Partial Clustering Ensemble

Peng Zhou, Liang Du, Xinwang Liu, Zhaolong Ling, Xia Ji, Xuejun Li, and Yi-Dong Shen

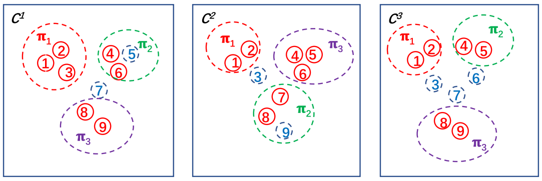
***Abstract*—Clustering ensemble often provides robust and stable results without accessing original features of data, and thus has been widely studied. The conventional clustering ensemble methods often take the full multiple base partitions as inputs and provide a consensus clustering result. However, in many real-world applications, full base partitions are hard to obtain because some data may be missing in some base partitions. To tackle this problem, in this paper, we propose a novel partial clustering ensemble method, which takes the partial multiple base partitions as inputs. In this method, we simultaneously fill the missing values in the base partitions and ensemble them by fully considering the consensus and diversity. Moreover, to address the unreliability issue in the partial data scenario, we seamlessly plug it into a self-paced learning framework. The extensive experiments on benchmark data sets demonstrate the effectiveness and efficiency of the proposed method when handling incomplete data.**

***Index Terms*—Clustering ensemble, incomplete data.**

I. Introduction

Clustering ensemble integrates multiple base partitions of data to obtain a consensus clustering result, and thus often provides a more robust and stable result compared with single clustering methods [1], [2]. When compared with other ensemble methods like multi-view clustering, clustering ensemble integrates information at the decision level, which takes the multiple base clustering results as inputs without accessing the features of original data. Therefore, it can protect the privacy of data to some extent [3].

Due to these advantages, clustering ensemble has been widely studied in recent years [4], [5], [6], [7], [8], [9]. For example, Zhou et al. provided an alignment method to ensemble multiple kmeans results [5]; Tao et al. designed a spectral clustering ensemble method via rank minimization [8]; Jia et al. applied low-rank tensor decomposition to clustering ensemble [9].



$$\mathbf{Y}^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ ? & ? & ? \\ 0 & 1 & 0 \\ ? & ? & ? \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{Y}^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ ? & ? & ? \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ ? & ? & ? \end{bmatrix} \quad \mathbf{Y}^{(3)} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ ? & ? & ? \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ ? & ? & ? \\ ? & ? & ? \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 1. Example of clustering ensemble on incomplete data. There are 9 instances and 3 base clusterings. Each red ball denotes an observed instance and each blue dotted ball represents a missing data. The dashed circles with red, green, and purple colors represent the clusters π_1, π_2, and π_3 in base clusterings, respectively. We also show the base partition matrices $\mathbf{Y}^{(1)}$, $\mathbf{Y}^{(2)}$ and $\mathbf{Y}^{(3)}$ below. The question marks in the matrices denote the missing values.

Although the above methods often achieve promising performance in clustering, they assume that *all* instances are observed in each base clustering. However, in many real applications, especially in federated learning scenarios, it often happens that some instances are missing in some base partitions. For example, in a bank system of a city, many people are customers of only a few banks, and when the banks do base clustering locally, some people are missing in some base results. When doing clustering ensemble in the cloud server, these conventional clustering ensemble methods may fail because all base results are incomplete. More formally, the inputs of clustering ensemble are multiple base partition matrices $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(m)} \in \{0,1\}^{n \times c}$, where n is the number of instances and c is the number of clusters. $\mathbf{Y}^{(i)}$ is the i-th base partition matrix where $Y_{pq}^{(i)} = 1$ if in the i-th base result, the p-th instance belongs to the q-th cluster, and $Y_{pq}^{(i)} = 0$ otherwise. If $\mathbf{Y}^{(i)}$ is incomplete, some rows of $\mathbf{Y}^{(i)}$ are missing. Fig. 1 shows an example with 9 instances and 3 base clusterings. In each base clustering $\mathcal{C}^i$, the dotted blue balls represent the missing data and the red balls represent the observed data. The partition matrices $\mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \mathbf{Y}^{(3)}$ are shown below, where the question mark ”?” represents the missing value in each base partition matrix.

To ensemble these partial $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(m)}$, one natural way is to impute each base partition matrix $\mathbf{Y}^{(i)}$ before ensemble. There are many imputation methods, such as zero filling, mean value filling, random filling, and k-nn filling [10]. However,

Manuscript received 2 October 2022; revised 2 July 2023; accepted 30 September 2023. Date of publication 4 October 2023; date of current version 5 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grants 62176001, 61806003, 62376146, 62306002, 62076002, 61976205, and 61972001, in part by the Natural Science Project of Anhui Provincial Education Department under Grant 2023AH030004, and in part by the Natural Science Foundation of Anhui Province under Grants 1908085MF188 and 2008085MF194. Recommended for acceptance by T. Weninger. *(Corresponding author: Peng Zhou.)*

Peng Zhou, Zhaolong Ling, Xia Ji, 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: zhoupeng@ahu.edu.cn; zlling@ahu.edu.cn; jixia@ahu.edu.cn; xjli@ahu.edu.cn).

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

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

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

This article has supplementary downloadable material available at https://doi.org/10.1109/TKDE.2023.3321913, provided by the authors.

Digital Object Identifier 10.1109/TKDE.2023.3321913

on one hand, since the original data are unaccessible and the only accessible $\mathbf{Y}^{(i)}$'s have a special structure, i.e., in each row, there should be only one 1 and other elements should be zeros, some filling methods like zero filling and mean value filling may be inappropriate; on the other hand, since the filling process is independent with the ensemble, some filling methods like random filling may introduce too many noises which makes the performance deteriorate. Some robust clustering ensemble methods, such as [11], [12], [13], view the missing data as noises, which can alleviate the incomplete data problem to some extent. However, they are originally designed for complete data instead of incomplete data, and just try to reduce the side-effects caused by missing data instead of filling them, and thus they may ignore some useful information behind the missing data.

Instead of filling the data before the ensemble, some incomplete multi-view learning methods provide another way, which simultaneously ensembles and fills data, so that the ensemble can guide the imputation [14], [15], [16], [17]. However, these methods take the features of original data as inputs, which are inaccessible in our setting, and thus they are hardly directly used to tackle our problem. One kind of the most related work is the late fusion method for incomplete multi-view learning, which generates multiple incomplete embedding from each view and ensembles them to obtain a consensus result [18], [19], [20]. To tackle our incomplete clustering ensemble problem, they regard $\mathbf{Y}^{(i)}$ as the incomplete embedding and use the late fusion methods to obtain the final consensus result. However, since these methods are designed for multi-view clustering specially, they ignore some important characteristics of clustering ensemble. For example, in the clustering ensemble problem, one important characteristic is the diversity of each base result, but these methods do not fully consider the diversity. Moreover, in clustering ensemble, one common assumption is that each base result is an unreliable weak result. Especially in the incomplete setting, since some data are missing, the base results are even more unreliable. However, these late fusion methods also ignore the unreliability of base results.

To tackle these problems, in this paper, we propose a novel Partial Clustering Ensemble (PCE) method, which takes multiple incomplete $\mathbf{Y}^{(i)}$ as inputs without accessing the original data. In our PCE, we impute the incomplete 0/1 discrete $\mathbf{Y}^{(i)}$ directly in the process of ensemble learning. When filling $\mathbf{Y}^{(i)}$, we design a simple yet effective objective function to keep the diversity of each filled $\mathbf{Y}^{(i)}$. When ensembling them, we apply self-paced learning to handle data in order of their reliability, i.e., we automatically evaluate the reliability of each data and ensemble them from more reliable data to less reliable data. Due to the self-paced learning, the side effects of unreliable data can be alleviated in the ensemble learning process. We seamlessly integrate the base partitions imputation and ensemble into a unified objective function, so that the two tasks can be boosted by each other. Then, we design an effective and efficient algorithm to optimize it, which can directly obtain the final partitions in an end-to-end way, without any uncertain postprocessing like kmeans.

The contributions of this paper are summarized as follows:

- We try to tackle a new problem of incomplete clustering ensemble. Although some robust clustering ensemble methods can also alleviate the side effects caused by the missing values, we propose a novel framework that simultaneously does the imputation and ensemble, which can make full use of the information in the base clusterings.
- We design a simple yet effective ensemble term that fully considers the consensus, diversity, and reliability of data.
- We provide an effective and efficient iterative algorithm to impute and ensemble the base results, and at last, it directly obtains the final consensus clustering result without any postprocessing.
- The extensive experiments demonstrate that the proposed method outperforms not only the conventional clustering ensemble methods with imputation as preprocessing, but also the state-of-the-art late fusion multi-view clustering methods.

II. Related Work

In this section, we briefly introduce some related work of clustering ensemble and incomplete multi-view clustering. We use bold uppercase letters to represent matrices and bold lowercase letters to represent vectors. Given a matrix $\mathbf{M}$, we use M_{pq} to denote the (p,q)-th element in the matrix $\mathbf{M}$.

A. Clustering Ensemble

Given a data set $\mathbf{X} \in \mathbb{R}^{n\times d}$ containing n instances and d features, we first run some standard clustering methods such as kmeans or spectral clustering on $\mathbf{X}$ to obtain m base clustering results $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(m)} \in \{0,1\}^{n\times c}$, where in each row of $\mathbf{Y}^{(i)}$ there is only one 1 and other elements are 0's. If $Y_{pq}^{(i)} = 1$, it means that the p-th instance $\mathbf{x}_p$ belongs to the q-th cluster in the i-th base result. Clustering ensemble takes the multiple base results $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(m)}$ as inputs and aims to learn a consensus clustering result $\mathbf{Y} \in \{0,1\}^{n\times c}$ from $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(m)}$. Notice that, in the clustering ensemble task, we only take the base results $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(m)}$ as inputs without accessing the original data set $\mathbf{X}$. It is the main difference between clustering ensemble and multi-view clustering [21], [22], [23], [24], [25], [26], [27], which is another related task to clustering ensemble. In multi-view clustering, it often takes the multiple feature matrices of original data $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(m)}$ as inputs and learns a consensus result. Since clustering ensemble does not need the original features of data, it can protect the privacy of data to some extent. Nevertheless, also due to this, clustering ensemble is more challenging than multi-view clustering.

