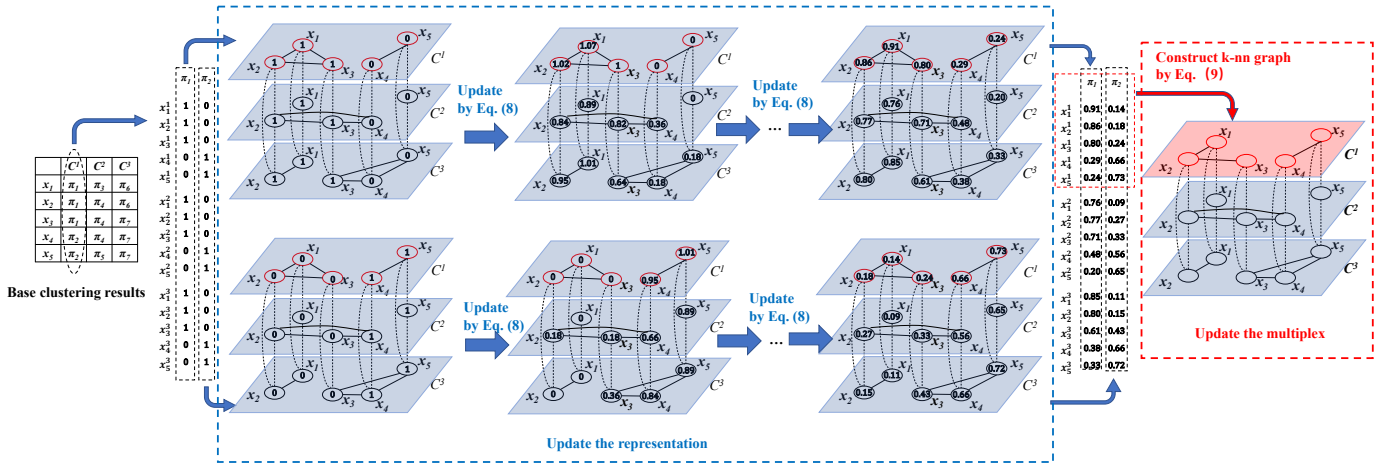


Fig. 3. An example of updating the representation and multiplex in one iteration, i.e., it handles the clusters π_1 and π_2 of the first base clustering. The number in each vertex is the value of F_{ir} which can be seen as the score of the i-th vertex w.r.t. the r-th cluster π_r. The blue dotted box shows the process of updating the representation and the red dotted box shows the process of updating the multiplex.

Then, by some steps of diffusions among these inter-layer and intra-layer edges, x_4^1 will obtain a positive score w.r.t. π_1. Since the diffusion process is on the whole multiplex, it can be seen as an ensemble of the structure information of multiple base results.

In this paper, we apply the manifold ranking [34] as the diffusion model. Manifold ranking first assigns an initial score to each vertex, and then propagates the scores in the vertices through the inter-layer and intra-layer edges to make the learned score preserve the manifold structure. More formally, suppose $\mathbf{W}$ is the adjacency matrix of the current projection graph of $\mathcal{G}'$, which is defined as Eq.(2). Manifold ranking wishes to learn the new representation $\mathbf{F}_{\cdot r}$ by optimizing the following objective function:

$$\min_{\mathbf{F}_{\cdot r}} \alpha \sum_{i,j=1}^{mn} \frac{W_{ij}}{2} \left(\frac{F_{ir}}{\sqrt{d_i}} - \frac{F_{jr}}{\sqrt{d_j}} \right)^2 + (1-\alpha)\|\mathbf{F}_{\cdot r} - \mathbf{Y}_{\cdot r}\|_2^2, \tag{4}$$

where d_i is the sum of the i-th row of $\mathbf{W}$, and $0 < \alpha < 1$ is a hyper-parameter. The second term is a fitting term, which makes the learned $\mathbf{F}_{\cdot r}$ not far away from its prior $\mathbf{Y}_{\cdot r}$. The first term is the propagation term, which makes that if x_i and x_j are closed in the multiplex, their representation F_{ir} and F_{jr} should also be closed.

Eq.(4) can be rewritten as the following form:

$$\min_{\mathbf{F}_{\cdot r}} \alpha \mathbf{F}_{\cdot r}^T (\mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2})\mathbf{F}_{\cdot r} + (1-\alpha)\|\mathbf{F}_{\cdot r} - \mathbf{Y}_{\cdot r}\|_2^2,$$

where $\mathbf{D}$ is a diagonal matrix, whose diagonal element is $D_{ii} = \sum_{j=1}^{mn} W_{ij}$. It has a closed-form solution:

$$\mathbf{F}_{\cdot r} = (1-\alpha)(\mathbf{I} - \alpha\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2})^{-1}\mathbf{Y}_{\cdot r} \tag{5}$$

However, Eq.(5) needs to compute the inversion of an mn-by-mn matrix, whose time complexity is $O(m^3n^3)$, which is untractable in practice. To address this issue, Zhou et al. [34] provides an iterative method to compute $\mathbf{F}_{\cdot r}$. We first initialize $\mathbf{F}_{\cdot r}$ as the same as the last epoch, and if it is the first epoch, we initialize $\mathbf{F}_{\cdot r} = \mathbf{Y}_{\cdot r}$. Then, we update it by:

$$\mathbf{F}_{\cdot r} = \alpha\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}\mathbf{F}_{\cdot r} + (1-\alpha)\mathbf{Y}_{\cdot r}. \tag{6}$$

According to [34], updating $\mathbf{F}_{\cdot r}$ by Eq.(6) will converge to the closed-form solution in Eq.(5). Although $\mathbf{W}$ is still an mn-by-mn matrix, since it is an adjacency matrix of a multiplex, which has a special structure as Eq.(2), we can further speed up the matrix multiplication. To see it, we denote $\mathbf{D}^{(k)}$ as a diagonal matrix whose diagonal element $D_{ii}^{(k)} = (\sum_{j=1}^{n} W_{ij}^{(k)} + m - 1)^{-1/2}$. It is easy to verify $\mathbf{D}^{-1/2}$ consists of $\mathbf{D}^{(1)}, \cdots, \mathbf{D}^{(m)}$. Then, we have

$$\begin{aligned}
&\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} \qquad (7)\\
&= \begin{bmatrix} \mathbf{D}^{(1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^{(2)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D}^{(m)} \end{bmatrix} \begin{bmatrix} \mathbf{W}^{(1)} & \mathbf{I}_n & \cdots & \mathbf{I}_n \\ \mathbf{I}_n & \mathbf{W}^{(2)} & \cdots & \mathbf{I}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_n & \mathbf{I}_n & \cdots & \mathbf{W}^{(m)} \end{bmatrix} \\
&\quad * \begin{bmatrix} \mathbf{D}^{(1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^{(2)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D}^{(m)} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{D}^{(1)}\mathbf{W}^{(1)}\mathbf{D}^{(1)} & \mathbf{D}^{(1)}\mathbf{D}^{(2)} & \cdots & \mathbf{D}^{(1)}\mathbf{D}^{(m)} \\ \mathbf{D}^{(2)}\mathbf{D}^{(1)} & \mathbf{D}^{(2)}\mathbf{W}^{(2)}\mathbf{D}^{(2)} & \cdots & \mathbf{D}^{(2)}\mathbf{D}^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{D}^{(m)}\mathbf{D}^{(1)} & \mathbf{D}^{(m)}\mathbf{D}^{(2)} & \cdots & \mathbf{D}^{(m)}\mathbf{W}^{(m)}\mathbf{D}^{(m)} \end{bmatrix}
\end{aligned}$$

