Ensemble Clustering via Co-Association Matrix Self-Enhancement

Yuheng Jia*, Member, IEEE, Sirui Tao*, Ran Wang*, Member, IEEE, and Yongheng Wang

Abstract—Ensemble clustering integrates a set of base clustering results to generate a stronger one. Existing methods usually rely on a co-association (CA) matrix that measures how many times two samples are grouped into the same cluster according to the base clusterings to achieve ensemble clustering. However, when the constructed CA matrix is of low quality, the performance will degrade. In this article, we propose a simple, yet effective CA matrix self-enhancement framework that can improve the CA matrix to achieve better clustering performance. Specifically, we first extract the high-confidence (HC) information from the base clusterings to form a sparse HC matrix. By propagating the highly reliable information of the HC matrix to the CA matrix and complementing the HC matrix according to the CA matrix simultaneously, the proposed method generates an enhanced CA matrix for better clustering. Technically, the proposed model is formulated as a symmetric constrained convex optimization problem, which is efficiently solved by an alternating iterative algorithm with convergence and global optimum theoretically guaranteed. Extensive experimental comparisons with 12 state-of-the-art methods on ten benchmark datasets substantiate the effectiveness, flexibility, and efficiency of the proposed model in ensemble clustering. The codes and datasets can be downloaded at https://github.com/Siritao/EC-SMCS.

Index Terms—Co-association (CA) matrix, ensemble clustering.

I. INTRODUCTION

Clustering is an important unsupervised machine-learning task that divides samples into a set of groups to exploit their intrinsic patterns [1]. In the past few years, many clustering methods have been proposed, like K-means [2], [3], mean-shift [4], hierarchical clustering [1], Gaussian mixture models [5], spectral clustering [6], [7], [8], deep embedding clustering [9], and so on. Different clustering methods have their own benefits and drawbacks and usually consist of several hyperparameters, which may fit different kinds of problems. However, how to select an appropriate clustering method for a specific problem and how to determine the hyperparameters of the selected clustering method are still quite challenging. To this end, ensemble clustering (also known as consensus clustering) [10], [11] was proposed to remedy this issue. Specifically, ensemble clustering first generates a set of clustering results by different clustering methods or by a typical clustering method with different hyperparameters and initializations and then integrates those base clustering results to generate a consensus one, which is stronger than all the base clusterings.

To achieve ensemble clustering, existing methods usually first construct a co-association (CA) matrix that records the co-occurrence relationships of samples, that is, the frequencies of every pair of samples being grouped into the same cluster by the base clusterings [13]. As the CA matrix reflects the pairwise similarity among samples, it can serve as a similarity matrix or adjacency matrix. Then any typical similarity matrix-based clustering methods like spectral clustering or graph partitioning (GP) techniques [10], [14], [15] can be applied to the CA matrix to generate the ensemble clustering result. Recently, many efforts have been made to improve the quality of the traditional co-occurrence CA matrix based on different priors and assumptions. For example, Li and Ding [16] globally weighted the CA matrix by ranking the quality of each base clustering result. Huang et al. [12] proposed the locally weighted CA (LWCA) matrix with an ensemble-driven cluster uncertainty estimation and the local weighting strategy. In probability trajectory accumulation (PTA) [17], the must-link relations among samples rather than all the individual links were focused on, so they built a microcluster-based CA (MCA) matrix that is smaller in scale. Some methods assume that the learned CA matrix holds a K-block-diagonal structure, with K being the number of clusters. For instance, Zhou et al. [18] used Fan’s [19] theorem to seek it and designed the CA matrix by self-paced learning and further extended it to a bipartite graph [20]. Zhou et al. [21] tried to construct a robust CA matrix by multiple graph learning. Apart from them, many approaches learned CA matrices with the low-rank assumption. Yi et al. [22] viewed reliable relations between samples as observed information and used the matrix completion technique to recover a low-rank
unfortunately, the majority of the elements in a CA matrix have samples are likely to be grouped into the same cluster, while a lower (resp. lower) value provides a more (resp. less) reliable description of the CA matrix. We found that in a CA matrix, a higher (resp. lower) value provides a more (resp. less) reliable description of the CA matrix. We also calculated the precision and recall for different values in the CA matrix. Note that we normalized the proportion of the elements of the CA matrix for better visualization.

To further improve the clustering performance, in this article, we propose a CA matrix self-enhancement method to strengthen the quality of the CA matrix, without any extra information. We found that in a CA matrix, a higher (resp. lower) value provides a more (resp. less) reliable description of the pairwise relationship between two samples, that is, two samples are likely to be grouped into the same cluster, while unfortunately, the majority of the elements in a CA matrix have low values. Fig. 1 vividly shows this observation. Motivated by this phenomenon, we first extract the high-confidence (HC) information from the base clustering results to constitute a sparse HC matrix, which provides limited but highly reliable pairwise descriptions of samples. Thereafter, we leverage the HC matrix and the original CA matrix to learn an ideal CA matrix. Specifically, for the HC matrix, we propose to propagate its information to the ideal CA matrix. While for the original CA matrix with many low-reliable, but relatively dense connections, we try to denoise it to recover the ideal one. By simultaneously using the information in the HC matrix and the original CA matrix, we are able to learn an enhanced CA matrix with relatively dense connections and more highly reliable information. The proposed model is finally formulated as a Laplacian-regularized convex optimization problem, and we develop an efficient iterative optimization method to solve it with convergence and global optimum theoretically guaranteed. Extensive experiments on ten benchmark datasets validate its excellent clustering performance when compared with 12 recent ensemble clustering methods. Moreover, the proposed model can adapt to multiple CA matrices and runs faster than most state-of-the-art methods with an iterative optimization process. The main contributions of this article are summarized as follows.

1) We propose a novel ensemble clustering model from the perspective of CA matrix self-enhancement. The proposed model can promote different CA matrices without any extra information.

2) The proposed model is formulated as a convex optimization problem, which is solved by an alternating iterative algorithm with convergence guaranteed. It also runs faster than most of the state-of-the-art ensemble clustering methods with an iterative optimization process.

3) The proposed model exceeds the state-of-the-art ensemble clustering methods significantly in clustering performance. Besides, the proposed model is robust to the hyperparameters.

The structure of this article is organized as follows. We first review related works in Section II. Thereafter, we introduce the proposed model with its numerical solution in Section III. Section IV presents the experimental results and the associated analysis, and Section V summarizes this article.