Due to the absence of the original data, the clustering ensemble needs to learn a consensus result directly from base results. One widely studied way is to construct graphs from the base results and learn the consensus result via the multiple graph learning [28], [29], [30], [31], [32], [33], [34]. For example, Iam-On et al. constructed a similarity matrix from base results and obtain the final clustering result from the similarity matrix [28], [29]; Tao et al. designed a robust clustering ensemble method via spectral clustering on the multiple graphs [30]; Huang et al. proposed a fast spectral clustering method on the multiple graphs which is appropriate for large scale data sets [31]; Zhou et al. plugged self-paced learning into the multiple graph learning leading to the reliable clustering ensemble methods [32], [33], [35].

Although graph based methods often achieve promising performance, since they need to construct multiple graphs, the time and space complexity is relatively high. To address this issue, some methods use other data structures for the ensemble. For example, Strehl et al. and Zhou et al. applied hyper-graph to represent base results and learn the consensus result via hyper-graph partitioning methods [1], [36]; Huang et al. and Zhou et al. ensemble base results via bipartite graph partitioning [37], [38], [39]; Bai et al. integrated multiple kmeans results to handle non-linear data [40]; Abbasi et al. and Bagherinia et al. characterized the quality and diversity of base results and did clustering ensemble based on the quality and diversity [41], [42].

All of the above-mentioned methods require that the base results are complete and cannot directly handle incomplete data. Except for these methods, some robust clustering ensemble methods, such as [11], [12], [13], can be recast to handle incomplete data. They regard the missing data as noises and apply denoising methods to alleviate the incomplete data problem to some extent. However, after all, they are originally designed for complete data instead of incomplete data. What they do is reduce the side effects caused by the missing data rather than filling them, and thus they may ignore some useful information behind the missing data.

B. Incomplete Multi-View Clustering

Although clustering ensemble on incomplete data is quite underexplored, another related task, incomplete multi-view clustering [43], has attracted much more attention. Incomplete multi-view clustering takes the multiple feature matrices of original data $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(m)}$ as inputs to learn a consensus clustering result, where $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(m)}$ are incomplete. It means that some instances may be missing in some views. Incomplete multi-view clustering often simultaneously fills the missing data and does multi-view clustering.

Although some data are missing in some views, it is possible to find some aligned instances, which can recover the missing data in each view. With the help of these aligned instances, some methods learn a consensus representation from multiple views via matrix factorization and obtain the final clustering result from the consensus representation [14], [44], [45]. Besides, some methods transfer the incomplete multi-view learning to incomplete multiple kernel learning [16], [46], [47], [48]. These methods construct the incomplete kernel from multiple views and impute such multiple incomplete kernels in the process of multiple kernel learning. Most recently, some deep incomplete multi-view learning methods are proposed [49], [50], [51]. They apply deep neural networks to learn a consensus representation of multi-view data. To handle the missing data, they often use generative models, such as Generative Adversarial Network (GAN), to fill in the missing values.

Unfortunately, these methods can hardly be applied to handle the incomplete clustering ensemble problem, because they need the original features of data, which are unavailable in clustering ensemble tasks. Among incomplete multi-view clustering methods, the most related methods to our task are the late fusion methods [18], [19], [20]. Late fusion methods generate the embedding from each view and aim to ensemble the multiple incomplete embedding. For example, Liu et al. proposed an alignment method to ensemble multiple kernel kmeans results [19]; Zhang et al. proposed a one-stage clustering method to ensemble the embedding of each view [20]. In our clustering ensemble task, the input base result can be viewed as an embedding of the original data, and then we can apply the late fusion methods to the incomplete base results. However, as introduced before, multi-view clustering and clustering ensemble are two different tasks after all. Directly applying the late fusion incomplete multi-view clustering methods may ignore some important characteristics of the clustering ensemble, such as diversity and unreliability. Therefore, it is worth designing a specialized method for the incomplete clustering ensemble problem.

III. Partial Clustering Ensemble

In this section, we introduce PCE in more detail. In the classical clustering ensemble setting, given m base partition matrices $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(m)} \in \{0,1\}^{n \times c}$ of n instances in c clusters, clustering ensemble aims to learn a consensus clustering result $\mathbf{Y} \in \{0,1\}^{n \times c}$. In the incomplete setting, some rows in each $\mathbf{Y}^{(i)}$ are missing. More formally, we extract two sub-matrices $\mathbf{Y}_o^{(i)}$ and $\mathbf{Y}_u^{(i)}$ from $\mathbf{Y}^{(i)}$. $\mathbf{Y}_o^{(i)} \in \{0,1\}^{n_o^i \times c}$ indicates the results of observed instances in the i-th base partition and n_o^i is the number of the observed instances; $\mathbf{Y}_u^{(i)} \in \{0,1\}^{n_u^i \times c}$ is the results of missing instances in the i-th base partition which we need to fill in and n_u^i is the number of the missing instances, where $n_o^i + n_u^i = n$. Our goal is to simultaneously learn $\mathbf{Y}_u^{(i)}$ (i.e., do imputation) and $\mathbf{Y}$ (i.e., do ensemble).

A. Ensemble by Considering Consensus and Diversity

Notice that the clusters in each base result are not aligned. For example, the first cluster in $\mathbf{Y}^{(1)}$ is not necessarily to be the same as the first one in $\mathbf{Y}^{(2)}$. To tackle this problem, for each base partition, we introduce an orthogonal rotation matrix $\mathbf{R}^{(i)} \in \mathbb{R}^{c \times c}$ for alignment. Therefore, $\mathbf{Y}^{(i)}\mathbf{R}^{(i)}$ is the aligned embedding of the i-th base clustering result, and then we will ensemble these aligned embeddings to obtain a consensus embedding $\mathbf{H} \in \mathbb{R}^{n \times c}$.

Since the quality of each base partition is different from others, we impose a weight on each base clustering. Intuitively, more reliable base results should have larger weights, which means they will contribute more to ensemble learning. More formally, for the i-th base result, we use $0 \leq \alpha_i \leq 1$ as its weight, and then we do imputation and ensemble by minimizing the following objective function:

$$\min_{\mathbf{H}, \mathbf{Y}_u^{(i)}, \mathbf{R}^{(i)}, \alpha_i} \left\| \mathbf{H} - \sum_{i=1}^{m} \alpha_i \mathbf{Y}^{(i)} \mathbf{R}^{(i)} \right\|_F^2 ,$$

$$s.t.\quad \mathbf{Y}_u^{(i)} \in \{0,1\}^{n_u^i \times c},\ \sum_{q=1}^{c} (Y_u^{(i)})_{pq} = 1,$$
$$\mathbf{R}^{(i)T}\mathbf{R}^{(i)} = \mathbf{I},\ \mathbf{H}^T\mathbf{H} = \mathbf{I},\quad 0 \le \alpha_i \le 1,\ \sum_{i=1}^{m} \alpha_i = 1, \tag{1}$$

where the constraints on $\mathbf{Y}_u^{(i)}$ make sure that in each row of $\mathbf{Y}_u^{(i)}$, there is only one 1 and other elements are 0's. Notice that each column of $\mathbf{H}$ is a representation of a cluster. In the conventional clustering setting, since each instance often belongs to only one cluster, clusters should be far away from each other. To achieve this, we impose an orthogonal constraint on $\mathbf{H}$ as spectral clustering does.

Although the objective function in (1) seems simple, it can leverage the consensus and diversity properties well. To see this, when imputing $\mathbf{Y}^{(i)}$ and learning the consensus $\mathbf{H}$, we have

$$\begin{aligned}
&\min_{\mathbf{Y}_u^{(i)},\mathbf{H}} \left\| \mathbf{H} - \sum_{i=1}^{m} \alpha_i \mathbf{Y}^{(i)}\mathbf{R}^{(i)} \right\|_F^2 \\
= &\min_{\mathbf{Y}_u^{(i)},\mathbf{H}} tr(\mathbf{H}^T\mathbf{H}) - 2tr\left(\mathbf{H}^T \sum_{i=1}^{m} \alpha_i \mathbf{Y}^{(i)}\mathbf{R}^{(i)}\right) \\
&+ \sum_{i,j=1}^{m} \alpha_i\alpha_j tr(\mathbf{R}^{(i)T}\mathbf{Y}^{(i)T}\mathbf{Y}^{(j)}\mathbf{R}^{(j)}) \\
= &\min_{\mathbf{Y}_u^{(i)},\mathbf{H}} -2tr\left(\mathbf{H}^T \sum_{i=1}^{m} \alpha_i \mathbf{Y}^{(i)}\mathbf{R}^{(i)}\right) \\
&+ \sum_{i=1}^{m} \alpha_i^2 tr\left(\mathbf{R}^{(i)T}\mathbf{Y}^{(i)T}\mathbf{Y}^{(i)}\mathbf{R}^{(i)}\right) \\
&+ \sum_{i\neq j} \alpha_i\alpha_j tr\left(\mathbf{R}^{(i)T}\mathbf{Y}^{(i)T}\mathbf{Y}^{(j)}\mathbf{R}^{(j)}\right) \\
= &\min_{\mathbf{Y}_u^{(i)},\mathbf{H}} \underbrace{-2tr\left(\mathbf{H}^T \sum_{i=1}^{m} \alpha_i \mathbf{Y}^{(i)}\mathbf{R}^{(i)}\right)}_{Consensus} \\
&\underbrace{+\sum_{i\neq j} \alpha_i\alpha_j tr(\mathbf{R}^{(i)T}\mathbf{Y}^{(i)T}\mathbf{Y}^{(j)}\mathbf{R}^{(j)})}_{Diversity},
\end{aligned} \tag{2}$$

where the second equation is due to $tr(\mathbf{H}^T\mathbf{H}) = c$ and the third equation is due to that $\sum_{i=1}^{m} \alpha_i^2 tr(\mathbf{R}^{(i)T}\mathbf{Y}^{(i)T}\mathbf{Y}^{(i)}\mathbf{R}^{(i)}) = n\sum_{i=1}^{m} \alpha_i^2$, which are irrelevant with $\mathbf{Y}_u^{(i)}$ and $\mathbf{H}$.