Denote $\mathbf{F}_{\cdot r} = [\mathbf{f}^{(1)T}, \mathbf{f}^{(2)T}, \cdots, \mathbf{f}^{(m)T}]^T$, where $\mathbf{f}^{(k)} \in \mathbb{R}^n$, and $\mathbf{Y}_{\cdot r} = [\mathbf{Y}_{\cdot s}^{(p)T}, \mathbf{Y}_{\cdot s}^{(p)T}, \cdots, \mathbf{Y}_{\cdot s}^{(p)T}]^T$ where $\mathbf{Y}_{\cdot s}^{(p)T} \in \mathbb{R}^n$ is the s-th column in matrix $\mathbf{Y}^{(p)}$. Taking them back into Eq.(6), for any $q = 1, \cdots, m$, we have

$$\mathbf{f}^{(q)} = \alpha\mathbf{D}^{(q)}\mathbf{W}^{(q)}\mathbf{D}^{(q)}\mathbf{f}^{(q)} + \alpha\sum_{l \neq q} \mathbf{D}^{(q)}\mathbf{D}^{(l)}\mathbf{f}^{(l)} + (1-\alpha)\mathbf{Y}_{\cdot s}^{(p)}. \tag{8}$$

Therefore, we can update $\mathbf{F}_{\cdot r}$ by using Eq.(8) when $q = 1, \cdots, m$. Notice that, $\mathbf{D}^{(q)}$'s are diagonal matrices, and $\mathbf{W}^{(q)}$'s are often sparse matrices. Therefore, the time complexity may be reduced further. The detailed time complexity analysis is shown in the following subsection.

3.3 Update Multiplex with Representation

After updating the representations of clusters in the p-th base result, which are $\mathbf{F}_{.(\sum_{q=1}^{p-1} c_q+1)}, \cdots, \mathbf{F}_{.(\sum_{q=1}^{p} c_q)}$, we will update the p-th layer of the multiplex.

The updation is quite simple. We first extract the representation matrix $\mathbf{F}^{(p)} \in \mathbb{R}^{n \times c}$ of the instances $\{x_1^p, \cdots, x_n^p\}$ in the p-th layer from $\mathbf{F}$. $\mathbf{F}^{(p)}$ is a submatrix of $\mathbf{F}$, which contains n rows of $\mathbf{F}$ (i.e., from the $(n(p-1)+1)$-th row to the np-th row of $\mathbf{F}$). Then, we construct a k-nn graph with $\mathbf{F}^{(p)}$. The process is shown in the red dashed box in Figure 3.

When constructing the k-nn graph, we need to compute the similarity between any two vertices. Here we use the cosine similarity as we did when constructing the initial graph. More formally, for data x_i^p and x_j^p, we extract the representations of these two vertices, which are $\mathbf{F}_{i.}^{(p)}$ and $\mathbf{F}_{j.}^{(p)}$, respectively. Then, we compute the similarity matrix $\mathbf{S}$ whose (i,j)-th element is defined as:

$$S_{ij} = \frac{\left\langle \mathbf{F}_{i.}^{(p)}, \mathbf{F}_{j.}^{(p)} \right\rangle}{\|\mathbf{F}_{i.}^{(p)}\|_2 \|\mathbf{F}_{j.}^{(p)}\|_2}, \tag{9}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner production of two vectors. Then we update the p-th layer of the multiplex as the k-nn graph from $\mathbf{S}$, i.e., if x_i^p is one of the k neighbors of x_j^p or x_j^p is one of the k neighbors of x_i^p according to $\mathbf{S}$, then there is an edge whose weight is $W_{ij}^{(p)} = S_{ij}$ between x_i^p and x_j^p. Notice that, according to Eq.(9), we have $S_{ij} \leq 1$, which means the weights of the intra-layer edges are no larger than the weights of inter-layer edges (which are fixed to 1). It is reasonable. The inter-layer edges link x_i^p and x_i^q, which are exactly the same instance, and thus they should be stronger than the intra-layer edges, which link two different instances.

3.4 Implementation Details and Discussion

We jointly update the representations and the base graphs in the m layers of the multiplex with T epochs. For simplicity, in our implementation, we fix $T = 5$. After T epochs, we obtain the final multiplex $\mathcal{G}'$, whose adjacency matrices of m layers are $\mathbf{W}^{(1)}, \cdots, \mathbf{W}^{(m)}$. We fix the hyper-parameter $\alpha = 0.85$. If the data is a small size data, which means the number of instances is smaller than 1000, we fix $k = 10$ for k-nn graph construction, and otherwise, we fix $k = 20$. The whole process is summarized in Algorithm 1.

After learning the final multiplex, we can use any multiple graph based clustering ensemble methods with $\mathbf{W}^{(1)}, \cdots, \mathbf{W}^{(m)}$ as inputs to obtain the final clustering result. In this paper, since the motivation is to show the importance of base refinement, we just use the most straightforward way to obtain the final clustering result. We compress the multiplex by averaging all layers to obtain a single matrix $\mathbf{W}'$ as: $\mathbf{W}' = \frac{1}{m} \sum_{p=1}^{m} \mathbf{W}^{(p)}$. Then, we run spectral clustering on $\mathbf{W}'$ to obtain the final result.

Now, we analyze the space and time complexity of Algorithm 1. Notice that, given a multiplex, we only need to save the m diagonal blocks $\mathbf{W}^{(1)}, \cdots, \mathbf{W}^{(m)}$. Other blocks are all identity matrices which we do not need to save. Therefore, the space complexity of the multiplex is $O(mn^2)$.

Algorithm 1 Clustering Ensemble with Adaptive Multiplex

Input: Multiple base clustering results $\mathcal{C}^1, \cdots, \mathcal{C}^m$, number of iterations $T = 5$, $\alpha = 0.85$, number $k = 10$ or 20 of neighbors in k-nn graph.
Output: The final multiplex $\mathcal{G}'$.
1: Obtain $\mathbf{Y}^{(1)}, \cdots, \mathbf{Y}^{(m)}$ from base results, construct the initial representation $\mathbf{Y}$, and initialize $\mathbf{F} = \mathbf{Y}$
2: Construct the initial multiplex $\mathcal{G}$ from $\mathbf{Y}^{(1)}, \cdots, \mathbf{Y}^{(m)}$, and initialize $\mathcal{G}' = \mathcal{G}$.
3: **for** $t = 1, \cdots, T$ **do**
4: **for** $p = 1, \cdots, m$ **do**
5: Update each column from the $(\sum_{q=1}^{p-1} c_q + 1)$-th to $(\sum_{q=1}^{p} c_q)$-th column in $\mathbf{F}$ by Eq.(8) until convergence.
6: Update the p-th layer of multiplex by constructing the k-nn graph on the similarity matrix defined in Eq.(9).
7: **end for**
8: **end for**

TABLE 1
Description of the data sets.