II. RELATED WORK

Given n samples $\mathcal{X} = \{x_1, \ldots, x_n\}$ and m base clustering results $\Pi = \{\pi_1, \ldots, \pi_m\}$, where $x_i$ denotes the $i$th sample and $\pi_j$ denotes the $j$th base clustering that partitions $\mathcal{X}$ into several clusters $C = \{c_1, \ldots, c_l\}$, with $l_j$ being the number of clusters in $C_j$. According to Fred and Jain [13], we can construct a co-occurrence CA matrix $A \in \mathbb{R}^{n \times n}$ by

$$\tilde{A}_{ij} = \frac{1}{m} \sum_{k=1}^{m} \delta(\hat{c}(x_i), \pi_j(k))$$

(1)

where $\hat{c}(x_i) \in C_k$ is the cluster membership of $x_i$ in the $k$th base clustering $\pi_k$, and $\delta(\cdot, \cdot)$ is the Kronecker delta function

$$\delta(x, y) = \begin{cases} 1, & x = y \\ 0, & x \neq y \end{cases}$$

As $\tilde{A}_{ij}$ measures the frequency that $x_i$ and $x_j$ occur in the same cluster, it reveals the pairwise similarity between them according to the base clusterings $\Pi$. Fred and Jain [13] apply hierarchical agglomerative clustering on the CA matrix to produce a consensus clustering result, which is known as evidence accumulation clustering (EAC). Hereafter, CA-based methods have become the mainstream of ensemble clustering due to their effectiveness and efficiency, and many advanced CA matrix construction methods were proposed [12], [16], [17], [18], [20], [21], [22], [23], [24]. In the following, we will introduce two representative ones.

A. Locally Weighted CA

Locally weighted ensemble clustering [12] refines the CA matrix by focusing on local diversity and cluster uncertainty.

Note that in this article, we use $A$ to denote the traditional CA matrix in [13] that measures the frequency of two samples being grouped into the same cluster according to the base clusterings and use $\tilde{A}$ to denote more advanced (general) CA matrices (like LWCA) to be enhanced by our method.
First, the uncertainty of a given cluster \(c'_i\) from \(\pi_i\) is estimated by the entropy, that is,
\[
U^\pi(c'_i) = -\sum_{j=1}^{l_i} \sum_{k=1}^{p} p(c'_i, c'_k) \log_2 p(c'_i, c'_k).
\]

Then the ensemble-driven cluster index (ECI) comes up to measure the reliability of a cluster that is negatively correlated with this entropy by a hyperparameter \(\theta\)
\[
\text{ECI}(c'_i) = e^{-\frac{U^\pi(c'_i)}{\alpha}}.
\]

Finally, the ECIs of all clusters are applied as local weights to compute an LWCA matrix \(A \in \mathbb{R}^{n \times n}\)
\[
A_{ij} = \frac{1}{m} \sum_{k=1}^{m} \delta(c^k(x_i), c^k(x_j)) \cdot \text{ECI}(c^k(x_i)).
\]

After building the LWCA matrix, hierarchical clustering or Tcut GP [15] is used to generate the final partition.

### B. MCA and PTS

PTA [17] first forms a compact MCA matrix. Specifically, if all the base clustering assigns a group of samples into one cluster, PTA will treat those samples indistinguishably and aggregate them into a microcluster, which replaces those individual samples in the follow-up steps. PTA creates \(n'\) nonoverlapping microclusters \(\mathcal{Y} = \{y_1, \ldots, y_{n'}\}\), with each microcluster \(y_i\) containing \(r_i\) samples. The MCA matrix \(A' \in \mathbb{R}^{n' \times n'}\) records the frequencies that every two microclusters occur in the same cluster from \(m\) base clustering results in \(\Pi\)
\[
A'_{ij} = \frac{1}{m} \sum_{k=1}^{m} \delta(c^k(y_i), c^k(y_j))
\]
where \(c^k(y_i) \in C^k\) is the cluster that \(\pi_k\) assigns microcluster \(y_i\) to.

Afterward, PTA proposes an elite neighbor selection strategy to improve the quality of MCA. Specifically, a sparse similarity graph \(W = \{w_{ij}\} \in \mathbb{R}^{n' \times n'}\) is obtained by reserving the top-\(V\) (\(V\) is a hyperparameter) links of each node and pruning other links, then the random walk is conducted on \(W\) to produce a dense similarity matrix. Considering that the transition probability from one node to another may be proportional to the size of microclusters and the weight of their link, the transition probability matrix \(Q = \{q_{ij}\} \in \mathbb{R}^{n' \times n'}\) can be calculated by
\[
q_{ij} = \frac{w_{ij}}{\sum_{i \neq k} w_{ik}}.
\]

Based on this, the \(T\)-step transition probability from \(y_i\) to \(y_j\) is represented by \([Q^T]\)_{i,j}, and the probability trajectory of a random walker from node \(y_i\) with step length \(T\) is defined as a \(Tn'\)-tuple \(R_i = [Q^T_{i,1}, \ldots, Q^T_{i,n'}]\), where \(Q^T_{i,j}\) denotes the \(i\)th row in \(Q^T\). The final probability trajectory similarity (PTS, also can be regarded as an advanced CA matrix) between two microclusters is the cosine similarity of their probability trajectory, that is,
\[
A_{ij} = \frac{(R_i, R_j)}{\sqrt{(R_i, R_i) \cdot (R_j, R_j)}}
\]
where \((\cdot, \cdot)\) denotes the inner product operation.

The PTS matrix serves as a similarity measure among microclusters. After applying hierarchical clustering or Tcut on it, the ensemble clustering result can be generated by mapping microclusters back to objects.

As the CA matrix plays a crucial role in ensemble clustering, in this work, we investigate how to improve its quality without additional information.

### III. PROPOSED METHOD

#### A. Formulation

In this section, we will present a novel CA matrix self-enhancement approach to promoting different kinds of CA matrices and accordingly produce better ensemble clustering performance. From Fig. 1, we can see that in a typical CA matrix, when the majority of base clusterings group two samples in the same cluster, those two samples are very likely to belong to the same class (with a high precision), but the total amount of this highly reliable information is limited (with a low recall). On the contrary, there are a lot of low-value elements (with a high recall) in the CA matrix, which means that the base clusterings provide relatively inconsistent descriptions (with low precision) of two samples. Apparently, there exists a complementary relationship between those two kinds of information. As an ideal CA matrix is expected to be filled with dense and highly reliable elements, we, therefore, propose to leverage these two complementary pieces of information to enhance a given CA matrix \(A\).