Now take a closer look at (2). The first term is equivalent to maximize $tr(\mathbf{H}^T \sum_{i=1}^{m} \alpha_i \mathbf{Y}^{(i)}\mathbf{R}^{(i)})$, which means the consensus $\mathbf{H}$ should be consistent with multiple aligned base results $\sum_{i=1}^{m} \alpha_i \mathbf{Y}^{(i)}\mathbf{R}^{(i)}$. Therefore this term characterizes the *consensus* information. The second term is to minimize $\alpha_i\alpha_j tr(\mathbf{R}^{(i)T}\mathbf{Y}^{(i)T}\mathbf{Y}^{(j)}\mathbf{R}^{(j)})$ with different i, j. It characterizes the *diversity* information in two folds: (1) Given similar embeddings of base partitions $\mathbf{Y}^{(i)}\mathbf{R}^{(i)}$ and $\mathbf{Y}^{(j)}\mathbf{R}^{(j)}$, whose inner product is large, to minimize the second term, at least one of α_i and α_j should be small, which means we mainly use at most one of them for the ensemble. (2) Given two large α_i and α_j, when imputing the base $\mathbf{Y}^{(i)}$ and $\mathbf{Y}^{(j)}$, to minimize this term, the embeddings of $\mathbf{Y}^{(i)}$ and $\mathbf{Y}^{(j)}$ should be far away from each other. Hence, (1) combines these two terms which can characterize both the consensus and diversity.

B. Self-Paced Ensemble

As introduced before, different instances have various reliability. For example, the instances which are observed in most base partitions are often more reliable than those which are missing in most base results; even for the observed instances, the ones laid in the core of a cluster are more reliable than those which are in the boundary of a cluster. Intuitively, those unreliable instances may mislead the learning model if we do not handle them carefully.

To address this issue, we plug our clustering ensemble method into a self-paced learning framework. The intuitive idea is that we use the instances for imputing and ensemble in order of their reliability. In the beginning, the model may be weak so it is easily misled by unreliable data, and thus we use reliable data for learning. With the process of learning, the model becomes increasingly stronger and has the capacity to handle some unreliable data, and thus we gradually involve them in learning. Since we do not access the original data, we can only use the base clustering results themselves to evaluate the reliability. Here, we characterize the reliability with consistency, i.e., one instance is reliable if its base results are consistent with its consensus result. $\mathbf{H}$ is the learned consensus result and $\sum_{i=1}^{m} \alpha_i \mathbf{Y}^{(i)}\mathbf{R}^{(i)}$ is the aligned base result. In the case without other priors, the most intuitive way to evaluate the reliability of $\mathbf{H}$ is to compare it with the base result $\sum_{i=1}^{m} \alpha_i \mathbf{Y}^{(i)}\mathbf{R}^{(i)}$. If the learned results coincide with the observed base results, we believe the results with high confidence, and thus we think the learned results are reliable. To achieve this, we introduce a weight vector $\mathbf{v} \in [0,1]^n$ to indicate the reliability of each instance. Then, by denoting diagonal matrix $\mathbf{V} = diag(\mathbf{v})$, we extend (1) to the following self-paced form:

$$\min_{\mathbf{H},\mathbf{Y}_u^{(i)},\mathbf{R}^{(i)},\alpha_i,\mathbf{v}} \left\| \mathbf{V}\left(\mathbf{H} - \sum_{i=1}^{m} \alpha_i \mathbf{Y}^{(i)}\mathbf{R}^{(i)}\right) \right\|_F^2 - \lambda\|\mathbf{v}\|_1,$$
$$s.t.\quad \mathbf{Y}_u^{(i)} \in \{0,1\}^{n_u^i \times c},\quad \sum_{q=1}^{c} (Y_u^{(i)})_{pq} = 1,$$
$$\mathbf{R}^{(i)T}\mathbf{R}^{(i)} = \mathbf{I},\quad \mathbf{H}^T\mathbf{H} = \mathbf{I},$$
$$0 \le v_p \le 1,\quad 0 \le \alpha_i \le 1,\quad \sum_{i=1}^{m} \alpha_i = 1, \tag{3}$$

where v_p is the p-th element in vector $\mathbf{v}$, and $-\lambda\|\mathbf{v}\|_1$ is a self-paced regularized term as suggested by [32], [52], [53]. $\lambda > 0$ is an adaptive "age" parameter, which increases with the process of learning. With the increase of λ, the instances will become more and more reliable.

C. Final Objective Function

By minimizing (3), we can learn a consensus orthogonal embedding $\mathbf{H}$. To obtain a final clustering result $\mathbf{Y}$, we need some postprocessing like kmeans to discretize $\mathbf{H}$. However, in this two-stage clustering strategy, the ensemble learning and clustering are separated, so that the two processes cannot be boosted by each other to achieve the optimal goal. To address this issue, inspired by spectral rotation [54], we add a rotation term $\|\mathbf{Y}-\mathbf{HR}\|_F^2$ on (3) to discretize $\mathbf{H}$, where $\mathbf{Y} \in \{0,1\}^{n\times c}$ is the discrete cluster indicator matrix and $\mathbf{R} \in \mathbb{R}^{c\times c}$ is an orthogonal rotation matrix. Therefore, the final objective function of PCE is

$$\begin{aligned}
&\min_{\boldsymbol{\theta}} \left\| \mathbf{V}\left(\mathbf{H}-\sum_{i=1}^{m}\alpha_i \mathbf{Y}^{(i)}\mathbf{R}^{(i)}\right)\right\|_F^2 - \lambda\|\mathbf{v}\|_1 + \gamma\|\mathbf{Y}-\mathbf{HR}\|_F^2,\\
&s.t.\quad \mathbf{Y}_u^{(i)} \in \{0,1\}^{n_u^i\times c},\quad \sum_{q=1}^{c}(Y_u^{(i)})_{pq}=1,\quad \mathbf{H}^T\mathbf{H}=\mathbf{I},\\
&\qquad \mathbf{R}^{(i)T}\mathbf{R}^{(i)}=\mathbf{I},\quad \mathbf{Y}\in\{0,1\}^{n\times c},\quad \sum_{q=1}^{c}Y_{pq}=1,\\
&\qquad 0\le\alpha_i\le 1,\quad \sum\nolimits_{i=1}^{m}\alpha_i=1,\quad 0\le v_p\le 1,\quad \mathbf{R}^T\mathbf{R}=\mathbf{I},
\end{aligned} \tag{4}$$

where $\boldsymbol{\theta}=\{\mathbf{H},\mathbf{Y}_u^{(i)},\mathbf{R}^{(i)},\alpha_i,\mathbf{v},\mathbf{Y},\mathbf{R}\}$ is the set of all learned parameters, and γ is a balanced hyper-parameter. It can be seen that (4) has the following four characters:

- It imputes and ensembles base partitions $\mathbf{Y}^{(i)}$ directly without accessing the original features of data.
- It simultaneously does imputation and ensemble by fully considering the consensus and diversity.
- It ensembles data by considering its reliability with self-paced learning.
- It directly obtains the final consensus partition $\mathbf{Y}$ without any uncertain postprocessing.

D. Optimization

We apply a block coordinate descent method to optimize (4), i.e., we optimize one variable while fixing other variables.

1) Optimizing $\mathbf{v}$*:* When fixing other variables, we rewrite (4) as follows:

$$\min_{\mathbf{0}\le\mathbf{v}\le\mathbf{1}} \quad \|diag(\mathbf{v})\mathbf{A}\|_F^2 - \lambda\|\mathbf{v}\|_1, \tag{5}$$

where $\mathbf{A}=\mathbf{H}-\sum_{i=1}^{m}\alpha_i\mathbf{Y}^{(i)}\mathbf{R}^{(i)}$. By setting its partial derivative w.r.t. $\mathbf{v}$ to zero, we obtain its closed-form solution:

$$v_p = \min\left(\frac{\lambda}{2\|\mathbf{A}_{p.}\|_2^2},1\right), \tag{6}$$

where v_p is the p-th element of $\mathbf{v}$ and $\mathbf{A}_{p.}$ is the p-th row vector of $\mathbf{A}$. Notice that $\mathbf{A}$ indicates the difference between the consensus embedding $\mathbf{H}$ and the aligned base embeddings $\sum_{i=1}^{m}\alpha_i\mathbf{Y}^{(i)}\mathbf{R}^{(i)}$. Smaller $\|\mathbf{A}_{p.}\|_2^2$, which means the consensus embedding of the p-th instance is more consistent with base partitions and thus is more reliable, will have a larger weight v_p. Due to the self-paced learning, these reliable instances will contribute more in the ensemble learning to obtain a relatively reliable consensus result, and then the reliable consensus result can guide the imputation of the missing values. Moreover, v_p is proportional to λ, which means with the learning process (i.e., λ increases), the weights of instances become increasingly larger, i.e., more and more instances will be involved in learning. This is consistent with the motivation of self-paced learning.

2) Optimizing $\mathbf{H}$*:* The $\mathbf{H}$-subproblem can be reformulated as follows:

$$\begin{aligned}
&\min_{\mathbf{H}} \quad tr(\mathbf{H}^T\mathbf{DH}) - 2tr(\mathbf{H}^T\mathbf{C}),\\
&s.t.\quad \mathbf{H}^T\mathbf{H}=\mathbf{I},
\end{aligned} \tag{7}$$

where $\mathbf{D}=\mathbf{V}^2+\gamma\mathbf{I}$ and $\mathbf{C}=\gamma\mathbf{YR}^T+\mathbf{V}^2\sum_{i=1}^{m}\alpha_i\mathbf{Y}^{(i)}\mathbf{R}^{(i)}$. It is a quadratic optimization on the Stiefel manifold and can be solved by an iterative method. Let $\mathbf{H}^t$ denote the value of $\mathbf{H}$ in the t-th iteration. Given a step size $\eta>0$, we denote $\mathbf{M}=[\eta(\mathbf{DH}^t-\mathbf{C}),-\eta\mathbf{H}^t]$ and $\mathbf{N}=[\mathbf{H}^t,\mathbf{DH}^t-\mathbf{C}]^T$. The following Theorem provides an updated formula of $\mathbf{H}^{t+1}$:

Theorem 1: Suppose $\mathbf{H}^t$, $\mathbf{M}$ and $\mathbf{N}$ be defined as before, if $\mathbf{H}^{tT}\mathbf{H}^t=\mathbf{I}$, update $\mathbf{H}^{t+1}$ as follows:

$$\mathbf{H}^{t+1}=\mathbf{H}^t-\mathbf{MNH}^t-\mathbf{M}(\mathbf{I}+\mathbf{NM})^{-1}(\mathbf{NH}^t-\mathbf{NMNH}^t). \tag{8}$$

Then, ${\mathbf{H}^{t+1}}^T\mathbf{H}^{t+1}=\mathbf{I}$, and this updating is in a descent direction of (7). Since (7) has a lower bound, the iteration converges. Moreover, it can converge to a stable point.