	#instances	#features	#classes	Type
20NG	3970	1000	4	Text
AR	840	768	120	Image
BBC	737	1000	5	Text
HAPT	10929	561	12	Time-Series
Hitech	2301	22498	6	Text
News4a	3840	4989	4	Text
News4b	3874	5652	4	Text
ORL	400	1024	40	Image
Orlraws	100	10304	10	Image
Yale	165	1024	15	Image

The space complexity of saving $\mathbf{F}$ and $\mathbf{Y}$ are $O(mnc)$ which is smaller than $O(mn^2)$. To sum up, the whole space complexity is $O(mn^2)$, which is comparable with the multiple graph based methods. Moreover, if the graph in each layer is sparse, the space complexity can be further reduced. For example, if the graph is k-nn graph, which means in each row, there are $O(k)$ non-zero elements, the space complexity can be reduced to $O(mnk)$. When updating the representation, we use the manifold ranking, which is an iterative method. Here, we suppose the number of iterations is t_1, where in our implementation, we fix $t_1 = 20$. In one iteration, we apply Eq.(8) to update $\mathbf{F}$. Notice that, in all iterations except the first one, $\mathbf{W}^{(q)}$ is a sparse adjacency matrix of a k-nn graph. Therefore, in each row, the number of non-zero elements is $O(k)$. The number of multiplication operation when computing $\mathbf{D}^{(q)}\mathbf{W}^{(q)}\mathbf{D}^{(q)}\mathbf{f}^{(q)}$ is $O(nk)$. In the first iteration, we compute $\mathbf{D}^{(q)}\mathbf{f}^{(q)}$ with $O(n)$ multiplication operations, and notice that $\mathbf{W}^{(q)}$ in the first iteration only contains 0 and 1, and thus $\mathbf{W}^{(q)}\mathbf{D}^{(q)}\mathbf{f}^{(q)}$ only involves add operations (which are much faster than multiplication operations) without any multiplication operations. The time complexity of computing $\sum_{l \neq q} \mathbf{D}^{(q)}\mathbf{D}^{(l)}\mathbf{f}^{(l)}$ is $O((m-1)n)$. Hence, the time complexity of Line 5 in Algorithm 1 is $O(m(nk + mn)t_1 c)$, which is linear with the number of instances. Line 6 is to construct the k-nn graph whose time complexity is $O(n^2 c)$. Therefore, the whole time complexity is $O(mT(m(k+m)t_1 cn + n^2 c))$. In practice, we often have $m, T, t_1, k, c \ll n$, and thus the bottleneck of the time complexity is to construct the k-nn graph. Fortunately, there are

TABLE 2
ACC results on all the data sets. The best results of methods are in **boldface**.

Methods	20NG	AR	BBC	HAPT	Hitech	News4a	News4b	ORL	Orlraws	Yale
km-avg	0.2590 ±0.0018	0.3282 ±0.0037	0.3896 ±0.0150	0.4923 ±0.0105	0.3201 ±0.0070	0.2823 ±0.0083	0.2577 ±0.0012	0.5118 ±0.0086	0.6475 ±0.0325	0.4670 ±0.0083
km-best	0.2789 ±0.0131	0.3471 ±0.0069	0.4624 ±0.0488	0.5341 ±0.0115	0.3459 ±0.0094	0.3026 ±0.0069	0.2646 ±0.0066	0.5620 ±0.0131	0.7630 ±0.0353	0.5430 ±0.0202
CSPA [1]	0.3144 ±0.0301	0.3546 ±0.0113	0.3632 ±0.0642	0.3912 ±0.0094	0.3017 ±0.0297	0.3479 ±0.0231	0.3177 ±0.0133	0.5795 ±0.0156	0.7370 ±0.0677	0.5291 ±0.0387
HGPA [1]	0.2564 ±0.0022	0.3669 ±0.0086	0.2796 ±0.0124	0.4001 ±0.0338	0.2366 ±0.0070	0.2547 ±0.0019	0.2572 ±0.0030	0.5902 ±0.0214	0.7680 ±0.0257	0.4945 ±0.0216
MCLA [1]	0.2638 ±0.0060	0.3207 ±0.1099	0.3228 ±0.0247	0.4707 ±0.0200	0.2841 ±0.0064	0.2809 ±0.0122	0.2644 ±0.0042	0.6090 ±0.0279	0.7430 ±0.0340	0.5297 ±0.0091
NMFC [35]	0.2767 ±0.0148	0.3560 ±0.0097	0.3820 ±0.0347	0.4832 ±0.0355	0.3177 ±0.0120	0.3002 ±0.01884	0.2592 ±0.0071	0.5695 ±0.0274	0.7060 ±0.0698	0.5242 ±0.0227
LWEA [36]	0.2535 ±0.0016	0.3738 ±0.0069	0.3843 ±0.0291	**0.5110** ±0.0309	0.3174 ±0.0034	0.2684 ±0.0176	0.2572 ±0.0012	0.5540 ±0.0309	0.7580 ±0.0636	0.5194 ±0.0259
LWGP [36]	0.2599 ±0.0068	0.3702 ±0.0097	0.4064 ±0.0299	0.5035 ±0.0196	0.3189 ±0.0082	0.2953 ±0.0070	0.2552 ±0.0007	0.6042 ±0.0305	0.7630 ±0.0392	0.5467 ±0.0200
RSEC [8]	0.2762 ±0.0139	0.3081 ±0.0113	0.3640 ±0.0403	0.4623 ±0.1064	0.3013 ±0.0268	0.2989 ±0.0203	0.2598 ±0.0067	0.4442 ±0.0229	0.5790 ±0.1068	0.4024 ±0.0391
DREC [37]	0.2569 ±0.0016	0.3637 ±0.0119	0.4098 ±0.0561	0.4831 ±0.0251	0.3234 ±0.0088	0.2807 ±0.0170	0.2568 ±0.0020	0.5973 ±0.0206	0.7480 ±0.0464	0.5533 ±0.0327
SCCBG [23]	0.2524 ±0.0012	0.3265 ±0.0108	0.3864 ±0.0189	0.4710 ±0.0580	0.3161 ±0.0337	0.2726 ±0.0009	0.2565 ±0.0012	0.5738 ±0.0199	0.7210 ±0.0524	0.5273 ±0.0350
SPCE [22]	0.2555 ±0.0022	0.3438 ±0.0070	0.3760 ±0.0218	0.4809 ±0.0218	0.3126 ±0.0242	0.2633 ±0.0180	0.2559 ±0.0012	0.5710 ±0.0212	0.7460 ±0.0499	0.4806 ±0.0302
TRCE [10]	0.2542 ±0.0028	0.3457 ±0.0075	0.3991 ±0.0283	0.5034 ±0.0250	0.3198 ±0.0087	0.2761 ±0.0221	0.2573 ±0.0018	0.5888 ±0.0125	0.7560 ±0.0207	0.5564 ±0.0228
CESHL [5]	0.2545 ±0.0023	0.3592 ±0.0103	0.3678 ±0.0111	0.4872 ±0.0403	0.3154 ±0.0289	0.2632 ±0.0012	0.2569 ±0.0010	0.5800 ±0.0386	0.7770 ±0.0408	0.5200 ±0.0482
CEAM	**0.4330** ±0.0207	**0.3861** ±0.0081	**0.4125** ±0.0766	0.5057 ±0.0485	**0.4425** ±0.0253	**0.4485** ±0.0261	**0.4360** ±0.0637	**0.6307** ±0.0208	**0.7870** ±0.0245	**0.5800** ±0.0326

several ways to speed up the k-nn graph construction, such as kd-tree and locality sensitivity hashing. In the future, we will study how to plug them into our framework to speed up the method.