We first define an HC matrix \(H = \{h_{ij}\} \in \mathbb{R}^{n \times n}\) to capture the limited but highly reliable information from base clusterings \(\Pi\), that is,
\[
H = \Psi_\Omega(\tilde{A})
\]
where \(\Omega = \{(i, j) \mid \tilde{A}_{ij} \geq \alpha\}\) records the locations of the highly reliable elements and \(\Psi_\Omega : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}\) is an element-wise mapping operator
\[
[\Psi_\Omega(\tilde{A})]_{ij} = \begin{cases} 
\tilde{A}_{ij}, & (i, j) \in \Omega \\
0, & (i, j) \notin \Omega.
\end{cases}
\]

Here, \(\tilde{A}\) is the traditional CA matrix described in (1), which directly captures the co-occurrence relationships among samples from base clustering results. If the ratio of the times that two samples \(x_i\) and \(x_j\) are grouped into the same class to the total number of base clusterings exceeds a predefined threshold \(\alpha \in [0, 1]\), the corresponding entry \(\tilde{A}_{ij}\) is viewed as a piece of highly reliable information (as well as an element in the HC matrix). As the entries in \(\tilde{A}\) are discrete and normalized in [0, 1], while that in \(A\) usually has an arbitrary scale and distribution, we use \(\tilde{A}\) rather than \(A\) to generate the HC matrix, making it easy to determine the value of \(\alpha\).

Note that \(\tilde{A}\) is a general similarity matrix that may have many concrete forms like the traditional co-occurrence CA matrix proposed by Fred and Jain [13], or LWCA [12], PTS [17], and so on. These similarity matrices can all be enhanced via our model; see related results in Section IV-C.
Second, we would directly use the given input CA matrix $A$ to depict the relatively low-reliable information. Accordingly, CA matrix self-enhancement becomes using $H$ and $A$ to recover an ideal CA matrix $C \in \mathbb{R}^{n \times n}$.

As $A$ may contain much inaccurate information, which can be regarded as the combination of the ideal CA matrix with noise. We, therefore, try to remove the noise from $A$ to construct $C$, that is,

$$
\min_{C,E} \|E\|_F^2 \\
\text{s.t. } A = C + E, \quad C = C^T, \quad 0 \leq C \leq 1 \quad (3)
$$

where $E \in \mathbb{R}^{n \times n}$ denotes the error term and $\| \cdot \|_F$ denotes the Frobenius norm of a matrix. Ideally, $C$ should be symmetric because if $x_i$ is similar to $x_j$, $x_j$ should also be similar to $x_i$. Besides, with $C_{ij}$ being the similarity measure, its value should lie in $[0,1]$.

However, solving (3) will lead to a trivial solution, that is, $C = A$ and $E = 0$. To avoid this, it is significant to further propagate the limited but highly valuable information from $H$ to help build a reasonable $C$. Let $C_{i,:}$ denote the $i$th row of the matrix $C$, which can reflect the pairwise similarity relationships of $x_i$ with the other samples. If $h_{ij} > 0$, $x_i$ and $x_j$ are very likely to belong to the same cluster. Accordingly, the pairwise relationships of $x_i$ with the other samples should also be close to that of $x_j$, that is, when $h_{ij} > 0$, we can infer that $C_{i,:}$ should be close to $C_{j,:}$. The intuition to minimize row-wise discrepancies with respect to every pair of samples that potentially belong to the same cluster can be mathematically expressed as

$$
\min_C \sum_{i,j=1}^n h_{ij} \|C_{i,:} - C_{j,:}\|^2_2.
$$

In addition, if $h_{ij} > 0$, we can expect that $A_{ij}$ is accurate enough to represent the pairwise similarity between $x_i$ and $x_j$. We, thus, copy those crucial entries of $A$–$C$ directly, which would serve as the supporter of the CA structure. Taking all the above analyses into consideration, the proposed model is further formulated as

$$
\min_{C,E} \frac{1}{2} \sum_{i,j=1}^n h_{ij} \|C_{i,:} - C_{j,:}\|^2_2 + \frac{\lambda}{2} \|E\|_F^2 \\
\text{s.t. } A = C + E, \quad C_{ij} = A_{ij} \quad \forall (i,j) \in \Omega \\
C = C^T, \quad 0 \leq C \leq 1 \quad (4)
$$

where $\lambda > 0$ balances the two loss terms. The first term can be written in a quadratic form $\text{tr}(C^T \Phi C)$ [25], where $\Phi = D - H$ is the graph Laplacian matrix, and $D$ is the diagonal degree matrix with its diagonal element $D_{ii} = \sum_j H_{ij}$. Also, operator $\Psi_\Omega$ can be assigned to matrix $E$ to compensate for unknown entries of $A$ but locks $C_{ij} = A_{ij}$ for $(i,j) \in \Omega$. Finally, the proposed model is rewritten as

$$
\min_{C,E} \frac{1}{2} \text{tr}(C^T \Phi C) + \frac{\lambda}{2} \|E\|_F^2 \\
\text{s.t. } A = C + E, \quad \Psi_\Omega(E) = 0, \quad C = C^T, \quad 0 \leq C \leq 1 \quad (5)
$$

As a summary, (5) propagates the highly reliable information in $H$ and removes the noise in $A$ simultaneously to learn an enhanced CA matrix $C$ without any extra information. It can be applied to enhance different kinds of CA matrices (or similarity matrices) like $A$, LWCA, PTS, and so on. After solving (5), average-link (AL) hierarchical agglomerative clustering is applied on $C$ to obtain the final clustering result. Other clustering approaches like spectral clustering are also applicable here.

### B. Optimization

We adopt the alternating direction method of multipliers (ADMM) method [26] to solve (5), which can handle the equality constraint and the multiple variables effectively. To alleviate the value range constraint and the symmetric constraint of $C$, we introduce an intermediate matrix $F$ to shoulder them and equivalently rewrite (5) as

$$
\min_{C,E,F} \text{tr}(C^T \Phi C) + \frac{\lambda}{2} \|E\|_F^2 \\
\text{s.t. } A = C + E, \quad \Psi_\Omega(E) = 0, \quad C = F \\
F = F^T, \quad 0 \leq F \leq 1. \quad (6)
$$

Let $Y_1$ and $Y_2$ denote the Lagrangian multipliers, the augmented Lagrangian function of (6) can be written as

$$
\mathcal{L}(C, E, F, Y_1, Y_2)
= \text{tr}(C^T \Phi C) + \frac{\lambda}{2} \|E\|_F^2 \\
+ \langle Y_1, A - C - E \rangle + \frac{\gamma_1}{2} \|A - C - E\|_F^2 \\
+ \langle Y_2, C - F \rangle + \frac{\gamma_2}{2} \|C - F\|_F^2 \\
\text{s.t. } \Psi_\Omega(E) = 0, \quad F = F^T, \quad 0 \leq F \leq 1 \quad (7)
$$

where $\gamma_1, \gamma_2 > 0$ introduce the augmented Lagrangian terms. In the experiments, we fixed $\gamma_1 = \gamma_2 = 1$, which is a good choice for many ADMM-based optimization solvers [26]. The solution to (5) can be obtained when (7) reaches its minimum. Specifically, (7) can be solved by the following iterative alternating procedures.

