Enhanced Ensemble Clustering via Fast Propagation of Cluster-Wise Similarities

Dong Huang, *Member, IEEE*, Chang-Dong Wang, *Member, IEEE*, Hongxing Peng, Jianhuang Lai, *Senior Member, IEEE*, and Chee-Keong Kwoh, *Senior Member, IEEE*

***Abstract*—Ensemble clustering has been a popular research topic in data mining and machine learning. Despite its significant progress in recent years, there are still two challenging issues in the current ensemble clustering research. First, most of the existing algorithms tend to investigate the ensemble information at the object-level, yet often lack the ability to explore the rich information at higher levels of granularity. Second, they mostly focus on the direct connections (e.g., direct intersection or pair-wise co-occurrence) in the multiple base clusterings, but generally neglect the multiscale indirect relationship hidden in them. To address these two issues, this paper presents a novel ensemble clustering approach based on fast propagation of cluster-wise similarities via random walks. We first construct a cluster similarity graph with the base clusters treated as graph nodes and the cluster-wise Jaccard coefficient exploited to compute the initial edge weights. Upon the constructed graph, a transition probability matrix is defined, based on which the random walk process is conducted to propagate the graph structural information. Specifically, by investigating the propagating trajectories starting from different nodes, a new cluster-wise similarity matrix can be derived by considering the trajectory relationship. Then, the newly obtained cluster-wise similarity matrix is mapped from the cluster-level to the object-level to achieve an enhanced co-association matrix, which is able to simultaneously capture the object-wise co-occurrence relationship as well as the multiscale cluster-wise relationship in ensembles. Finally, two novel consensus functions are proposed to obtain the consensus clustering result. Extensive experiments on a variety of real-world datasets have demonstrated the effectiveness and efficiency of our approach.**

***Index Terms*—Cluster-wise similarity, consensus clustering, data clustering, ensemble clustering, random walk.**

I. Introduction

DATA clustering is an unsupervised learning technique that aims to partition a set of data objects (i.e., data points) into a certain number of homogeneous groups [1]–[10]. It is a fundamental yet very challenging topic in the field of data mining and machine learning, and has been successfully applied in a wide variety of areas, such as image processing [11], [12], community discovery [13], [14], recommender systems [15]–[17], and text mining [18]. In the past few decades, a large number of clustering algorithms have been developed by exploiting various techniques [19]. Different algorithms may lead to very different clustering performances for a specific dataset. Each clustering algorithm has its own advantages as well as weaknesses. However, there is no single algorithm that is suitable for all data distributions and applications. Given a clustering task, it is generally not easy to choose a proper clustering algorithm for it, especially without prior knowledge. Even if a specific algorithm is given, it may still be very difficult to decide the optimal parameters for the clustering task.

Unlike the conventional practice that typically uses a single algorithm to produce a single clustering result, ensemble clustering has recently emerged as a powerful tool whose purpose is to combine multiple different clustering results (generated by different algorithms or the same algorithm with different parameter settings) into a probably better and more robust consensus clustering [20]. Ensemble clustering has been gaining increasing attention, and many ensemble clustering algorithms have been proposed in recent years [21]–[32]. Despite its significant progress, there are still two challenging issues in the current research. First, most of the existing ensemble clustering algorithms investigate the ensemble information at the object-level, and often fail to explore the higher-level information in the ensemble of multiple base clusterings. Second, they mostly focus on the direct relationship in ensembles, such as

Manuscript received July 12, 2018; accepted October 2, 2018. This work was supported in part by the NSFC under Grant 61602189, Grant 61502543, Grant 61573387, and Grant 61876193, in part by the National Key Research and Development Program of China under Grant 2016YFB1001003, in part by the Natural Science Foundation of Guangdong Province under Grant 2016A030310457 and Grant 2016A030306014, in part by the Singapore Ministry of Education Tier-1 Grant under Contract RG21/15, and in part by the Singapore Ministry of Education Tier-2 Grant under Contract MOE2014-T2-2-023. This paper was recommended by Associate Editor G. Nicosia. *(Corresponding author: Dong Huang.)*

D. Huang is with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China, and also with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: huangdonghere@gmail.com).

C.-D. Wang is with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China, and also with the Guangdong Province Key Laboratory of Computational Science, Guangzhou, China (e-mail: changdongwang@hotmail.com).

H. Peng is with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China (e-mail: xyphx@scau.edu.cn).

J. Lai is with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China, also with the Guangdong Key Laboratory of Information Security Technology, Guangzhou, China, and also with the Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China (e-mail: stsljh@mail.sysu.edu.cn).

C.-K. Kwoh is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: asckkwoh@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2018.2876202

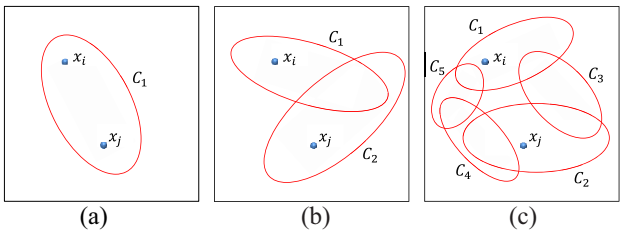


Fig. 1. Relationship between two objects x_i and x_j. (a) If they appear in the same cluster. (b) If they appear in two different but intersected clusters. (c) If they appear in two different clusters that are indirectly connected by some other clusters.