4 EXPERIMENTS

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

4.1 Data Sets

We use 10 data sets, including 20NG [38], AR [39], BBC [40], HAPT [41], Hitech [40], News4a [42], News4b [42], ORL [43], Orlraws[2], Yale [44]. The detailed information can be found in Table 1.

4.2 Experimental Setup

We run kmeans 100 times with random initializations to obtain 100 base results, and then partition them into 10 groups, where each group contains 10 base results. We run clustering ensemble methods on each group and report the average results over the 10 groups. We report **km-avg** which is the average result of the base results, and **km-best** which is the best result of the base results, as the baselines. Moreover, we compare with the following clustering ensemble methods:

- **CSPA** [1]. It is a cluster-based similarity partition algorithm, which constructs the similarity with the base results and obtains the consensus result from the similarity.
- **HGPA** [1]. It is a hypergraph partitioning based algorithm for clustering ensemble.
- **MCLA** [1]. It is a meta-clustering algorithm for clusterin ensemble which reformulate the clustering ensemble problem to a cluster correspondence problem.
- **NMFC** [35]. It is a clustering ensemble method with nonnegative matrix factorization.
- **LWEA** [36]. It is an evidence accumulation method for clustering ensemble with a local weighting strategy.
- **LWGP** [36]. It is a graph partitioning method for clustering ensemble with a local weighting strategy.
- **RSEC** [8]. It is a robust spectral clustering method on the co-association matrix.
- **DREC** [37]. It is a clustering ensemble method with dense representation learning.
- **SCCBG** [23]. It is a clustering ensemble method with structured bipartite graph learning.
- **SPCE** [22]. It is a self-paced clustering ensemble method by multiple graph learning.
- **TRCE** [10]. It is a tri-level robust multiple graph learning method for clustering ensemble.
- **CESHL** [5]. It is a clustering ensemble method based

2. https://jundongl.github.io/scikit-feature/datasets.html

TABLE 3
NMI results on all the data sets. The best results of methods are in **boldface**.

Methods	20NG	AR	BBC	HAPT	Hitech	News4a	News4b	ORL	Orlraws	Yale
km-avg	0.0098 ±0.0013	0.6519 ±0.0026	0.0807 ±0.0199	0.5464 ±0.0049	0.0976 ±0.0067	0.0266 ±0.0078	0.0066 ±0.0007	0.7213 ±0.0047	0.7395 ±0.0172	0.5138 ±0.0084
km-best	0.0261 ±0.0113	0.6634 ±0.0035	0.1774 ±0.0514	0.5709 ±0.0122	0.1444 ±0.0256	0.0477 ±0.0077	0.0125 ±0.0025	0.7524 ±0.0078	0.8157 ±0.0224	0.5585 ±0.0171
CSPA [1]	0.0245 ±0.0220	0.7004 ±0.0051	0.1115 ±0.0522	0.4715 ±0.0114	0.1139 ±0.0305	0.0483 ±0.0130	0.0227 ±0.0062	0.7656 ±0.0062	0.7893 ±0.0484	0.5612 ±0.0185
HGPA [1]	0.0002 ±0.0002	0.7004 ±0.0043	0.0487 ±0.0122	0.3814 ±0.0560	0.0311 ±0.0063	0.0001 ±0.0001	0.0004 ±0.0003	0.7741 ±0.0102	0.8158 ±0.0164	0.5438 ±0.0216
MCLA [1]	0.0042 ±0.0019	0.6068 ±0.213	0.0452 ±0.0222	0.5239 ±0.0116	0.0710 ±0.0232	0.0068 ±0.0040	0.0023 ±0.0007	0.7738 ±0.0087	0.7963 ±0.0244	0.5540 ±0.0097
NMFC [35]	0.0228 ±0.0129	0.6841 ±0.0038	0.1224 ±0.0351	0.5337 ±0.0129	0.1227 ±0.0285	0.0413 ±0.0144	0.0079 ±0.0037	0.7630 ±0.0110	0.7867 ±0.0324	0.5664 ±0.0193
LWEA [36]	0.0047 ±0.0024	0.6659 ±0.0066	0.0579 ±0.0326	0.5559 ±0.0271	0.0913 ±0.0026	0.0143 ±0.0157	0.0049 ±0.0014	0.7656 ±0.0143	0.8171 ±0.0310	0.5374 ±0.0248
LWGP [36]	0.0116 ±0.0064	0.6734 ±0.0063	0.1142 ±0.0356	0.5461 ±0.0085	0.1081 ±0.0088	0.0393 ±0.0072	0.0064 ±0.0014	0.7798 ±0.0154	0.8219 ±0.0233	0.5672 ±0.0164
RSEC [8]	0.0224 ±0.0123	0.6058 ±0.0130	0.0938 ±0.0327	0.4767 ±0.1692	0.0995 ±0.0480	0.0395 ±0.0178	0.0078 ±0.0039	0.6564 ±0.0237	0.6492 ±0.0765	0.4496 ±0.0261
DREC [37]	0.0088 ±0.0024	0.6561 ±0.0047	0.1334 ±0.0562	0.5393 ±0.0075	0.1200 ±0.0143	0.0263 ±0.0162	0.0053 ±0.0011	0.7747 ±0.0090	0.8104 ±0.0279	0.5742 ±0.0196
SCCBG [23]	0.0032 ±0.0018	0.5720 ±0.0115	0.0712 ±0.0263	0.5578 ±0.0419	0.0889 ±0.0350	0.0233 ±0.0008	0.0222 ±0.0010	0.7441 ±0.0163	0.7736 ±0.0383	0.5435 ±0.0281
SPCE [22]	0.0072 ±0.0030	**0.7294** ±0.0025	0.0302 ±0.0293	0.5560 ±0.0293	0.0820 ±0.0205	0.0106 ±0.0171	0.0032 ±0.0018	0.7728 ±0.0069	0.8045 ±0.0400	0.5758 ±0.0116
TRCE [10]	0.0044 ±0.0040	0.6120 ±0.0137	0.0915 ±0.0340	0.5471 ±0.0132	0.0921 ±0.0110	0.0214 ±0.0197	0.0036 ±0.0020	0.7599 ±0.0080	0.8053 ±0.0161	0.5720 ±0.0102
CESHL [5]	0.0059 ±0.0034	0.5856 ±0.0197	0.0161 ±0.0210	0.5649 ±0.0326	0.0612 ±0.0324	0.0095 ±0.0010	0.0040 ±0.0013	0.7613 ±0.0201	0.8181 ±0.0323	0.5450 ±0.0363
CEAM	**0.2294** ±0.0494	0.7006 ±0.0052	**0.1861** ±0.0697	**0.5665** ±0.0304	**0.2579** ±0.0284	**0.2399** ±0.0280	**0.2703** ±0.0899	**0.7899** ±0.0091	**0.8295** ±0.0160	**0.5884** ±0.0192

TABLE 4
Ablation Study. "w.o. multiplex" represents the results without the multiplex. "w.o. updating multiplex" represents the results with multiplex but without updating the multiplex. CEAM is our original method. The best results of methods are in **boldface**.