1) Updating $C$: With other variables fixed, the $C$ subproblem is formulated as

$$
C_{k+1} = \arg \min_C \text{tr}(C^T \Phi C) + \frac{\gamma_1}{2} \|C - P_{1k}\|_F^2 \\
+ \frac{\gamma_2}{2} \|C - P_{2k}\|_F^2
$$

where $P_{1k} = A - E_k + Y_{1k}/\gamma_1$ and $P_{2k} = F_k - Y_{2k}/\gamma_2$. By setting the derivative of (8) to 0, we have the following analytical solution of the $C$ subproblem:

$$
C_{k+1} = (2\Phi + (\gamma_1 + \gamma_2)I)^{-1}(\gamma_1 P_{1k} + \gamma_2 P_{2k}). \quad (9)
$$

2) Updating $E$: With other variables fixed, the formulation of the $E$ subproblem is

$$
E_{k+1} = \arg \min_E \frac{\gamma_1}{2} \|E - (A - C_{k+1})\|_F^2 + \frac{\lambda}{2} \|E\|_F^2 \\
+ \langle Y_{1k}, A - C_{k+1} - E \rangle \\
\text{s.t. } \Psi_\Omega(E) = 0. \quad (10)
$$

As the Frobenius norm, the inner product, and the equality constraint are all computed element-wise, we can get the
Algorithm 1: Ensemble Clustering via CA Matrix Self-Enhancement

Input: CA matrix A, base clusterings Π, threshold α, trade-off parameter λ.

Initialization: Penalty parameter γ1 = γ2 = 1, tolerance ε = 1e-2, k = 0, E_k = C_k = F_k = 0 \in \mathbb{R}^{n \times n}, Y_{ik} = A - C_k - E_k, Y_{2k} = C_k - F_k.

1: Construct the co-occurrence CA matrix \tilde{A} by Eq. (1) from Π, then generate the HC indices set Ω and derive the HC matrix H via Eq. (2).
2: Compute Φ = diag[H \cdot (1 \in \mathbb{R}^n)] − H.
3: while not converged do
4: Update C_{k+1} via Eq. (9).
5: Update E_{k+1} via Eq. (11).
6: Update F_{k+1} via Eq. (13).
7: Update Y_{1k+1}, Y_{2k+1} via Eq. (14).
8: Check stopping criteria:
9: \begin{align*}
\max(σ_2(C), σ_2(E), σ_2(F), σ_2(Y_1), σ_2(Y_2)) ≤ ε
\end{align*}
with σ_2(C) = (\|C_{k+1} - C_k\|_F^2)/\|C_k\|_F^2.
10: end while
11: return the enhanced CA matrix \tilde{C}.

The global solution of (10) through

\begin{align*}
E^* = \frac{γ1(A - C_{k+1}) + Y_{ik}}{λ + γ1}, E_{k+1} = \Psi_Ω(E^*) \tag{11}
\end{align*}

where \tilde{Ω} is the complementary set of Ω.

3) Updating F: With other variables fixed, the F subproblem can be expressed as

\begin{align*}
F_{k+1} &= \arg \min_F \frac{1}{2}\|F - P_{3k}\|_F^2 \\
& \quad \text{s.t. } F = F^T, \quad 0 ≤ F ≤ 1 \tag{12}
\end{align*}

where P_{3k} = C_{k+1} + Y_{3k}/γ2. Since F = F^T suggests that \|F - P_{3k}\|_F^2 = \|F - P_{3k}\|_F^2, (12) is equivalent to

\begin{align*}
F_{k+1} &= \arg \min_F \frac{1}{4}(\|F - P_{3k}\|_F^2 + \|F - P_{3k}\|_F^2) \\
& = \arg \min_F \frac{1}{2}\|F - P_{3k} + P_{3k}\|_F^2 + c(P) \\
& \quad \text{s.t. } F = F^T, \quad 0 ≤ F ≤ 1
\end{align*}

with c(P) irrelevant to F. Thereafter, the optimum of (12) is achieved through an element-wise truncation

\begin{align*}
F_{k+1} = \min \left(\max\left(\frac{P_{3k} + P_{3k}^T}{2}, 0\right), 1\right). \tag{13}
\end{align*}

4) Updating Y: The ADMM algorithm updates the multiplier matrices \{E, F\} by

\begin{align*}
Y_{1(k+1)} &= Y_{1k} + γ1(A - C_{k+1} - E_{k+1}) \\
Y_{2(k+1)} &= Y_{2k} + γ2(C_{k+1} - F_{k+1}) \tag{14}
\end{align*}

The whole procedure is summarized in Algorithm 1. As (5) is a convex problem with two blocks of variables (i.e., \{C\}
and \{E, F\}, since updating variables E and F are independent of each other), Algorithm 1 is theoretically guaranteed to converge to a global solution by ADMM [26].

C. Computational Complexity Analysis

Updating C involves a matrix inverse operation and a matrix multiplication operation. As in each iteration, the
to-be-inverted matrix 2F + (γ1 + γ2)I is fixed so that it can be calculated in advance, which means the computational
difficulty of solving the C subproblem is O(n^3) [31]. Solving subproblems of E and F and updating multiplier
matrices Y_1 and Y_2 only include element-wise calculations, leading to computational complexity of O(n^2). Therefore, the overall computational complexity of Algorithm 1 is O(n^2.37) in one iteration.

IV. Experiment

We conducted a series of experiments on eight real-world benchmark datasets with a modest scale, and two large-scale datasets to verify the effectiveness of the proposed algorithm, whose detailed information is summarized in Tables I and II. Following Huang et al. [12], for all the datasets, 100 candidate base clusterings were generated by K-means with K randomly selecting in [2, \sqrt{n}] and n being the number of samples.

We compared the proposed method with 12 representative state-of-the-art ensemble clustering algorithms, which are listed as follows.

1) PTA-AL, PTA-CL, and PTGP [17]: Three microcluster representation-based ensemble clustering methods.
using similarity matrix derived from probability trajectories of random walkers. They adopt AL hierarchical agglomerative clustering, complete-link (CL) hierarchical agglomerative clustering, and Tcut GP [15], respectively, to generate clustering results.

2) **LWEA and LWGP [12]**: Two locally weighted ensemble clustering methods with AL hierarchical agglomerative clustering and Tcut, corresponding to LWEA (evidence accumulation) and LWGP (GP). The local weights are obtained via an ensemble-driven cluster uncertainty estimation.