Proof: See Appendix A, available online.

Theorem 1 shows that updating $\mathbf{H}$ by (8) can converge to a stable point. In our implementation, we set the step size η by a curvilinear search method as was done in [23], [55] for fast convergence. Notice that $\mathbf{M}$ is an n-by-$2c$ matrix and $\mathbf{N}$ is a $2c$-by-n matrix, the time complexity of computing (8) in each iteration is $O(nc^2+c^3)$, which is linear with n.

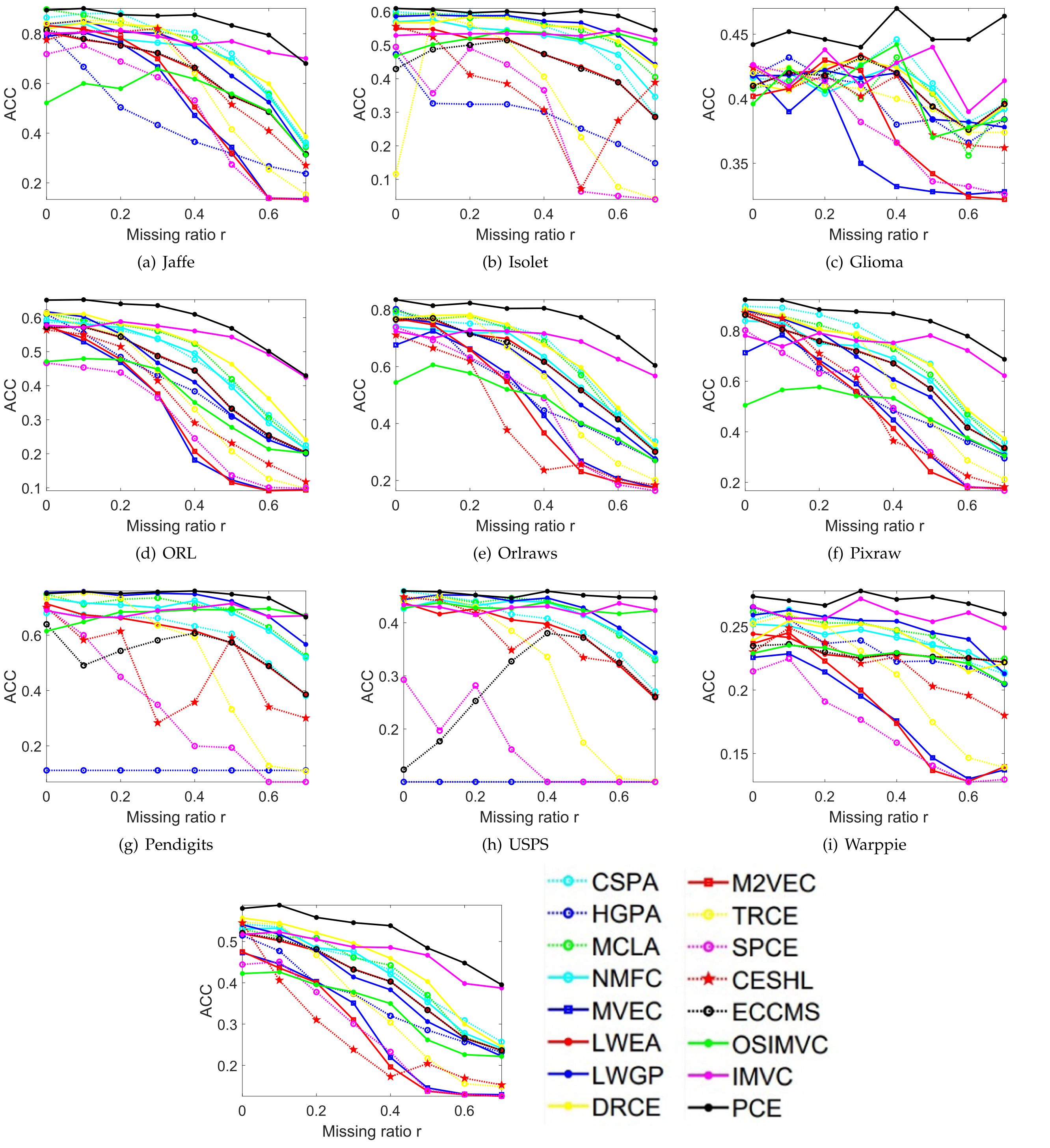
3) Optimizing $\mathbf{Y}_u^{(i)}$*:* The subproblem w.r.t. $\mathbf{Y}_u^{(i)}$ is:

$$\begin{aligned}
&\min_{\mathbf{Y}_u^{(i)}} \quad \frac{1}{2}\alpha_i tr(\mathbf{R}^{(i)T}\mathbf{Y}^{(i)T}\mathbf{V}^2\mathbf{Y}^{(i)}\mathbf{R}^{(i)}) - tr(\mathbf{H}^T\mathbf{V}^2\mathbf{Y}^{(i)}\mathbf{R}^{(i)})\\
&\qquad + \sum_{j:j\neq i}\alpha_j tr(\mathbf{R}^{(j)T}\mathbf{Y}^{(j)T}\mathbf{V}^2\mathbf{Y}^{(i)}\mathbf{R}^{(i)}),\\
&s.t.\quad \mathbf{Y}_u^{(i)}\in\{0,1\}^{n_u^i\times c},\quad \sum_{q=1}^{c}(Y_u^{(i)})_{pq}=1.
\end{aligned} \tag{9}$$

Taking a closer look at the first term, we have

$$\begin{aligned}
tr(\mathbf{R}^{(i)T}\mathbf{Y}^{(i)T}\mathbf{V}^2\mathbf{Y}^{(i)}\mathbf{R}^{(i)}) &= tr(\mathbf{R}^{(i)}\mathbf{R}^{(i)T}\mathbf{Y}^{(i)T}\mathbf{V}^2\mathbf{Y}^{(i)})\\
&= tr(\mathbf{V}^2\mathbf{Y}^{(i)}\mathbf{Y}^{(i)T})\\
&= \mathbf{v}^T\mathbf{v}.
\end{aligned} \tag{10}$$

Therefore, the first term is irrelevant with $\mathbf{Y}_u^{(i)}$, and we can remove it safely. Then, denoting $\mathbf{E}=\sum_{j:j\neq i}\alpha_j\mathbf{R}^{(i)}\mathbf{R}^{(j)T}\mathbf{Y}^{(j)T}\mathbf{V}^2-\mathbf{R}^{(i)}\mathbf{H}^T\mathbf{V}^2$, (9) can be rewritten as $\min tr(\mathbf{Y}^{(i)}\mathbf{E})$. Since by some appropriate permutations $\mathbf{Y}^{(i)}$ can be written as $\mathbf{Y}^{(i)}=[\mathbf{Y}_o^{(i)T},\mathbf{Y}_u^{(i)T}]^T$,



(a) Jaffe (b) Isolet (c) Glioma

(d) ORL (e) Orlraws (f) Pixraw

(g) Pendigits (h) USPS (i) Warppie

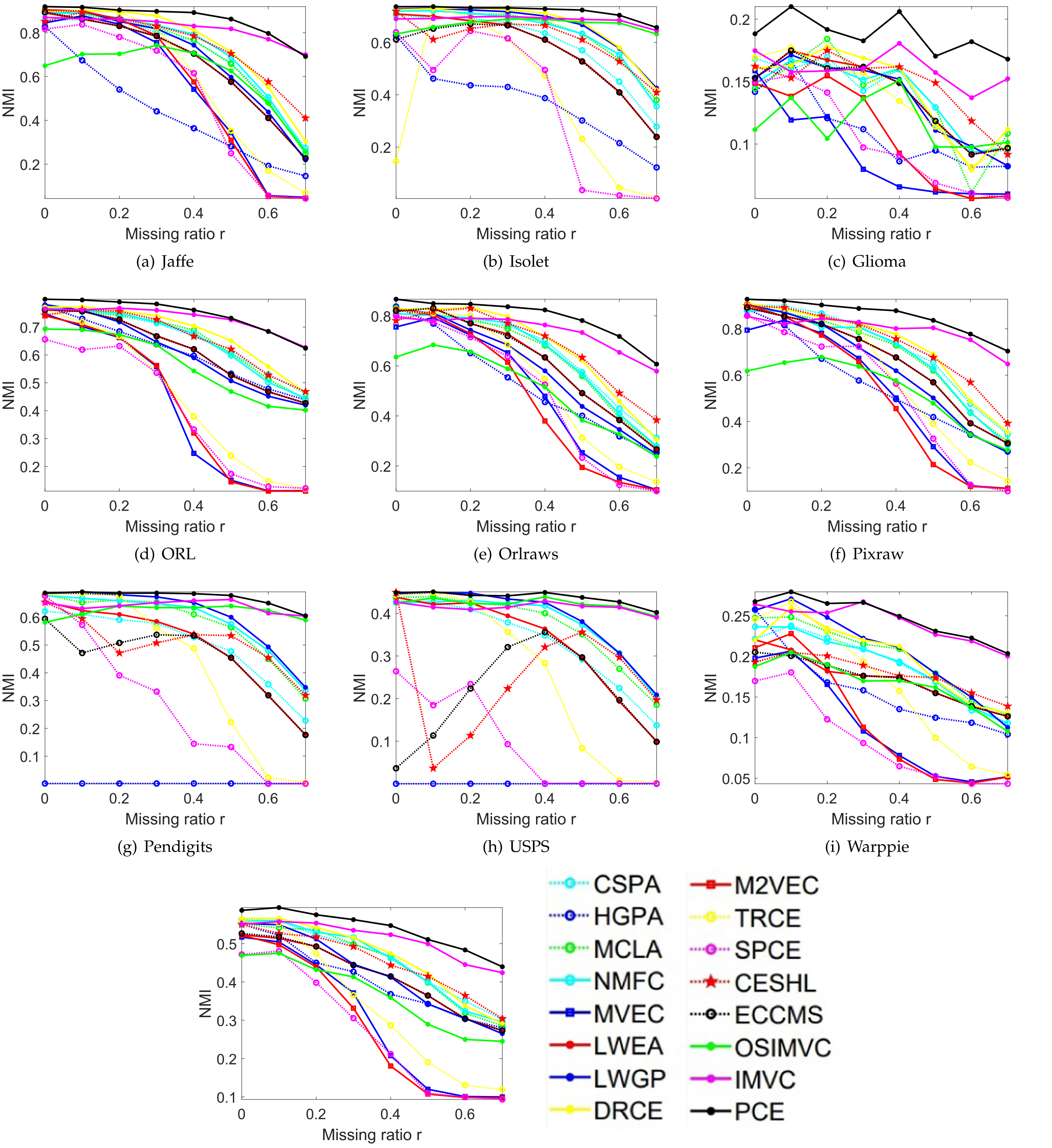
(j) Yale (k) Legend

Fig. 2. ACC results with different missing ratios on all data sets.