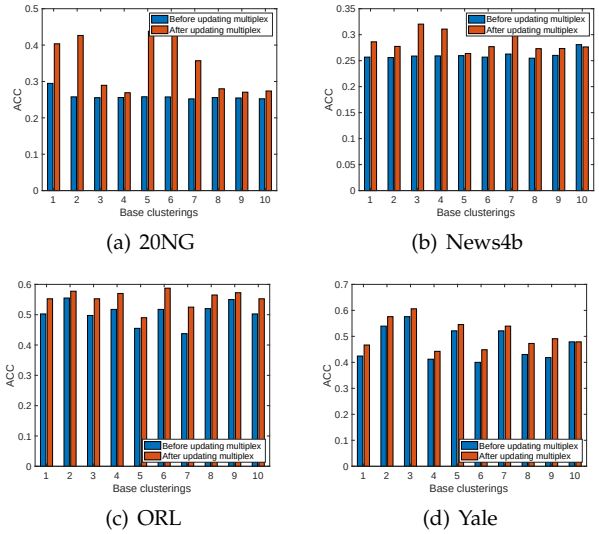
Methods	Metric	20NG	AR	BBC	HAPT	Hitech	News4a	News4b	ORL	Orlraws	Yale
w.o. multiplex	ACC	0.2753 ±0.0148	0.3701 ±0.0137	0.4038 ±0.0667	0.4953 ±0.0198	0.3159 ±0.0163	0.3152 ±0.0194	0.2631 ±0.0075	0.6085 ±0.0093	0.7620 ±0.0476	0.5515 ±0.0431
	NMI	0.0218 ±0.0118	0.6829 ±0.0062	0.1747 ±0.0670	0.5352 ±0.0164	0.1530 ±0.0274	0.0565 ±0.0208	0.0128 ±0.0038	0.7792 ±0.0042	0.8087 ±0.0308	0.5750 ±0.0241
w.o. updating multiplex	ACC	0.2797 ±0.0108	0.3849 ±0.0122	0.3788 ±0.0232	0.4570 ±0.0310	0.3489 ±0.0293	0.3329 ±0.0167	0.2990 ±0.0401	0.6198 ±0.0153	0.7690 ±0.0285	0.5788 ±0.0254
	NMI	0.0280 ±0.0128	0.6982 ±0.0069	0.1308 ±0.0541	0.5234 ±0.0181	0.2182 ±0.0468	0.0676 ±0.0228	0.0604 ±0.0237	0.7865 ±0.0071	0.8124 ±0.0188	0.5852 ±0.0180
CEAM	ACC	**0.4330** ±0.0207	**0.3861** ±0.0081	**0.4125** ±0.0766	**0.5057** ±0.0485	**0.4425** ±0.0253	**0.4485** ±0.0261	**0.4360** ±0.0637	**0.6307** ±0.0208	**0.7870** ±0.0245	**0.5800** ±0.0326
	NMI	**0.2294** ±0.0494	**0.7006** ±0.0052	**0.1861** ±0.0697	**0.5665** ±0.0304	**0.2579** ±0.0284	**0.2399** ±0.0280	**0.2703** ±0.0899	**0.7899** ±0.0091	**0.8295** ±0.0160	**0.5884** ±0.0192

TABLE 5
Results of SPCE with and without the refined base results. "SPCE w.o. refining" denotes the SPCE with original base graphs as inputs. "SPCE w. refining" denotes the SPCE with the refined graphs obtained by our method as inputs. The best results of methods are in **boldface**.

Methods	Metric	20NG	AR	BBC	HAPT	Hitech	News4a	News4b	ORL	Orlraws	Yale
SPCE w.o. refining	ACC	0.2555 ±0.0022	0.3438 ±0.0070	0.3760 ±0.0218	0.4809 ±0.0218	0.3126 ±0.0242	0.2633 ±0.0180	0.2559 ±0.0012	0.5710 ±0.0212	0.7460 ±0.0499	0.4806 ±0.0302
	NMI	0.0072 ±0.0030	0.7294 ±0.0025	0.0302 ±0.0293	0.5560 ±0.0293	0.0820 ±0.0205	0.0106 ±0.0171	0.0032 ±0.0018	0.7728 ±0.0069	0.8045 ±0.0400	0.5758 ±0.0116
SPCE w. refining	ACC	**0.4545** ±0.0134	**0.3511** ±0.0089	**0.4518** ±0.0716	**0.4863** ±0.0426	**0.4186** ±0.0284	**0.4673** ±0.0105	**0.4678** ±0.0090	**0.5963** ±0.0232	**0.7650** ±0.0389	**0.5224** ±0.0225
	NMI	**0.2455** ±0.0441	**0.7355** ±0.0023	**0.2550** ±0.0209	**0.7355** ±0.0023	**0.2640** ±0.0125	**0.2430** ±0.0206	**0.3124** ±0.0192	**0.7905** ±0.0056	**0.8113** ±0.0236	**0.5931** ±0.0122

on a structured dynamical hypergraph.

For all methods on all data sets, the number of clusters is set as the true number of the classes for a fair comparison. Since the hyper-parameter selection is difficult for unsupervised learning, the hyper-parameters of our method are fixed as introduced in the previous section and we do not tune them in experiments. We use Accuracy (ACC) and Normalized Mutual Information (NMI) to evaluate the performance.



(a) 20NG (b) News4b

(c) ORL (d) Yale

Fig. 4. ACC of each base result (before updating multiplex (blue bars) v.s. after updating multiplex (red bars)).

4.3 Experimental Results

Tables 2 and 3 show the ACC and NMI results of the proposed CEAM and other compared methods. CEAM outperforms the km-avg significantly which shows the effectiveness of the ensemble. Even compared with km-best, CEAM outperforms it on most data sets, whereas many other ensemble methods cannot outperform km-best on many data sets. It shows that although CEAM uses some unreliable base results, these weak results do not mislead CEAM as they do to other methods. The reason may be that CEAM refines the base results by the ensemble, and thus it ensembles the better base results instead of the weak initial ones, which can easily achieve better performance.

When compared with other state-of-the-art clustering ensemble methods, CEAM also achieves better performance on most data sets. Notice that, the ensemble process in CEAM is very simple and straightforward, which is just running spectral clustering on the averaged graph of the refined multiplex. It well demonstrates that the performance gain may mainly come from the base results refinement. Notice that on some data sets, the improvements of our CEAM are limited and on some data sets, such as HAPT, the compared method can even perform better than our CEAM. We think the reason may be that on some data sets the performance improvement caused by our base results refining is limited. According to the ablation study in Table 4, we can see that, although on all data sets our base refining can improve the performance, on some data sets, such as HAPT and AR, the improvement is limited. This may limit the performance of our method because the main part of CEAM is the base refining and our ensemble strategy is quite simple. So, there is another interesting question here why on some data sets the improvement caused by our base refining is large while on some other data sets the improvement is limited? We think it may be related to the quality or diversity of the base results. For example, if all base results are too similar, which means the diversity is low, the base refining process will hardly work, because other base results cannot provide more useful information to refine one base result. On the contrary, if the diversity is high, which means other base results often contain much different information which may have a large chance to improve one base result.