3) **RSEC-H and RSEC-Z [24]**: Robust spectral ensemble clustering methods via rank minimization. The algorithm simultaneously learns a consensus partition $H$ and a low-rank representation $Z$, and, respectively, adopts $K$-means or spectral clustering on them to get final results.

4) **DREC [32]**: A dense representation-based ensemble clustering algorithm. It also uses the idea of forming microclusters but with the consideration of outliers.

5) **SPCE [18]**: Ensemble clustering with a self-paced manner. It learns a CA matrix from easy-to-learn samples to difficult-to-learn ones.

6) **ECPCS-MC and ECPCS-HC [33]**: Two ensemble clustering methods that improve the CA matrix by propagating cluster-wise similarities. Here, MC refers to cluster-level meta-clustering and HC refers to AL hierarchical agglomerative clustering.

7) **TRCE [21]**: A multiple graph learning-based ensemble clustering method combining three levels of robustness, that is, base clustering level, graph level, and instance level.

For a fair comparison, we carefully tuned the hyperparameters of those compared methods according to their original papers and reported the best performances. As for the proposed method, the LWCA matrix [12] was employed as the input CA matrix. For all the methods, we randomly selected 20 base clusterings from the candidate base clustering pool and recorded the average clustering performance with the standard deviation over 20 repetitions. All experiments were conducted by MATLAB R2020a on a PC with a 2.3-GHz CPU and 16-GB memory.

We adopted five widely used metrics, that is, adjusted rand index (ARI) [34], normalized mutual information (NMI) [10], F-score (the harmonic mean of precision and recall), Accuracy, and Purity to evaluate the clustering performance of different methods. All of them lie in the range of $[0, 1]$, and larger values reflect better clustering performance.

### A. Parameter Sensitivity

1) **Sensitivity of Hyperparameters**: There are two hyperparameters $\alpha$ and $\lambda$ in our model, where $\alpha$ decides to what extent we can accept an entry in $A$ as an HC element, and $\lambda$ balances the importance of the error term. We investigated their influence on the proposed model in Figs. 2 and 3.

   1) First, it can be seen from Fig. 2 that the optimal performance is achieved when $\alpha = 0.7$ on Caltech20, 0.75 on Ecoli, and 0.8 on all other datasets. Moreover, an apparent improvement of our method over the baseline LWEA usually occurs when $\alpha \in \{0.7, 0.75, 0.8\}$ since a bigger $\alpha$ will largely reduce the amount of the highly reliable information, while a smaller $\alpha$ will degrade the quality of HC matrix.

   2) Second, from Fig. 3, we can observe that when $\lambda$ is too large (e.g., $\lambda \geq 100$), the performance of the proposed model will be close to that of LWEA, as with a large $\lambda$, the self-enhanced CA matrix will be close to the original one. $\lambda$ should also not be too small, as with a small $\lambda$, the majority of elements of the original CA matrix will be regarded as noise and be removed. Moreover, our model significantly surpasses the baseline on a wide range of $\lambda$, that is, $\lambda \in [0.01, 1]$ on most of the datasets, proving its robustness.

As a summary, we suggest setting $\alpha = 0.8$ and $\lambda = 0.4$ for our model.

---

**Fig. 2.** Clustering performances with respect to ARI with varying $\alpha$. Both LWEA and the proposed method adopt AL hierarchical agglomerative clustering on the CA matrix to produce the clustering result. All the subfigures share the same legend.
Influence of ADMM Parameters ($\gamma_1$ and $\gamma_2$) on the UMIST Dataset (\(\alpha\) and $\lambda$ were set to the recommended value). We fixed one of them and varied the other, then recorded the offset of ARI, NMI, and F-Score compared with default setting ($\gamma_1 = \gamma_2 = 1$), and the average variation ratio of C, measured by Frobenius norm.

| ADMM Parameter | $\gamma_1 = 1, \gamma_2 = \_\_\_\_\_\_\_\_$ | $\gamma_1 = 1, \gamma_2 = \_\_\_\_\_\_\_$ |
|----------------|---------------------------------|---------------------------------|
| Measure       | 0.01 | 0.1 | 1 | 10 | 100 | 0.01 | 0.1 | 1 | 10 | 100 |
| ARI           | +0.001 | 0 | 0.367 | 0 | 0 | -0.034 | -0.001 | 0.367 | -0.002 | -0.012 |
| NMI           | +0.001 | 0 | 0.682 | 0 | 0 | -0.035 | 0 | 0.682 | -0.001 | -0.010 |
| P-score       | +0.001 | 0 | 0.406 | 0 | 0 | -0.028 | 0 | 0.406 | -0.002 | -0.013 |

$\frac{|C_{\gamma_1, \gamma_2} - C_{\gamma_1=\gamma_2=1}|_F}{|C_{\gamma_1=\gamma_2=1}|_F} \approx 0.05$ 0.02 0 0 0.02 0.38 0.38 0.05 0 0.03 0.11

2) Influence of ADMM Parameters: The adopted optimization method ADMM introduces two parameters $\gamma_1$ and $\gamma_2$. In all the experiments, we fixed them as $\gamma_1 = \gamma_2 = 1$. Here, we studied their influence on the proposed method. Table III shows when varying the values of $\gamma_1$ and $\gamma_2$ from a wide range $[0.1, 10]$, the clustering performance barely changes. Besides, in that range, the converged variables of the proposed model also do not change dramatically. Therefore, we can conclude that ADMM is quite robust to $\gamma_1$ and $\gamma_2$, and setting $\gamma_1 = \gamma_2 = 1$ in all the experiments is a reasonable choice.

B. Clustering Performance Comparison

1) Results Comparison: Table IV shows the average ARI as well as the standard deviation of the compared and proposed algorithms on the eight modest-scale benchmark datasets. We provide two groups of results with our algorithm:
TABLE V
CLUSTERING PERFORMANCES OF DIFFERENT ALGORITHMS ON TWO LARGE-SCALE DATASETS MEASURED BY ARI, NMI, AND F-SCORE