$\mathbf{E}$ can also be written as $\mathbf{E} = [\mathbf{E}_o, \mathbf{E}_u]$ correspondingly. Then, (9) can be written as:

$$\min_{\mathbf{Y}_u^{(i)}} \quad tr(\mathbf{Y}_u^{(i)}\mathbf{E}_u),$$
$$s.t. \quad \mathbf{Y}_u^{(i)} \in \{0,1\}^{n_u^i \times c}, \quad \sum\nolimits_{q=1}^{c} (Y_u^{(i)})_{pq} = 1. \qquad (11)$$

Equation (11) can be decoupled into n_u^i independent subproblems, where each row is one of the subproblems. For the p-th subproblem, we can obtain its global optima by finding the smallest element in the p-th column of $\mathbf{E}_u$. In more detail, supposing $r = \text{argmin}_q (E_u)_{qp}$, we set the p-th row of $\mathbf{Y}_u^{(i)}$ as $(Y_u^{(i)})_{pr} = 1$ and other $(Y_u^{(i)})_{pq} = 0$ where $q \neq r$.



(a) Jaffe (b) Isolet (c) Glioma

(d) ORL (e) Orlraws (f) Pixraw

(g) Pendigits (h) USPS (i) Warppie

(j) Yale (k) Legend

Fig. 3. NMI results with different missing ratios on all data sets.

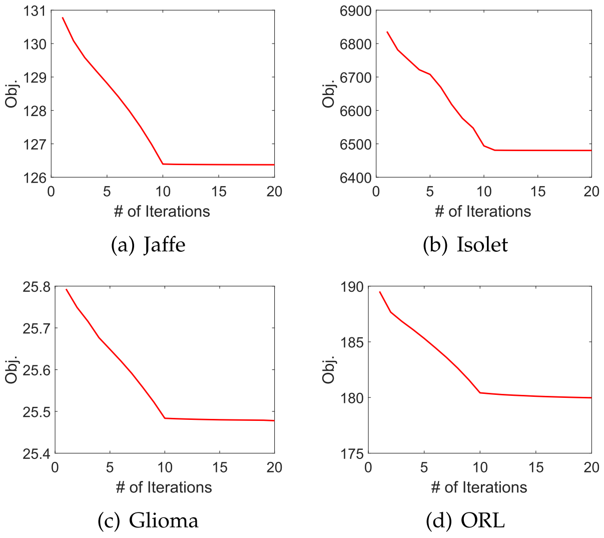
4) Optimizing α_i: When fixing other variables, we obtain the subproblem w.r.t. $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m]$ as:

$$\min_{\boldsymbol{\alpha}} \quad \boldsymbol{\alpha}^T \mathbf{G} \boldsymbol{\alpha} - 2\mathbf{f}^T \boldsymbol{\alpha},$$
$$s.t. \quad 0 \le \alpha_i \le 1, \quad \sum_{i=1}^{m} \alpha_i = 1, \qquad (12)$$

where the (i,j)-th element of $\mathbf{G}$ is $G_{ij} = tr(\mathbf{R}^{(i)T}\mathbf{Y}^{(i)T}\mathbf{V}^2\mathbf{Y}^{(j)}\mathbf{R}^{(j)})$ and the i-th element of vector $\mathbf{f}$ is $f_i = tr(\mathbf{R}^{(i)T}\mathbf{Y}^{(i)T}\mathbf{V}^2\mathbf{H})$. Then, we have the following Theorem about its convexity:

Theorem 2: Equation (12) is a convex quadratic programming.

Proof: See Appendix B, available online.



(a) Jaffe (b) Isolet (c) Glioma (d) ORL

Fig. 4. Convergence curves on Jaffe, Isolet, Glioma, and ORL.

Since (12) is convex, the global optima can be found by some standard methods. In our implementation, we use the *quadprog* function provided by Matlab.

5) Optimizing $\mathbf{R}^{(i)}$*:* Similar to the derivation of (9), we can obtain the $\mathbf{R}^{(i)}$-subproblem:

$$\min_{\mathbf{R}^{(i)}} \quad tr(\mathbf{K}\mathbf{R}^{(i)}),$$
$$s.t. \quad \mathbf{R}^{(i)T}\mathbf{R}^{(i)} = \mathbf{I}, \tag{13}$$

where $\mathbf{K} = \sum_{j:j\neq i} \alpha_j \mathbf{R}^{(j)T}\mathbf{Y}^{(j)T}\mathbf{V}^2\mathbf{Y}^{(i)} - \mathbf{H}^T\mathbf{V}^2\mathbf{Y}^{(i)}$.

The following Theorem provides its global optima:

Theorem 3: Supposing the singular value decomposition (SVD) of $-\mathbf{K}^T$ is $-\mathbf{K}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{S}^T$, then the global optima of (13) is $\mathbf{R}^{(i)} = \mathbf{U}\mathbf{S}^T$.

Proof: See Appendix C, available online.

6) Optimizing $\mathbf{R}$*:* When solving $\mathbf{R}$, we have

$$\min_{\mathbf{R}^{(i)}} \quad tr(\mathbf{K}'\mathbf{R}^{(i)}),$$
$$s.t. \quad \mathbf{R}^{(i)T}\mathbf{R}^{(i)} = \mathbf{I}, \tag{14}$$

where $\mathbf{K}' = -\mathbf{Y}^T\mathbf{H}$. Obviously, it can also be solved by Theorem 3.

7) Optimizing $\mathbf{Y}$*:* The $\mathbf{Y}$-subproblem is:

$$\min_{\mathbf{Y}} \quad \|\mathbf{Y} - \mathbf{H}\mathbf{R}\|_F^2,$$
$$s.t. \quad \mathbf{Y} \in \{0,1\}^{n\times c}, \quad \sum_{q=1}^{c} Y_{pq} = 1. \tag{15}$$

It can be decoupled into n independent subproblems by rows. Denote $\mathbf{L} = \mathbf{H}\mathbf{R}$. When solving the p-th row of $\mathbf{Y}$, we first find the largest element in the p-th row of $\mathbf{L}$, i.e., $r = \text{argmax}_q L_{pq}$, and set $Y_{pr} = 1$ and $Y_{pq} = 0$ where $q \neq r$.

Algorithm 1: Partial Clustering Ensemble.

```
Input: Partial base partitions $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(m)}$,
  hyper-parameter $\gamma$, number of clusters $c$.
Output: Final consensus clustering result $\mathbf{Y}$.
Initialize the parameters.
while not convergence do
  Compute $\mathbf{v}$ by (6).
  Compute $\mathbf{H}$ by Theorem 1.
  Compute $\mathbf{Y}_u^{(1)}, \cdots \mathbf{Y}_u^{(m)}$ by solving (11).
  Compute $\boldsymbol{\alpha}$ by solving (12).
  Compute $\mathbf{R}^{(1)}, \cdots \mathbf{R}^{(m)}$ by Theorem 3.
  Compute $\mathbf{R}$ by Theorem 3.
  Compute $\mathbf{Y}$ by solving (15).
  Update $\lambda \leftarrow \lambda * 1.1$ in the first 10 iterations.
end while
```

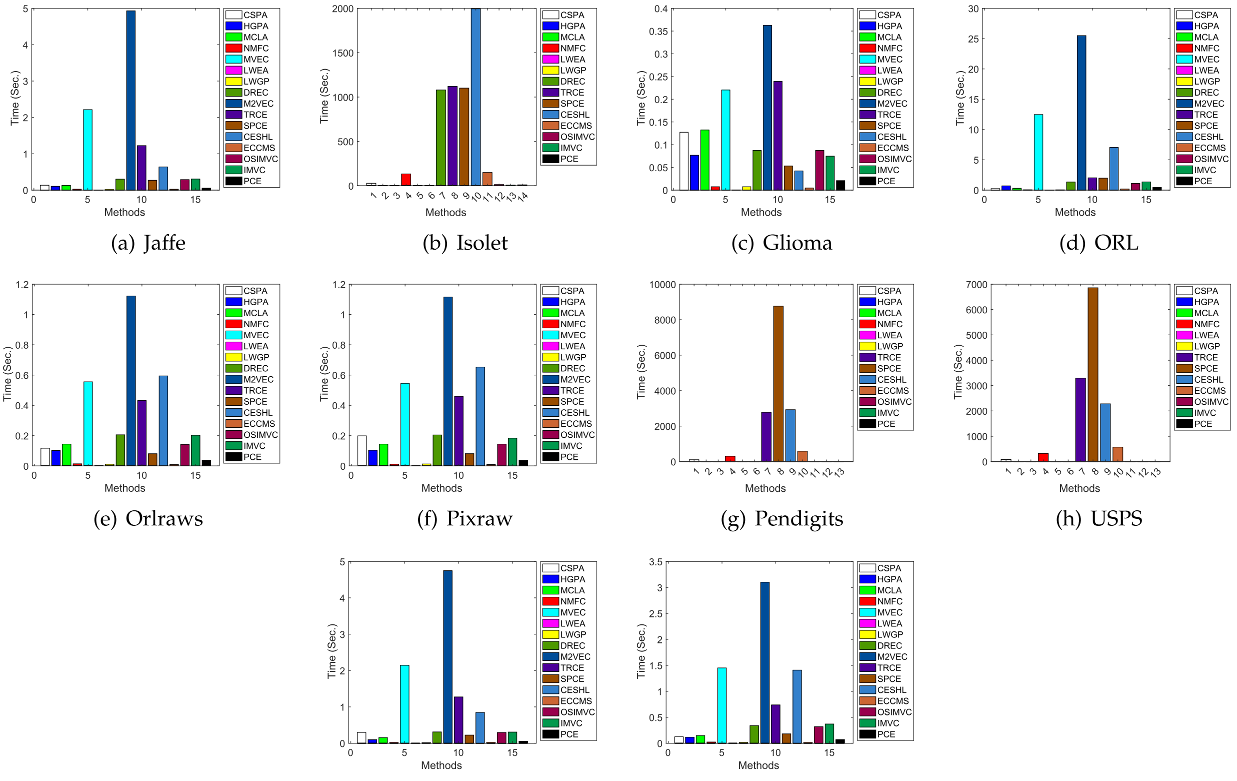
E. Algorithm and Discussion

We first introduce the initializations and parameter settings. Given m base partitions, we initialize $\alpha_i = \frac{1}{m}$, i.e., each base result has the same initial weight. All $\mathbf{Y}_u^{(i)}$'s are initialized as $\mathbf{Y}_u^{(i)} = \mathbf{0}$, which means the unobserved base results are initialized to zero. All $\mathbf{R}^{(i)}$'s are initialized as $\mathbf{R}^{(i)} = \mathbf{I}$, which means the initial rotation is an identity transformation. We compute $\mathbf{H} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{Y}^{(i)}\mathbf{R}^{(i)}$ to make it be an equal-weighted linear combination of the aligned embedding and then we orthogonalize it to meet the constraint. We initialize $\mathbf{Y}$ and $\mathbf{R}$ directly by doing a spectral rotation on $\mathbf{H}$.