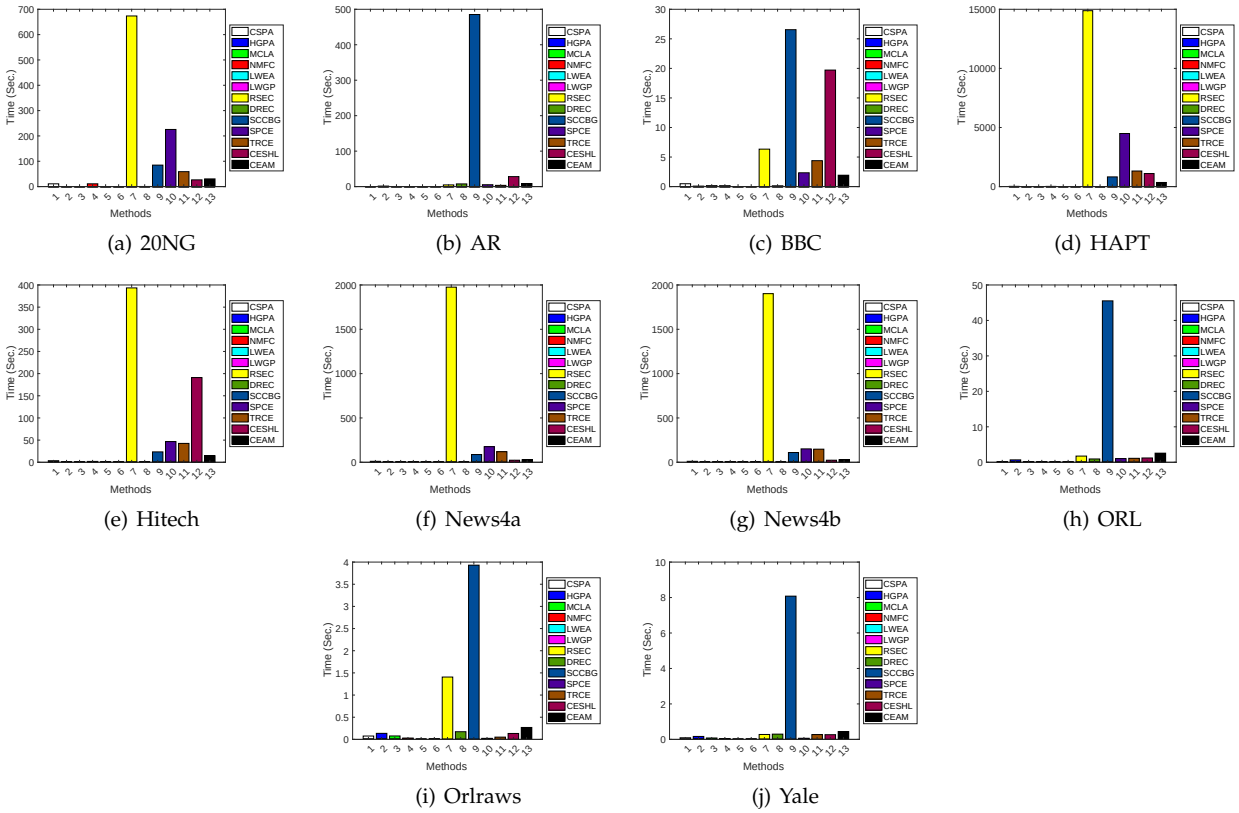
To verify it, given any two base results, we compute the NMI between these two base results. If the NMI is high, it means that the two base results are similar and thus the diversity is low. On the contrary, if the NMI is low, which means these two base results are dissimilar, the diversity between them is high. We compute the average NMI between all these base result pairs to evaluate the overall diversity of the base results. On some data sets with large improvements, such as 20NG and News4a, we obtain the average NMI of 0.4444 and 0.3612, respectively. However, on some data sets with limited improvements, such as HAPT and ORL, we obtain the average NMI of 0.7778 and 0.7490, which are relatively higher. Therefore, given the base results with large diversity, our method can improve more remarkably.

To further show the effectiveness of updating the multiplex, we show the comparison of the 10 base results before and after updating the multiplex in Figure 4. Figure 4 shows ACC on 20NG, News4b, ORL, and Yale, and the other results are similar. The blue bars represent the ACC of the initial base result before updating, and the red ones represent the ACC of each base result after updating. Figure 4 clearly shows that the multiplex updation indeed improves the quality of base results, which demonstrates our motivation. Moreover, on 20NG data set, the improvements in base results are more remarkable, which also demonstrates the previous discussion.

4.4 Ablation Study

In this subsection, we conduct some ablation studies to show the effects of the multiplex and multiplex updation. In more detail, we compare our method with two degenerated versions of CEAM. The first is the one without the multiplex, which is denoted as w.o. multiplex. It does not construct the multiplex and does not learn the representation. It directly runs spectral clustering on the averaged graph. The second is the one that uses the multiplex but does not update it, which is denoted as w.o. updating multiplex. It constructs the initial multiplex, learns the representation on the multiplex, and at last obtains the consensus clustering result from the representation without updating the multiplex. The results are shown in Table 4.

From Table 4, we can see that, on most data sets, the version using multiplex performs better than the version without multiplex, which shows the effectiveness of the

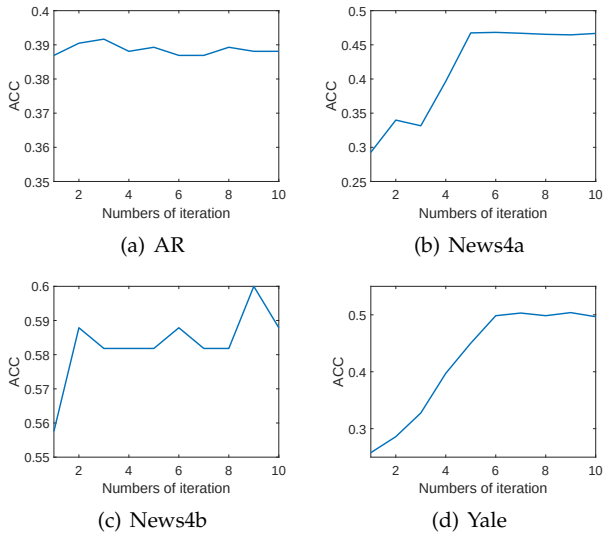


(a) 20NG (b) AR (c) BBC (d) HAPT

(e) Hitech (f) News4a (g) News4b (h) ORL

(i) Orlraws (j) Yale

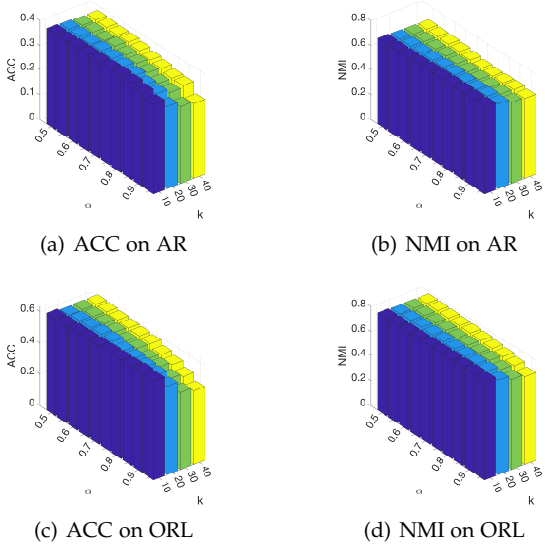
Fig. 5. Running time (Sec.) of all methods on all data sets.