| Dataset         | ARI     | ISOLET    | F-score | ARI     | NMI     | F-score |
|-----------------|---------|-----------|---------|---------|---------|---------|
| Base clusterings (average) | 0.404±0.074 | 0.697±0.059 | 0.426±0.070 | 0.215±0.066 | 0.535±0.035 | 0.249±0.079 |
| Base clusterings (best)     | 0.516   | 0.739     | 0.537   | 0.339   | 0.578   | 0.399   |
| PTA-AL [TKDE, 2016]         | 0.522±0.012 | 0.752±0.004 | 0.542±0.012 | 0.535±0.043 | 0.662±0.026 | 0.586±0.038 |
| PTA-CL [TKDE, 2016]         | 0.527±0.008 | 0.752±0.005 | 0.546±0.008 | 0.384±0.040 | 0.557±0.032 | 0.450±0.036 |
| PTCG [TKDE, 2016]           | 0.526±0.022 | 0.752±0.009 | 0.546±0.021 | 0.531±0.050 | 0.649±0.029 | 0.580±0.044 |
| LWOP [TICY, 2018]           | 0.514±0.017 | 0.753±0.006 | 0.535±0.016 | 0.477±0.035 | 0.643±0.023 | 0.535±0.031 |
| DREC [Neurocomputing, 2019] | 0.531±0.011 | 0.755±0.006 | 0.550±0.011 | 0.546±0.045 | 0.667±0.028 | 0.593±0.040 |
| SPCE [TNNLS, 2021]          | 0.435±0.055 | 0.759±0.018 | 0.465±0.050 | 0.402±0.064 | 0.605±0.032 | 0.473±0.050 |
| ECPCS-MS [TSMC-S, 2021]     | 0.513±0.017 | 0.751±0.016 | 0.533±0.004 | 0.472±0.012 | 0.627±0.011 | 0.530±0.012 |
| ECPCS-HC [TSMC-S, 2021]     | 0.522±0.021 | 0.757±0.020 | 0.543±0.008 | 0.454±0.024 | 0.622±0.019 | 0.520±0.017 |
| TRCE [AAAI, 2021]           | 0.503±0.007 | 0.754±0.025 | 0.524±0.024 | 0.471±0.021 | 0.650±0.023 | 0.534±0.019 |
| LWEA [TICY, 2018]           | 0.553±0.016 | 0.764±0.008 | 0.573±0.015 | 0.525±0.033 | 0.663±0.019 | 0.578±0.028 |
| Proposed (α = 0.8, λ = 0.4) | 0.553±0.020 | 0.775±0.008 | 0.574±0.019 | 0.519±0.035 | 0.663±0.019 | 0.574±0.029 |
| Proposed               | 0.562±0.014 | 0.774±0.007 | 0.582±0.013 | 0.540±0.040 | 0.672±0.019 | 0.591±0.034 |

Proposed and Proposed (α = 0.8, λ = 0.4). The former demonstrates the best-tuned results of the proposed algorithm with respect to ARI and the latter corresponds to the proposed algorithm with fixing hyperparameters α = 0.8 and λ = 0.4. Note that for the compared methods, their hyperparameters were carefully tuned, and the best performance was reported. Additionally, the best and average performance of the base clusterings from the candidate base clustering pool were also included. The clustering performances with other metrics (NMI, F-Score, Accuracy, and Purity) are shown in the supplementary file. From those tables, we have the following observations.

1) First, nearly all the ensemble clustering methods perform better than the average performance of the base clusterings with different metrics, which indicates that leveraging multiple base clusterings is useful. But many compared methods cannot exceed the best base clustering on most datasets, for example, PTA-CL and RSEC-Z, suggesting their limitations. Differently, the proposed method can significantly outperform both the average performance of the base clusterings and the best base clustering in the candidate base clustering pool on most datasets, which proves that the proposed method can really exploit the useful information from the base clusterings and improve the clustering performance.

2) Second, the proposed method takes the CA matrix of LWEA as input, and it can largely improve the clustering performance of LWEA on all the datasets, for example, on Ecoli, our method improves the ARI from 0.430 to 0.754, NMI from 0.615 to 0.716, and F-score from 0.544 to 0.823. This is a straightforward proof that our method can improve the quality of the input CA matrix.

3) Third, the proposed method outperforms all the compared methods on all eight datasets with respect to ARI and F-score. For example, on Caltech20, the proposed method improves ARI by about 40% compared with the best comparison. In terms of NMI, our method achieves the best performance on six out of eight cases, while it is slightly inferior to SPCE on UMIST and SPF. Taking all the metrics into consideration, we can conclude that the proposed model ranks first, in general.

4) Finally, from the results of the Proposed (α = 0.8, λ = 0.4), it can be found that without tuning hyperparameters, the proposed method can still achieve the second-best performance on average (slightly inferior to the tuned one). This phenomenon proves that our method is very practical in real-world applications.

2) Results on Large-Scale Datasets: In Table V, we also tested the performance of the proposed algorithm on two large-scale datasets, that is, ISOLET and USPS. We can observe that on large-scale datasets, the proposed model also outperforms the state-of-the-art ensemble clustering methods in most cases with different metrics.

3) Visualization: Fig. 4 visually compares the LWCA matrix, the CA matrix of our method, their difference, and the ground-truth CA matrix on the Aggregation dataset. We can see that the proposed method can considerably build more strong and more reliable links than LWCA, which is much closer to the ideal one. Moreover, it can also weaken the wrong entries of the LWCA matrix [please zoomed-in view on the highlighted regions in Fig. 4(c)]. This further explains why our method can significantly improve the clustering performance compared with LWEA.
C. Adaptation to Diverse CAs

In this section, we check whether the proposed approach can adapt to other CA matrices except for LWCA. Here, we used the CA matrix from EAC [13] and the similarity matrix PTS from PTA [17] as the input and assessed the performance of our method on them, respectively. The associated results are listed in Tables VI and VII, where the proposed model consistently promotes the clustering performances of original algorithms to a great extent on the eight modest-scale datasets with different metrics. To be specific, when compared with EAC, the average improvements of our method are around 30% in ARI, 15% in NMI, and 20% in F-score. Especially, our method has pushed these three metrics nearly up to 1 on the Aggregation dataset. The improvements on PTA are 35%, 15%, and 30% on average, and 120%, 25%, and 75% singly on the Ecoli dataset with respect to ARI, NMI, and F-score, which is remarkable. Note that after the self-enhancement procedure by our method, when referring to Table IV and the results of