When setting λ, since it is relevant with the reliability $\mathbf{v}$, we should initialize it more carefully. According to (6), we first compute all $2\|A_{p.}\|_2^2$ and sort them by ascending order, then we initialize λ as the first ten quantile value of $2\|A_{p.}\|_2^2$. It means that for the top 10% most reliable instances, the weights are set to 1, i.e., they are completely used for learning. Then after each iteration, we increase λ as $\lambda \leftarrow \lambda * 1.1$. After 10 iterations, we do not change λ.

Algorithm 1 shows the whole process of PCE. Notice that for all subproblems except $\mathbf{H}$-subproblem, we can find the global optima. For $\mathbf{H}$-subproblem, according to Theorem 1, we can find its stable point which also makes the objective function decrease. Moreover, the objective function has a lower bound and thus Algorithm 1 always converges. In fact, it often converges very fast.

Now we analyze the time and space complexity. Since we need to save m n-by-c matrices (i.e., $\mathbf{Y}^{(i)}$) and m c-by-c matrices (i.e., $\mathbf{R}^{(i)}$), the space complexity is $O(mnc + mc^2)$. Solving $\mathbf{v}$ costs $O(nc^2)$ time. When solving $\mathbf{H}$, as introduced in Appendix, available online, in each iteration, it costs $O(nc^2 + c^3)$ time. The time complexity of computing $\mathbf{Y}_u^{(1)}, \ldots, \mathbf{Y}_u^{(m)}$ is $O(mnc^2)$. Solving the convex quadratic programming w.r.t. $\boldsymbol{\alpha}$ costs $O(m^3)$ time. When optimizing $\mathbf{R}^{(1)} \cdots, \mathbf{R}^{(m)}$ and $\mathbf{R}$, we need SVD on $m+1$ c-by-c matrix, which costs $O(mc^3)$ time. Solving $\mathbf{Y}$ also costs $O(nc^2)$ time. Therefore, the whole time complexity is $O(mnc^2 + mc^3 + m^3)$. Notice that in practice it often happens that $c, m \ll n$, and thus the time complexity is linear



(a) Jaffe (b) Isolet (c) Glioma (d) ORL

(e) Orlraws (f) Pixraw (g) Pendigits (h) USPS

(i) Warppie (j) Yale

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

with n. However, the time complexity of many state-of-the-art clustering ensemble methods is often square or even cubic in the number of instances, which is less efficient than the proposed PCE.

IV. Experiments

In this section, we compare PCE with other clustering ensemble methods on benchmark data sets.

A. Data Sets

We conduct experiments on 10 public benchmark data sets, including Isolet [56], Jaffe [57], Glioma [58], ORL [59], Orlraws,[1] Pixraw[1], Pendigits [60], USPS [61], Warppie[1] and Yale [62]. The information of these data sets is shown in Table I.

To show the effectiveness in the real incomplete data scenario, we also conduct experiments on a real incomplete data set 3Sources.[2] 3Sources contains 416 news stories on 6 topics from three online news sources: BBC, Reuters, and Guardian. Each source contains missing data. In more detail, among the 416 instances, 169 instances are shared by three sources, 194 instances are only shared by two sources, and 53 instances appear only in one source.

TABLE I
Description of the Data Sets

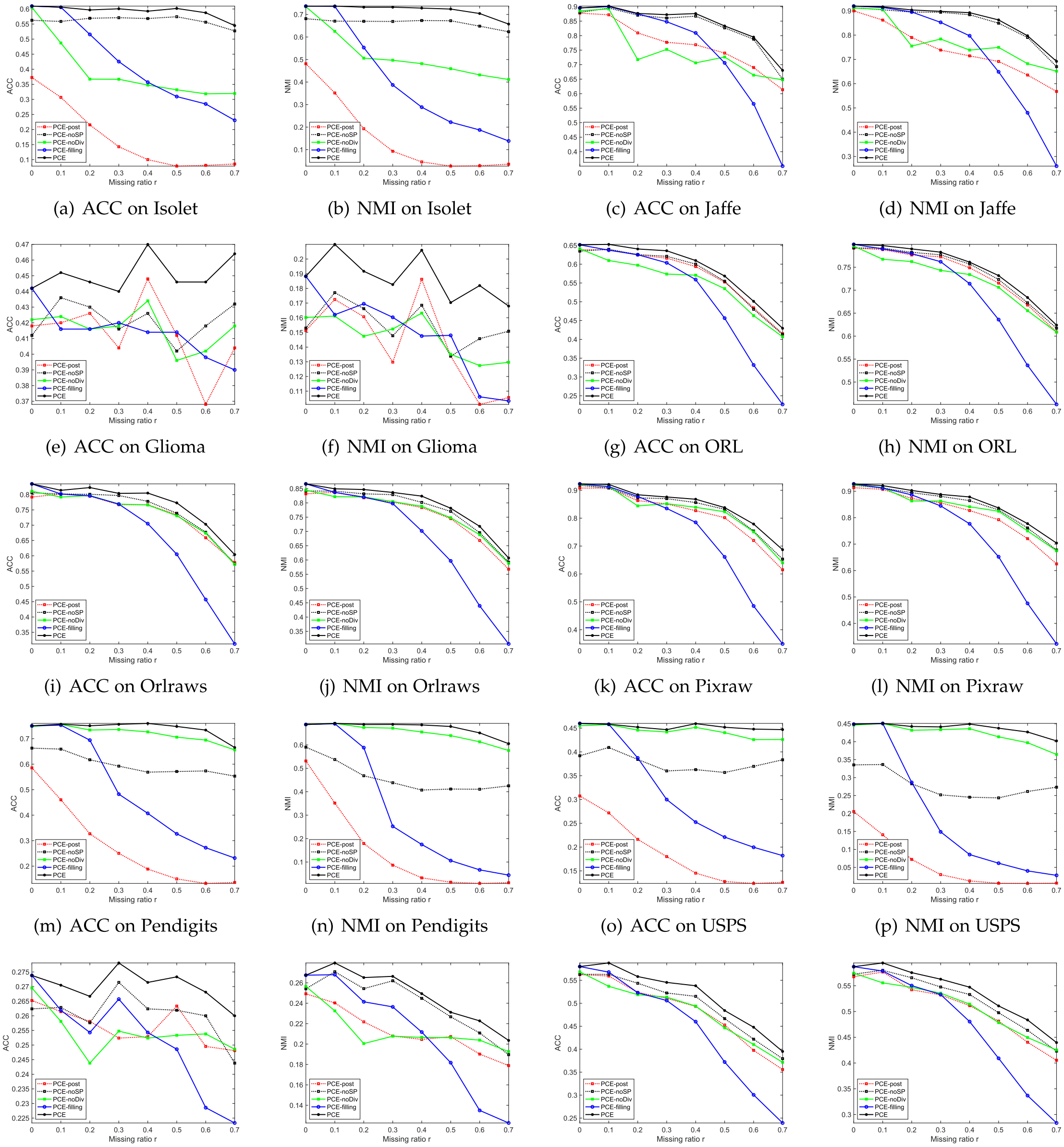
	#instances	#features	#classes
Jaffe	213	676	10
Isolet	7797	617	26
Glioma	50	4434	4
ORL	400	1024	40
Orlraws	100	10304	10
Pixraw	100	10000	10
Pendigits	10992	16	10
USPS	11000	256	10
Warppie	210	2420	10
Yale	165	1024	15

B. Experimental Setup

We compare with the following clustering ensemble methods:

- *CSPA* [1], which is a cluster-based similarity partitioning algorithm for clustering ensemble.
- *HGPA* [1], which is hyper-graph partitioning based algorithm for clustering ensemble.
- *MCLA* [1], which is a meta-clustering algorithm, which transforms the clustering ensemble task to a cluster correspondence problem.
- *NMFC* [63], which is a non-negative matrix factorization based clustering ensemble method.

[1] https://jundongl.github.io/scikit-feature/datasets.html
[2] http://mlg.ucd.ie/datasets/3sources.html



(a) ACC on Isolet (b) NMI on Isolet (c) ACC on Jaffe (d) NMI on Jaffe

(e) ACC on Glioma (f) NMI on Glioma (g) ACC on ORL (h) NMI on ORL

(i) ACC on Orlraws (j) NMI on Orlraws (k) ACC on Pixraw (l) NMI on Pixraw

(m) ACC on Pendigits (n) NMI on Pendigits (o) ACC on USPS (p) NMI on USPS

(q) ACC on Warppie (r) NMI on Warppie (s) ACC on Yale (t) NMI on Yale

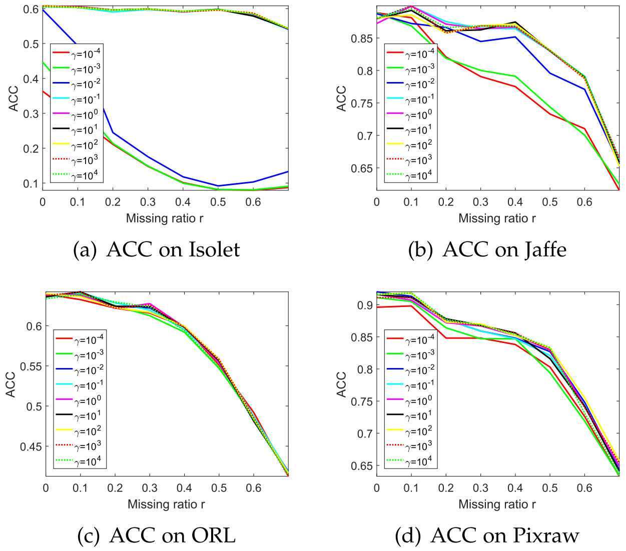
Fig. 6. Ablation study.