(a) AR (b) News4a

(c) News4b (d) Yale

Fig. 6. ACC results w.r.t. different numbers of iterations on AR, News4a, News4b, and YAle.



(a) ACC on AR (b) NMI on AR

(c) ACC on ORL (d) NMI on ORL

Fig. 7. ACC and NMI w.r.t. different α and k on AR and ORL.

multiplex. Moreover, when comparing CEAM with the one without updating the multiplex, we can find that if we update the multiplex, the performance can further be improved. It well demonstrates the motivation of refining base results.

4.5 Effects of Base Results Refining

As claimed in Section Introduction, our base results refining method can be followed by any multiple graph based methods. To further show the effects of the base results refining, we run SPCE [22], which is a multiple graph based clustering ensemble, with the original graphs and with the refined graphs for comparison. The results are shown in Table 5. SPCE w.o. refining denotes the SPCE with original

base graphs as inputs and SPCE w. refining denotes the SPCE with the refined graphs obtained by our method as inputs. From Table 5, we can find that the refined graphs can indeed improve the performance of SPCE. Even on some data sets, e.g. 20NG and News4a, the improvements are also large. It well demonstrates the superiority of the multiplex refining in our method.

4.6 Effects of Numbers of Iterations

In our method, we fix the numbers of iterations with $T = 5$. In this subsection, we show the ACC performance with different numbers of iterations T. The results on AR, News4a, News4b, and Yale are shown in Figure 6. The results on other data sets are similar. From 6, we can see that, our method can achieve relatively good results when $T = 5$. Although on some data sets more iterations can lead to better results, more iterations also increase the running time. By considering the trade-off of effectiveness and efficiency, we choose $T = 5$ for our method.

4.7 Efficiency Experiments

We show the running time of all methods on all data sets in Figure 6. From Figure 6, we can see that CEAM is faster than recent graph based methods, such as RSEC, SPCE, and SCCBG, on most data sets. Especially on the largest data set HAPT, CEAM needs 349s and RSEC needs 14883s. CEAM is about 42 times faster than RSEC. It demonstrates the efficiency of CEAM among the graph based clustering ensemble methods.

4.8 Hyper-parameter Study

Although in our experiments, we fix the hyper-parameters α and k, in this subsection, we still show the sensitivity of these hyper-parameters. Since α is suggested to be set large in previous study [34], [45], we show α in the set $\{0.5, 0.55, \cdots, 0.95\}$. We show the k-nn parameter k in the set $\{10, 20, 30, 40\}$. The results on AR and ORL are shown in Figure 7. Results on other data sets are similar. From Figure 7, we can see that CEAM is insensitive w.r.t. α and k in the given range. Therefore, we do not need to tune k and α in the experiments. Notice that, sometimes, the performance decreases when α is very large, e.g. when $\alpha = 0.95$. The reason is that, if α is too large, which means the term $\|\mathbf{F}_{\cdot r} - \mathbf{Y}_{\cdot r}\|_2^2$ hardly works, and we almost only optimize the first term $\sum_{i,j=1}^{mn} \frac{W_{ij}}{2} \left(\frac{F_{ir}}{\sqrt{d_i}} - \frac{F_{jr}}{\sqrt{d_j}} \right)^2$, which is prone to obtain a trivial solution that all elements in $\mathbf{F}_{\cdot r}$ are the same and cause the over-smoothing problem.

5 Conclusion

This paper proposed a simple yet effective clustering ensemble method via adaptive multiplex. Different from conventional methods, which focus on how to do the ensemble, this paper presented an alternative idea, which applied the ensemble to refine the base results. We constructed the initial multiplex and representations from the base results, and then simultaneously updated them to make the ensemble and refinement boost each other. At last, we obtained the final consensus clustering result directly from the refined multiplex. Extensive experiments shew that the proposed method outperformed the state-of-the-art clustering ensemble methods, which revealed its effectiveness and superiority.

Although the proposed method achieves good performance, there still exist some limitations. One is the high space complexity of the multiplex, which limits more complicated operations on the multiplex. In our method, we apply the sparse matrix to alleviate this limitation. In the future, we will explore some more sophisticated methods which can further reduce the space and time complexity and try to apply this method to some large-scale data. Moreover, the experimental results show that on some data sets the performance improvement of our base refining is limited. In the future, we will further study the reason and solution of this problem.

Acknowledgments

This work is supported by the National Natural Science Foundation of China grants 62176001, 61806003, and 61976129.

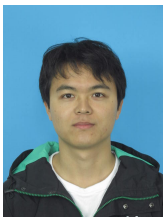
References

[1] A. Strehl and J. Ghosh, "Cluster ensembles — a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, no. 3, pp. 583–617, 2003.
[2] F. Wang, X. Wang, and T. Li, "Generalized cluster aggregation," in *IJCAI*, 2009, pp. 1279–1284.
[3] D. Huang, J. Lai, and C. Wang, "Ensemble clustering using factor graph," *Pattern Recognition*, vol. 50, pp. 131–142, 2016.
[4] F. Li, Y. Qian, J. Wang, and J. Liang, "Multigranulation information fusion: A dempster-shafer evidence theory-based clustering ensemble method," *Inf. Sci.*, vol. 378, pp. 389–409, 2017.
[5] P. Zhou, X. Wang, L. Du, and X. Li, "Clustering ensemble via structured hypergraph learning," *Inf. Fusion*, vol. 78, pp. 171–179, 2022.
[6] H. Liu, T. Liu, J. Wu, D. Tao, and Y. Fu, "Spectral ensemble clustering," in *SIGKDD*, 2015, pp. 715–724.
[7] Z. Tao, H. Liu, and Y. Fu, "Simultaneous clustering and ensemble." in *AAAI*, 2017, pp. 1546–1552.
[8] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "Robust spectral ensemble clustering via rank minimization," *ACM Transactions on Knowledge Discovery From Data*, vol. 13, no. 1, pp. 1–25, 2019.
[9] Z. Tao, H. Liu, J. Li, Z. Wang, and Y. Fu, "Adversarial graph embedding for ensemble clustering," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, S. Kraus, Ed. ijcai.org, 2019, pp. 3562–3568.
[10] P. Zhou, L. Du, Y.-D. Shen, and X. Li, "Tri-level robust clustering ensemble with multiple graph learning," in *Thirty-fifth AAAI Conference on Artificial Intelligence*, 2021, pp. 11 125–11 133.
[11] P. Zhou, B. Sun, X. Liu, L. Du, and X. Li, "Active clustering ensemble with self-paced learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023.
[12] L. S. Conde, M. R. del Río, R. Criado, J. F. Álvarez, A. G. del Amo, and S. Boccaletti, "Eigenvector centrality of nodes in multiplex networks," *Chaos*, vol. 23, no. 3, pp. 033 131–1, September 2013.
[13] N. Iam-on, T. Boongoen, S. M. Garrett, and C. J. Price, "A link-based approach to the cluster ensemble problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2396–2409, 2011.
[14] 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, 2019.
[15] A. P. Topchy, A. K. Jain, and W. F. Punch, "Clustering ensembles: Models of consensus and weak partitions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1866–1881, 2005.
[16] N. Nguyen and R. Caruana, "Consensus clusterings," in *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), October 28-31, 2007, Omaha, Nebraska, USA*. IEEE Computer Society, 2007, pp. 607–612.