\begin{table}[h]
\centering
\caption{Illustration of the Clustering Improvement on EAC With Respect to ARI, NMI, and F-Score}
\begin{tabular}{|c|c|c|c|c|c|}
\hline
Metric & Method & EAC & Proposed & EAC & Proposed \\
\hline
\hline
\multirow{6}{*}{Caltech20} & 0.313 & 0.490 & 0.467 & 0.495 & 0.386 & 0.573 \\
\multirow{6}{*}{Ecoli} & 0.548 & 0.753 & 0.651 & 0.717 & 0.649 & 0.823 \\
\multirow{6}{*}{LS} & 0.467 & 0.638 & 0.563 & 0.675 & 0.575 & 0.714 \\
\multirow{6}{*}{HCT} & 0.121 & 0.140 & 0.210 & 0.201 & 0.268 & 0.365 \\
\multirow{6}{*}{Aggregation} & 0.896 & 0.989 & 0.942 & 0.984 & 0.916 & 0.991 \\
\multirow{6}{*}{Texture} & 0.575 & 0.673 & 0.731 & 0.802 & 0.617 & 0.707 \\
\multirow{6}{*}{UMIST} & 0.333 & 0.368 & 0.651 & 0.690 & 0.372 & 0.407 \\
\multirow{6}{*}{SPF} & 0.057 & 0.086 & 0.123 & 0.191 & 0.271 & 0.358 \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{Illustration of the Clustering Improvement on PTA-AL With Respect to ARI, NMI, and F-Score}
\begin{tabular}{|c|c|c|c|c|c|}
\hline
Metric & Method & EAC & Proposed & EAC & Proposed \\
\hline
\hline
\multirow{6}{*}{Caltech20} & 0.356 & 0.439 & 0.440 & 0.475 & 0.436 & 0.537 \\
\multirow{6}{*}{Ecoli} & 0.340 & 0.738 & 0.561 & 0.700 & 0.460 & 0.811 \\
\multirow{6}{*}{LS} & 0.593 & 0.664 & 0.635 & 0.687 & 0.673 & 0.735 \\
\multirow{6}{*}{HCT} & 0.136 & 0.190 & 0.242 & 0.277 & 0.271 & 0.335 \\
\multirow{6}{*}{Aggregation} & 0.600 & 0.896 & 0.786 & 0.932 & 0.681 & 0.920 \\
\multirow{6}{*}{Texture} & 0.658 & 0.702 & 0.772 & 0.800 & 0.691 & 0.730 \\
\multirow{6}{*}{UMIST} & 0.323 & 0.370 & 0.633 & 0.687 & 0.361 & 0.411 \\
\multirow{6}{*}{SPF} & 0.081 & 0.098 & 0.152 & 0.188 & 0.261 & 0.337 \\
\hline
\end{tabular}
\end{table}
TABLE VIII
ABLATION STUDY ON THE PROPOSED METHOD WITH RESPECT TO ARI, NMI, AND F-SCORE

| Metric        | Algorithm            | Caltech20 | Ecoli | LS | PCT | Aggregation | Texture | UMIST | SPF |
|---------------|----------------------|-----------|-------|----|-----|-------------|---------|-------|-----|
| ARI w/o tr($C^T \Phi C$) | 0.367                | 0.430     | 0.558 | 0.130 | 0.928 | 0.705     | 0.348   | 0.084 |
| Proposed     | 0.525                | 0.754     | 0.657 | 0.179 | 0.989 | 0.723     | 0.371   | 0.105 |
| NMI w/o tr($C^T \Phi C$) | 0.480                | 0.615     | 0.630 | 0.236 | 0.958 | 0.794     | 0.664   | 0.156 |
| Proposed     | 0.497                | 0.690     | 0.677 | 0.273 | 0.978 | 0.813     | 0.685   | 0.197 |
| F-score w/o tr($C^T \Phi C$) | 0.441                | 0.544     | 0.649 | 0.260 | 0.941 | 0.733     | 0.386   | 0.280 |
| Proposed     | 0.603                | 0.823     | 0.729 | 0.322 | 0.991 | 0.750     | 0.411   | 0.348 |

TABLE IX
EXECUTION TIME OF ALGORITHMS INVOLVING AN ITERATION PROCESS ON EIGHT MODEST-SCALE DATASETS. WE HIGHLIGHT THE FASTEST AND THE SECOND FASTEST ONES BY BOLD AND UNDERLINE. EXECUTION TIME OF THE BASELINE ALGORITHM—LWEA IS ADDED TO SHOW HOW MUCH THE PROPOSED METHOD PAID TO IMPROVE THE PERFORMANCE

| Time(s) | RSEC-H [24] | RSEC-Z [24] | DREC [32] | SPCE [18] | TRCE [21] | Proposed | Baseline—LWEA [12] |
|---------|-------------|-------------|-----------|-----------|-----------|----------|---------------------|
| Caltech20 | 700         | 707         | 2.16      | 39.4      | 31.5      | 19.5     | 0.332               |
| Ecoli   | 2.91        | 2.94        | 0.130     | 0.273     | 0.491     | 0.0582   | 0.061               |
| LS      | 14006       | 14107       | 21.3      | 1068      | 354       | 277      | 1.85                |
| PCT     | 3633        | 3663        | 4.24      | 244       | 131       | 60.3     | 0.751               |
| Aggregation | 17.0       | 17.1        | 0.167     | 2.53      | 2.22      | 1.17     | 0.095               |
| Texture | 8207        | 8271        | 6.14      | 409       | 262       | 164      | 1.19                |
| UMIST   | 8.41        | 8.48        | 0.309     | 2.50      | 2.79      | 0.292    | 0.119               |
| SPF     | 285         | 288         | 0.258     | 39.5      | 17.1      | 12.5     | 0.230               |

NMI and F-score in the supplementary file, EAC and PTA-AL will attain the highest ARI/NMI/F-score on 6/5/7 and 6/3/6 datasets over all the compared state-of-the-art algorithms, respectively, while these numbers are only 0/0/0 and 1/0/1 corresponding to their original algorithms. Therefore, we can conclude that the proposed method can enhance diverse CA matrices significantly, illustrating its robustness.

D. Influence of Ensemble Size

Fig. 5 evaluates the clustering performance with respect to the ARI of the proposed method and all the compared methods with different numbers of base clusterings as input. From Fig. 5, we can observe that with a bigger ensemble size, most methods generally perform better, which is consistent with the basic idea of ensemble clustering that combining a set of clustering results can generate a better one. It is noticeable that on five out of eight datasets, our method keeps a strictly monotonic increasing ARI when the ensemble size grows. In contrast, some of the compared methods sometimes experience an opposite trend such as DREC and TRCE. Moreover, it is evident that the proposed method outperforms all the compared methods with different ensemble sizes except case $m = 10$ on Ecoli, being the most robust model among all comparisons. On datasets Caltech20, LS, and FCT, the performance of our method is far ahead of compared methods, and our method even performs better with ten input base clusterings than all the compared methods though they take 40 base clusterings as input.

E. Ablation Study

In this section, we study the necessity of the two terms involved in our method. First, we dropped the Laplacian regularization term, then the proposed model degenerates into (3), and we named this case as "w/o tr($C^T \Phi C$).” Second, we removed the matrix-completion constraint $\Psi_{E}(E) = 0$, naming it as “w/o $\Psi_{E}(E) = 0$.”