- *MVEC* [12], which is a clustering ensemble method with multi-view learning.
- *LWEA* [37], which is a locally weighted evidence accumulation method for clustering ensmeble.
- *LWGP* [37], which is a locally weighted graph partitioning method for clustering ensemble.
- *DREC* [64], which is a clustering ensemble method with dense representation learning.
- *M2VEC* [13], which is a marginalized multiview clustering ensemble method with deep learning.
- *TRCE* [33], which is a tri-level robust clustering ensemble method with multiple graph learning.
- *SPCE* [32], which is a self-paced clustering ensemble method with multiple graph learning.
- *CESHL* [36], which is a clustering ensemble with structured hypergraph learning.
- *ECCMS* [65], which is a clustering ensemble method with an enhanced co-association matrix.

To show the effectiveness of handling incomplete data, we also compare with the state-of-the-art incomplete late fusion methods for multi-view clustering:

- *OSIMVC* [20], which is a one-stage clustering method to ensemble the embedding of each view.
- *IMVC* [19], which is an alignment method to ensemble multiple results of kernel kmeans.

For all data sets except the real incomplete data set 3Sources, given a missing ratio $r\% \in \{0, 10\%, \ldots, 70\%\}$, we randomly remove $r\%$ instances and run kmeans on the remaining to obtain



(a) ACC on Isolet (b) ACC on Jaffe

(c) ACC on ORL (d) ACC on Pixraw

Fig. 7. ACC w.r.t. γ on Isolet, Jaffe, ORL, and Pixraws.

one partial base partition. $r = 0$ means the data are complete without missing instances. We repeat this process 10 times to obtain the partial $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(10)}$. For our PCE and the incomplete late fusion methods IMVC and OSIMVC, we directly ensemble the partial $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(10)}$ to learn a consensus clustering result. For the remaining conventional clustering ensemble methods, we first impute $\mathbf{Y}^{(i)}$ and run them on the filled base partitions. Since $\mathbf{Y}^{(i)}$ has a special discrete structure as introduced before, some imputation methods like zero filling and mean value filling may be inappropriate, we use the random filling, i.e., for a missing instance, we randomly put it into one of the c clusters. On all data sets, for all methods, we set c as the true number of classes. For the real partial data set 3Sources, we run kmeans on the data of each source to obtain 3 base clustering results, and ensemble the 3 partial base results. Other settings are similar to that of other data sets. For our method, we tune γ in the range $[10^{-4}, 10^{4}]$. We use Accuracy (ACC) and Normalized Mutual Information (NMI) to evaluate the clustering performance. We repeat the experiments with different base partitions 10 times and report the average results.

C. Experimental Results

Figs. 2 and 3 show the ACC and NMI results of all methods with different missing ratios, respectively. Notice that due to their high time complexity, MVEC, M2VEC, and DREC cannot run a result in reasonable time on the large data sets Pendigits and USPS; MVEC and M2VEC also cannot run a result in Isolet data set. In Figs. 2 and 3, PCE is denoted as the black solid line. We can find that, on the complete data, i.e., $r = 0$, our PCE is comparable to or sometimes even better than the state-of-the-art clustering ensemble methods. The performance of some clustering ensemble methods deteriorates rapidly with the increase of the missing ratio, whereas our PCE is relatively stable when the missing ratio increases. When compared with incomplete late fusion methods, PCE still outperforms them. The reasons may be two folds: (1) when imputing the base partitions, PCE fully considers the diversity which is helpful for the clustering ensemble; (2) PCE also considers the reliability of data, which can alleviate the side effects caused by unreliable data. Although MVEC and M2VEC are robust clustering ensemble methods and also can handle incomplete data, since they are not designed specifically for the missing data, our PCE still performs better.

Notice that, although on few data sets with a large missing ratio, IMVC performs better than PCE, e.g., on the Jaffe data set with 70% missing ratio, PCE is still comparable with IMVC. IMVC is one of the state-of-the-art incomplete multi-view clustering methods with late fusion, and thus it is very competitive. Moreover, when the missing ratio is large, it means that most of the data are missing, e.g., 70% of the data are missing in this case. Most of the random filling methods fail in this extreme case. This case is very hard because we may not have enough useful clues to fill in so many missing values. Despite this, in this case, our method can still outperform the random filling methods and perform comparably with IMVC, which shows the effectiveness of our method. In the future, we will focus on this extreme case, i.e., we will further study how to handle the data with a very high missing ratio.

Although most experimental results show that the clustering performance decreases with the increase of the missing ratio, some results show that the performance may increase as the missing ratio increases, which seems strange. Notice that the inputs of clustering ensemble methods are base clustering results instead of the original data. Due to the limitation of base clustering methods, the base clustering results are imperfect and unreliable. It means that we cannot guarantee that all observed values (i.e., the observed base clustering results) are correct. Sometimes, the incorrect values in base clustering results are missing, which is helpful for our clustering ensemble instead, because these missing incorrect values will not mislead our model and our method fills them by considering all other data, which has a chance to fill in the correct values and make the performance increase.

The results on the real incomplete data 3Sources are shown in Table II. It shows that the PCE also outperforms other compared methods on the real application, which demonstrates its superiority and effectiveness.

D. Results on Efficiency

To show the efficiency of the proposed PCE, we show the convergence curves and running time in Figs. 4 and 5, respectively. Fig. 4 shows the convergence curves on Jaffe, Isolet, Glioma, and ORL data sets. Results on other data sets are similar. It shows that PCE often converges within 20 iterations, which demonstrates the claim in Section III-E. Fig. 5 shows the running time on all data sets. Notice that MVEC and M2VEC cannot run a result in a reasonable time on the large data sets Isolet, Pendigits, and USPS, and DREC cannot run a result in a reasonable time on the large data sets Pendigits and USPS. We can find that the proposed PCE is faster than most compared methods, which demonstrates its efficiency. Notice that the complexity of PCE is linear with the number of instances, whereas the complexity

TABLE II
ACC AND NMI OF KMEANS ON 3SOURCES DATA SET

Methods	CSPA [1]	HGPA [1]	MCLA [1]	NMFC [63]	MVEC [12]	LWEA [37]	LWGP [37]	DREC [64]
ACC	0.2969 ±0.0555	0.2332 ±0.0186	0.2500 ±0.0000	0.3050 ±0.0332	0.3060 ±0.0344	0.3195 ±0.0449	0.3087 ±0.0403	0.2808 ±0.0180
NMI	0.0900 ±0.0488	0.0263 ±0.0135	0.0000 ±0.0000	0.0862 ±0.0285	0.0665 ±0.0210	0.0653 ±0.0356	0.0618 ±0.0332	0.0389 ±0.0156
Methods	M2VEC [13]	TRCE [33]	SPCE [32]	CESHL [36]	ECCMS [65]	OSIMVC [20]	IMVC [19]	PCE
ACC	0.3065 ±0.0390	0.2974 ±0.0260	0.2500 ±0.0000	0.2784 ±0.0281	0.3195 ±0.0449	0.2899 ±0.0439	0.3512 ±0.0426	**0.3649** ±0.0599
NMI	0.0646 ±0.0282	0.0579 ±0.0247	0.0000 ±0.0000	0.0398 ±0.0312	0.0653 ±0.0356	0.0793 ±0.0566	0.1502 ±0.0544	**0.1824** ±0.0695

of many other clustering ensemble methods is square or cubic in the number of instances.

E. Ablation Study

For an ablation study, we compare with three degenerated versions of PCE: *PCE-post*, *PCE-noSP*, *PCE-noDiv*, and *PCE-filling*. In PCE-post, we remove the rotation term $\|\mathbf{Y}-\mathbf{HR}\|_F^2$ to learn a consensus orthogonal embedding $\mathbf{H}$ first, and then apply spectral rotation as postprocessing to discretize $\mathbf{H}$ to obtain the final clustering result, i.e., it is a two-stage model. In PCE-noSP, we remove the self-paced learning by fixing $\mathbf{v}=\mathbf{1}$ to show the effectiveness of the self-paced learning framework. PCE-noDiv is a degenerated version without diversity. In more detail, we remove the second term (i.e., the diversity term) in (2) to show the effects of the diversity. In PCE-filling, we abandon the proposed imputing strategy and fill in the missing values with random filling as other methods did.

The results of the ablation study are shown in Fig. 6. We can see that PCE often outperforms PCE-post significantly, which demonstrates the necessity and superiority of our one-stage clustering framework. Compared with the non-self-paced learning method PCE-noSP, PCE also performs better. It shows that considering the reliability of data with self-paced learning can further improve performance, which is in line with our motivation. If removing the diversity term, PCE-noDiv decreases the performance, which demonstrates the effectiveness of the diversity term. PCE also outperforms PCE-filling, which means the proposed filling method is more effective than the random filling method.

F. Hyper-Parameter Study

We show the effects of hyper-parameter γ on Isolet, Jaffe, ORL, and Pixraw in Fig. 7. The results on other data sets are similar. From Fig. 7, we find that the performance of PCE is stable when $10^{-1}\leq\gamma\leq 10^4$. Therefore, we can easily set $10^{-1}\leq\gamma\leq 10^4$ to perform relatively well. However, in some data sets like Isolet and Jaffe, the performance is much worse when γ is too small. Notice that γ is to control the discretization term. If $\gamma=0$, PCE will degenerate to the two-stage model as PCE-post. As shown in the ablation study, PCE outperforms PCE-post significantly on most data sets. When comparing the results of the hyper-parameter study and the ablation study, we can find that, when γ is too small, PCE performs like PCE-post. That is why the performance is much worse when γ is too small.

V. CONCLUSION

This paper proposed an innovative partial clustering ensemble method that directly imputed and ensembled the partial base partitions without accessing the original data. It seamlessly integrated the imputation and ensemble into a unified framework by fully considering the consensus, diversity, and reliability of data. We also designed an effective and efficient iterative algorithm to optimize the objective function, which was theoretically guaranteed to converge. At last, we conducted extensive experiments by comparing with state-of-the-art clustering ensemble methods and incomplete late fusion methods. The experimental results shew the effectiveness and superiority of the proposed method when handling incomplete data.