[17] L. Bai, J. Liang, H. Du, and Y. Guo, "An information-theoretical framework for cluster ensemble," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 8, pp. 1464–1477, 2019.
[18] Z. Zhou and W. Tang, "Clusterer ensemble," *Knowledge Based Systems*, vol. 19, no. 1, pp. 77–83, 2006.
[19] P. Hore, L. O. Hall, and D. B. Goldgof, "A scalable framework for cluster ensembles," *Pattern Recognit.*, vol. 42, no. 5, pp. 676–688, 2009.
[20] N. Iam-on, T. Boongoen, S. M. Garrett, and C. J. Price, "A link-based cluster ensemble approach for categorical data clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 3, pp. 413–425, 2012.
[21] P. Zhou, L. Du, H. Wang, L. Shi, and Y. Shen, "Learning a robust consensus matrix for clustering ensemble via kullback-leibler divergence minimization," in *IJCAI*, 2015, pp. 4112–4118.
[22] P. Zhou, L. Du, X. Liu, Y. Shen, M. Fan, and X. Li, "Self-paced clustering ensemble," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 4, pp. 1497–1511, 2021.
[23] P. Zhou, L. Du, and X. Li, "Self-paced consensus clustering with bipartite graph," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, C. Bessiere, Ed. ijcai.org, 2020, pp. 2133–2139.
[24] P. Zhou, X. Liu, L. Du, and X. Li, "Self-paced adaptive bipartite graph learning for consensus clustering," *ACM Trans. Knowl. Discov. Data*, 2022, just Accepted.
[25] Z. Tao, H. Liu, S. Li, and Y. Fu, "Robust spectral ensemble clustering," in *CIKM*, 2016, pp. 367–376.
[26] D. Huang, C. Wang, J. Wu, J. Lai, and C. Kwoh, "Ultra-scalable spectral clustering and ensemble clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1212–1226, 2020.
[27] M. Magnani, O. Hanteer, R. Interdonato, L. Rossi, and A. Tagarelli, "Community detection in multiplex networks," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–35, 2021.
[28] F. Karimi, S. Lotfi, and H. Izadkhah, "Multiplex community detection in complex networks using an evolutionary approach," *Expert Systems with Applications*, vol. 146, p. 113184, 2020.
[29] D. T. Nguyen, H. Zhang, S. Das, M. T. Thai, and T. N. Dinh, "Least cost influence in multiplex social networks: Model representation and analysis," in *2013 IEEE 13th International Conference on Data Mining*. IEEE, 2013, pp. 567–576.
[30] W. Li, S. Tang, W. Fang, Q. Guo, X. Zhang, and Z. Zheng, "How multiple social networks affect user awareness: The information diffusion process in multiplex networks," *Physical Review E*, vol. 92, no. 4, p. 042810, 2015.
[31] D. N. Frank, "Barcrawl and bartab: software tools for the design and implementation of barcoded primers for highly multiplexed dna sequencing," *BMC bioinformatics*, vol. 10, no. 1, pp. 1–13, 2009.
[32] P. Rastin and R. Kanawati, "A multiplex-network based approach for clustering ensemble selection," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2015, Paris, France, August 25 - 28, 2015*. ACM, 2015, pp. 1332–1339.
[33] C. Lyu, Y. Shi, L. Sun, and C.-T. Lin, "Community detection in multiplex networks based on evolutionary multi-task optimization and evolutionary clustering ensemble," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2022.
[34] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on data manifolds," in *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*. MIT Press, 2003, pp. 169–176.
[35] T. Li and C. H. Q. Ding, "Weighted consensus clustering." in *SDM*, 2008, pp. 798–809.
[36] D. Huang, C.-D. Wang, and J.-H. Lai, "Locally weighted ensemble clustering," *IEEE Transactions on Cybernetics*, vol. 48, no. 5, pp. 1460–1473, 2018.
[37] J. Zhou, H. Zheng, and L. Pan, "Ensemble clustering based on dense representation," *Neurocomputing*, 2019.
[38] D. Cai, X. He, W. V. Zhang, and J. Han, "Regularized locality preserving indexing via spectral regression," in *Proceedings of the 16th ACM conference on Conference on information and knowledge management (CIKM'07)*, 2007, pp. 741–750.
[39] H. Wang, F. Nie, and H. Huang, "Globally and locally consistent unsupervised projection," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.
[40] D. Greene and P. Cunningham, "Producing accurate interpretable clusters from high-dimensional data," in *Knowledge Discovery in Databases: PKDD 2005, 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal, October 3-7, 2005, Proceedings*, ser. Lecture Notes in Computer Science, vol. 3721. Springer, 2005, pp. 486–494.
[41] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *21st European Symposium on Artificial Neural Networks, ESANN 2013, Bruges, Belgium, April 24-26, 2013*, 2013.
[42] M. Wu and B. Schölkopf, "A local learning approach for clustering," in *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*. MIT Press, 2006, pp. 1529–1536.
[43] D. Cai, X. He, J. Han, and H.-J. Zhang, "Orthogonal laplacianfaces for face recognition," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3608–3614, 2006.
[44] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, 1997.
[45] D. Lin, V. J. Wei, and R. C. Wong, "First index-free manifold ranking-based image retrieval with output bound," in *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*, 2019, pp. 1216–1221.

Peng Zhou received the B.E. degree in computer science and technology from University of Science and Technology of China in 2011, and Ph.D. degree in Computer Science from Institute of Software, Chinese Academy of Sciences in 2017. He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University. He has published more than 30 papers in highly regarded conferences and journals, including IEEE TKDE, IEEE TNNLS, IEEE TCYB, ACM TKDD, Pattern Recognition, Information Fusion, IJCAI, AAAI, SDM, ICDM, etc. His research interests include machine learning, data mining and artificial intelligence.

Boao Hu received the B.E. degree from Hainan University in 2020. He is currently pursuing the M.S. degree with the School of Computer Science and Technology, Anhui University. His research interests include machine learning and data mining.



Dengcheng Yan received the B.S. and Ph.D. degrees from the University of Science and Technology of China, in 2011 and 2017, respectively. From 2017 to 2018, he was a Core Technology Researcher and a Big Data Engineer with Research Institute of Big Data, iFlytek Company Ltd. He is currently a Lecturer with Anhui University, China. His research interests include graph neural networks and complex networks.

Liang Du received the B.E. degree in software engineering from Wuhan University, in 2007, and the Ph.D. degree in computer science from the Institute of Software, University of Chinese Academy of Sciences, in 2013. From July 2013 to July 2014, he was a Software Engineer with Alibaba Group. He was also an Assistant Researcher with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. He is currently an Associate Professor with Shanxi University. He has published more than 40 papers in top conferences and journals, including KDD, IJCAI, AAAI, ICDM, TKDE, SDM, and CIKM. His research interests include clustering with noise and heterogeneous data, ranking for feature selection, active learning, and document summarization.