The clustering performance without those terms is shown in Table VIII. It is obvious that “w/o tr($C^T \Phi C$)” always receives the worst scores, indicating that Laplacian regularization plays a leading role in propagating HC information. Additionally, the average performance of “w/o $\Psi_{E}(E) = 0$” exceeds that of the proposed method slightly on Texture and UMIST, while falls behind it on the other six datasets distinctly. Thus, it can be concluded that retaining those crucial HC values also contributes to enhancing the CA matrix.

F. Execution Time

In Table IX, we count the average running time of our method with five compared methods that also involve an iteration optimization process, that is, RSEC-H, RSEC-Z, DREC, SPCE, and TRCE, on the eight modest-scale datasets. We also show the running time of LWEA as a reference. Note that LWEA does not need iterative optimization (can be computed analytically), which runs faster than other methods. Table IX suggests that the proposed method is more efficient than RSEC-H, RSEC-Z, SPCE, and TRCE due to the convexity of the proposed model and the low computational complexity of the proposed optimization algorithm. Besides, our method is more time-saving than DREC on the Ecoli and UMIST datasets, but runs slower than DREC when the scale of the dataset gets larger, mainly because DREC uses the microcluster-based representation that reduces the scale of the problem. Considering that our method significantly
outperforms DREC in clustering performance, a little sacrifice in running time is acceptable.

V. CONCLUSION

In this article, we have presented a novel CA-matrix self-enhancement model for ensemble clustering. Without any extra information, our method can promote a CA matrix by exploiting the highly reliable information and denoising error connections simultaneously. We formulated this model as a well-defined convex optimization problem and proposed an ADMM-based algorithm to solve it with convergence and global optimum theoretically guaranteed. Extensive experimental comparisons validate that the proposed model outperforms the state-of-the-art methods significantly in clustering performance. Moreover, it is robust to hyperparameters, can adapt to diverse CA matrices, produces better performance with more base clusterings, and is more efficient than many other approaches.

This article only uses the high-value elements in the CA matrix as the highly reliable information; however, the low-value elements in the CA matrix are also reliable, which indicates that the two samples do not belong to the same cluster with a high probability. In the future, we will investigate how to exploit such information.

REFERENCES

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: A review,” ACM Comput. Surv., vol. 31, no. 3, pp. 264–323, Nov. 1999, doi: 10.1145/331499.331504.
[2] J. A. Hartigan and M. A. Wong, “Algorithm AS 136: A k-means clustering algorithm,” Appl. Statist., vol. 28, no. 1, pp. 100–108, Jan. 1979, doi: 10.2307/2346830.
[3] K. Chen, L. Yao, D. Zhang, X. Wang, X. Chang, and F. Nie, “A semisupervised recurrent convolutional attention model for human activity recognition,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 5, pp. 1747–1756, May 2020.
[4] Y. Cheng, “Mean shift, mode seeking, and clustering,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 17, no. 8, pp. 790–799, Aug. 1995.
[5] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics). Berlin, Germany: Springer-Verlag, 2006.
[6] U. von Luxburg, “A tutorial on spectral clustering,” Statist. Comput., vol. 17, no. 4, pp. 395–416, 2007, doi: 10.1007/s11222-007-9033-z.
[7] Z. Li, F. Nie, X. Chang, Y. Yang, C. Zhang, and N. Sebe, “Dynamic affinity graph construction for spectral clustering using multiple features,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 12, pp. 6232–6332, Dec. 2018.
[8] Z. Li, F. Nie, X. Chang, L. Nie, H. Zhang, and Y. Yang, “Rank-constrained spectral clustering with flexible embedding,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 12, pp. 6073–6082, Dec. 2018.
[9] J. Xie, R. Girshick, and A. Farhadi, “Unsupervised deep embedding for clustering analysis,” in Proc. 33rd Int. Conf. Mach. Learn. (Proceedings of Machine Learning Research), M. F. Balcan and K. Q. Weinberger, Eds., vol. 48, New York, NY, USA: PMLR, Jun. 2016, pp. 478–487. [Online]. Available: http://proceedings.mlr.press/v48/xie16.html
[10] A. Strehle and J. Ghosh, “Cluster ensembles—A knowledge reuse framework for combining multiple partitions,” J. Mach. Learn. Res., vol. 3, pp. 583–617, Jan. 2003, doi: 10.1186/1367-5668-3-29.
[11] M. Ganaie, M. Hu, A. Malik, T. Tanver, and P. Suganthan, “Ensemble deep learning: A review,” Eng. Appl. Artif. Intell., vol. 115, Oct. 2022, Art. no. 105151. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095219762200269X
[12] 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.
[13] A. Fred and A. Jain, “Combining multiple clusterings using evidence accumulation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 6, pp. 835–850, Jun. 2005.
Yuheng Jia (Member, IEEE) received the B.S. degree in automation and the M.S. degree in control theory and engineering from Zhengzhou University, Zhengzhou, China, in 2012 and 2015, respectively, and the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, China, in 2019.

He is currently an Associate Professor with the School of Computer Science and Engineering, Southeast University, Nanjing, China. He is also a Senior Visiting Scholar with the Research Center for Data Mining and Knowledge Discovery, Zhejiang Lab, Hangzhou, China. His research interests broadly include topics in machine learning and data representation, such as semisupervised learning, high-dimensional data modeling and analysis, and low-rank tensor/matrix approximation and factorization.

Sirui Tao is currently pursuing the B.Eng. degree with the School of Automation, Southeast University, Nanjing, China. He will pursue the M.Sc. degree with Academy for Advanced Interdisciplinary Studies, Peking University, Beijing, China. His current research interests lie in machine learning and high-performance computing.

Ran Wang (Member, IEEE) received the B.Eng. degree in computer science from the College of Information Science and Technology, Beijing Forestry University, Beijing, China, in 2009, and the Ph.D. degree from the Department of Computer Science, City University of Hong Kong, Hong Kong, in 2014. From 2014 to 2016, she was a Postdoctoral Researcher with the Department of Computer Science, City University of Hong Kong. She is currently an Associate Professor with the College of Mathematics and Statistics, Shenzhen University, Shenzhen, China. Her current research interests include pattern recognition, machine learning, fuzzy sets and fuzzy logic, and their related applications.

Yongheng Wang received the Ph.D. degree in computer science and technology from the National University of Defense Technology, Changsha, China, in 2006. He is currently a Research Specialist with the Research Center for Data Mining and Knowledge Discovery, Zhejiang Lab, Hangzhou, China. His research interests include big data analytics, machine learning, and intelligent decision-making. His current research is on intelligent interactive data analysis and simulation-based intelligent decision-making in the economic field.