One limitation of the proposed method is that we need a pre-given number of clusters c, which may be difficult to determine in some real applications. In the future, we will consider how to automatically determine c. Moreover, we will further study some more complex issues in the incomplete clustering ensemble. For example, in some distributed computing scenarios, the instances in different clients may be unaligned. However, in the proposed PCE, we assume that the observed data in each base partition should be aligned. In the future, we will design new models to address this issue.

REFERENCES

[1] 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.

[2] A. Topchy, A. K. Jain, and W. F. Punch, “Combining multiple weak clusterings,” in *Proc. IEEE Int. Conf. Data Mining*, 2003, pp. 331–338.

[3] 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.

[4] X. Z. Fern and C. E. Brodley, “Solving cluster ensemble problems by bipartite graph partitioning,” in *Proc. Int. Conf. Mach. Learn.*, 2004, Art. no. 36.

[5] Z. Zhou and W. Tang, “Clusterer ensemble,” *Knowl. Based Syst.*, vol. 19, no. 1, pp. 77–83, 2006.

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

[7] Z. Tao, H. Liu, and Y. Fu, “Simultaneous clustering and ensemble,” in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1546–1552.

[8] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, “Robust spectral ensemble clustering via rank minimization,” *ACM Trans. Knowl. Discov. Data*, vol. 13, no. 1, pp. 1–25, 2019.

[9] Y. Jia, H. Liu, J. Hou, and Q. Zhang, “Clustering ensemble meets low-rank tensor approximation,” in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 7970–7978.

[10] Z. Ghahramani and M. I. Jordan, “Supervised learning from incomplete data via an EM approach,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 1993, pp. 120–127.

[11] J. Wu, H. Liu, H. Xiong, J. Cao, and J. Chen, “K-means-based consensus clustering: A unified view,” *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 1, pp. 155–169, Jan. 2015.

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

[13] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, “Marginalized multiview ensemble clustering,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 600–611, Feb. 2020.

[14] S. Li, Y. Jiang, and Z. Zhou, “Partial multi-view clustering,” in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1968–1974.

[15] C. Xu, D. Tao, and C. Xu, “Multi-view learning with incomplete views,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5812–5825, Dec. 2015.

[16] S. Bhadra, S. Kaski, and J. Rousu, “Multi-view kernel completion,” *Mach. Learn.*, vol. 106, no. 5, pp. 713–739, 2017.

[17] P. Wang and X. Chen, “Three-way ensemble clustering for incomplete data,” *IEEE Access*, vol. 8, pp. 91 855–91 864, 2020.

[18] X. Liu et al., “Efficient and effective incomplete multi-view clustering,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 4392–4399.

[19] X. Liu et al., “Efficient and effective regularized incomplete multi-view clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2634–2646, Aug. 2021.

[20] Y. Zhang, X. Liu, S. Wang, J. Liu, S. Dai, and E. Zhu, “One-stage incomplete multi-view clustering via late fusion,” in *Proc. ACM Multimedia Conf. Virtual Event*, 2021, pp. 2717–2725.

[21] X. Cai, F. Nie, and H. Huang, “Multi-view K-means clustering on big data,” in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2598–2604.

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

[23] P. Zhou, Y.-D. Shen, L. Du, F. Ye, and X. Li, “Incremental multi-view spectral clustering,” *Knowl.-Based Syst.*, vol. 174, pp. 73–86, 2019.

[24] X. Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou, “COMIC: Multi-view clustering without parameter selection,” in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 5092–5101.

[25] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu, “Large-scale multi-view subspace clustering in linear time,” in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 4412–4419.

[26] S. Wang et al., “Fast parameter-free multi-view subspace clustering with consensus anchor guidance,” *IEEE Trans. Image Process.*, vol. 31, no. 31, pp. 556–568, 2022.

[27] W. Liang et al., “Multi-view spectral clustering with high-order optimal neighborhood Laplacian matrix,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 7, pp. 3418–3430, Jul. 2022.

[28] 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, Dec. 2011.

[29] 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, Mar. 2012.

[30] Z. Tao, H. Liu, S. Li, and Y. Fu, “Robust spectral ensemble clustering,” in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 367–376.

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

[32] P. Zhou, L. Du, X. Liu, Y. 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.

[33] P. Zhou, L. Du, Y.-D. Shen, and X. Li, “Tri-level robust clustering ensemble with multiple graph learning,” in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 11 125–11 133.

[34] P. Zhou, L. Du, and X. Li, “Adaptive consensus clustering for multiple K-means via base results refining,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 10, pp. 10251–10264, Oct. 2023.

[35] P. Zhou, B. Sun, X. Liu, L. Du, and X. Li, “Active clustering ensemble with self-paced learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2023, doi: 10.1109/TNNLS.2023.3252586.

[36] P. Zhou, X. Wang, L. Du, and X. Li, “Clustering ensemble via structured hypergraph learning,” *Inf. Fusion*, vol. 78, pp. 171–179, 2022.

[37] 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.

[38] P. Zhou, L. Du, and X. Li, “Self-paced consensus clustering with bipartite graph,” in *Proc. 29th Int. Joint Conf. Artif. Intell.*, C. Bessiere Ed., 2020, pp. 2133–2139.

[39] P. Zhou, X. Liu, L. Du, and X. Li, “Self-paced adaptive bipartite graph learning for consensus clustering,” *ACM Trans. Knowl. Discov. Data*, vol. 17, no. 5, pp. 62:1–62:35, 2023.

[40] 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, 2020.

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

[42] 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.

[43] J. Wen et al., “A survey on incomplete multiview clustering,” *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 53, no. 2, pp. 1136–1149, Feb. 2023.

[44] H. Zhao, H. Liu, and Y. Fu, “Incomplete multi-modal visual data grouping,” in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2392–2398.

[45] N. Xu, Y. Guo, X. Zheng, Q. Wang, and X. Luo, “Partial multi-view subspace clustering,” in *Proc. ACM Multimedia Conf. Multimedia Conf.*, 2018, pp. 1794–1801.

[46] X. Zhu et al., “Localized incomplete multiple kernel k-means,” in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3271–3277.

[47] X. Liu et al., “Multiple kernel k-means with incomplete kernels,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1191–1204, May 2020.

[48] X. Liu, “Incomplete multiple kernel alignment maximization for clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021, doi: 10.1109/TPAMI.2021.3116948.

[49] Q. Wang, Z. Ding, Z. Tao, Q. Gao, and Y. Fu, “Partial multi-view clustering via consistent GAN,” in *Proc. IEEE Int. Conf. Data Mining*, 2018, pp. 1290–1295.

[50] C. Xu, Z. Guan, W. Zhao, H. Wu, Y. Niu, and B. Ling, “Adversarial incomplete multi-view clustering,” in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 3933–3939.

[51] C. Zhang, Y. Cui, Z. Han, J. T. Zhou, H. Fu, and Q. Hu, “Deep partial multi-view learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2402–2415, May 2022.

[52] M. P. Kumar, B. Packer, and D. Koller, “Self-paced learning for latent variable models,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 1189–1197.

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

[54] S. X. Yu and J. Shi, “Multiclass spectral clustering,” in *Proc. IEEE 9th Int. Conf. Comput. Vis.*, 2003, pp. 313–319.

[55] Z. Wen and W. Yin, “A feasible method for optimization with orthogonality constraints,” *Math. Program.*, vol. 142, no. 1/2, pp. 397–434, 2013.

[56] M. A. Fanty and R. A. Cole, “Spoken letter recognition,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Morgan Kaufmann, 1990, pp. 220–226.

[57] M. J. Lyons, J. Budynek, and S. Akamatsu, “Automatic classification of single facial images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 12, pp. 1357–1362, Dec. 1999.

[58] J. Li et al., “Feature selection: A data perspective,” *ACM Comput. Surv.*, vol. 50, no. 6, 2018, Art. no. 94.

[59] D. Cai, X. He, J. Han, and H.-J. Zhang, “Orthogonal laplacianfaces for face recognition,” *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3608–3614, Nov. 2006.

[60] F. Alimoglu and E. Alpaydin, “Combining multiple representations and classifiers for pen-based handwritten digit recognition,” in *Proc. IEEE 4th Int. Conf. Document Anal. Recognit.*, 1997, pp. 637–640 vol.2.

[61] J. J. Hull, “A database for handwritten text recognition research,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, May 1994.

[62] 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, Jul. 1997.

[63] T. Li and C. H. Q. Ding, “Weighted consensus clustering,” in *Proc. SIAM Int. Conf. Data Mining*, 2008, pp. 798–809.

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

[65] Y. Jia, S. Tao, R. Wang, and Y. Wang, “Ensemble clustering via co-association matrix self-enhancement,” *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–12, 2023, doi: 10.1109/TNNLS.2023.3249207.

Peng Zhou received the BE degree in computer science and technology from the University of Science and Technology of China, in 2011, and the PhD degree in computer science from the 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 40 papers in highly regarded conferences and journals, including *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Cybernetics*, *ACM Transactions on Knowledge Discovery from Data*, IJCAI, AAAI, MM, etc. His research interests include machine learning and data mining. More information can be found at: https://doctor-nobody.github.io/.

Liang Du received the BE degree in software engineering from Wuhan University, in 2007, and the PhD 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.

Xinwang Liu received the PhD degree from the National University of Defense Technology (NUDT), China, in 2013. Currently, he is a professor with the School of Computer, NUDT. His current research interests include kernel learning, multi-view clustering, and unsupervised feature learning. He has published 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*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Multimedia*, *IEEE Transactions on Information Forensics and Security*, ICML, NeurIPS, CVPR, ICCV, AAAI, IJCAI, etc. He is an associate editor of the *IEEE Transactions on Neural Networks and Learning Systems* and *Information Fusion Journal*. More information can be found at https://xinwangliu.github.io/.

Zhaolong Ling received the the PhD degree from the School of Computer and Information, Hefei University of Technology, China, in 2020. He is a lecturer with the School of Computer Science and Technology, Anhui University, China. His research interests include feature selection, casual discovery, and data mining.

Xia Ji received the PhD degree from Anhui University. She is currently a lecturer with the School of Computer Science and Technology, Anhui University. Her research interests include machine learning, data mining, and granular computing.

Xuejun Li received the PhD degree from Anhui University, in 2008. He is currently a professor with the School of Computer Science and Technology, Anhui University, China. His major research interests include workflow systems, cloud computing, and intelligent software.

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