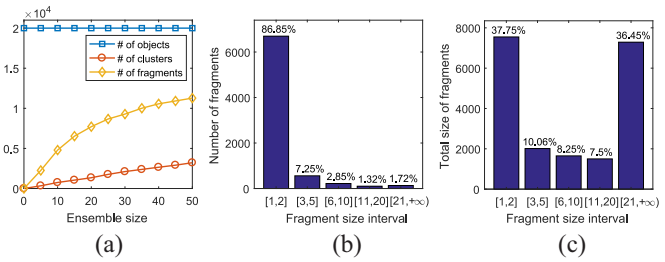


Fig. 2. Statistics of the intersection fragments on the *LR* datasets. (a) Numbers of data objects, clusters, and fragments as the ensemble size M increases from 0 to 50. (b) Number of fragments in each size interval (with $M = 20$). (c) Total size of fragments in each size interval (with $M = 20$).

direct intersections and pair-wise co-occurrence, but generally neglect the multiscale indirect connections in the base clusterings, which may exhibit a negative influence on the robustness of their consensus clustering performances.

In ensemble clustering, the direct co-occurrence relationship between objects is the most basic information. Fred and Jain [20] captured the co-occurrence relationship by presenting the concept of co-association matrix, which reflects how many times two objects occur in the *same* cluster among the multiple base clusterings. The drawback of the conventional co-association matrix lies in that it only considers the direct co-occurrence relationship, yet lacks the ability to take into consideration the rich information of indirect connections in ensembles. As shown in Fig. 1, if two objects occur in the same cluster in a base clustering, then we say these two objects are directly connected. If two objects are in two different clusters *and* the two clusters are directly or indirectly related to each other, then we say these two objects are indirectly connected. The challenge here is twofold.

1) How to improve the object-wise relationship by exploiting the higher-level (e.g., cluster-level) connections.
2) How to explore the direct and indirect structural relationship in a unified model.

To partially address this, Iam-On *et al.* [33] proposed the weighted connected triple (WCT) method to incorporate the common neighborhood information between clusters into the conventional co-association matrix, which exploits the direct neighborhood information between clusters but cannot utilize their indirect neighboring connections. Further, Iam-On *et al.* [34] took advantage of the SimRank similarity (SRS) to investigate the indirect connections for refining the co-association matrix, which, however, suffers from its very high computational cost and is not feasible for large datasets. More recently, Huang *et al.* [25] investigated the ensemble information by performing the random walk on a set of data fragments (also known as microclusters). Specifically, the set of data fragments are generated by intersecting the cluster boundaries of multiple base clusterings, and can be used as a set of basic operating units in the consensus process [25]. Although using data fragments instead of original objects may provide better computational efficiency, the approach in [25] still suffers from two limitations. In one aspect, when the ensemble size (i.e., the number of base clusterings) grows larger, the number of the generated fragments may increase dramatically [as shown in Fig. 2(a)], which eventually leads to a rapidly increasing computational burden. In another aspect, by intersecting the cluster boundaries of multiple base clusterings, the generated fragments may be associated with very *imbalanced* sizes. As an example, we use 20 base clusterings on the letter recognition (*LR*) dataset to generate a set of data fragments. Different intersection fragments may have very different sizes, i.e., they may consist of very different numbers of data objects. The number of the fragments in each size interval is illustrated in Fig. 2(b), while the total size of fragments in each size interval is shown in Fig. 2(c). It can be observed that over 80% of the fragments have a very small size (of 1 or 2), whereas only 1.72% of the total fragments have a size greater than 20. However, the 1.72% of these large fragments surprisingly amounts to as large as 36.45% of the entire set of objects, which shows the heavy imbalance of the fragment sizes and places an unstable factor on the overall consensus process. Despite the efforts that these algorithms have made [25], [33], [34], it remains an open problem how to effectively and efficiently investigate the higher-level ensemble information as well as incorporate multiscale direct and indirect connections in ensembles for enhancing the consensus clustering performance.

To address the aforementioned challenging issues, in this paper, we propose a new ensemble clustering approach based on fast propagation of cluster-wise similarities via random walks. Different from the existing techniques that work at the object-level [20] or the fragment-level [25], in this paper, we explore the rich information of the ensembles at the base-cluster-level with multiscale integration and cluster-object mapping. Specifically, a cluster similarity graph is first constructed by treating the base clusters as graph nodes and using the Jaccard coefficient to build the weighted edges. By defining a transition probability matrix, the random walk process is then performed to explore the multiscale structural information in the cluster similarity graph. Thereafter, a new cluster-wise similarity matrix can be derived by utilizing the random walk trajectories starting from different nodes in the original graph. Further, an enhanced co-association (ECA) matrix is constructed by mapping the newly obtained cluster-wise similarity back to the object-level. Finally, by performing the partitioning process at the object-level and at the cluster-level, respectively, two novel consensus functions are therefore proposed, i.e., ensemble clustering by propagating cluster-wise similarities with hierarchical consensus function (ECPCS-HC) and ensemble clustering by propagating

cluster-wise similarities with meta-cluster-based consensus function (ECPCS-MC). Extensive experiments have been conducted on a variety of real-world datasets, which demonstrate the effectiveness and efficiency of our ensemble clustering approach when compared to the state-of-the-art approaches.

For clarity, the main contributions of this paper are summarized as follows.

1) A new cluster-wise similarity measure is derived, which captures the higher-level ensemble information and incorporates the multiscale indirect connections by means of the random walk process starting from each cluster node.
2) An ECA matrix is presented based on the cluster-object mapping, which simultaneously reflects the object-wise co-occurrence relationship as well as the cluster-wise structural information.
3) Two novel consensus functions are devised, namely ECPCS-HC and ECPCS-MC, which perform the partitioning process at the object-level and at the cluster-level, respectively, to obtain the final consensus clustering.
4) Experiments on multiple datasets have shown the superiority of the proposed approach over the existing ensemble clustering approaches.

The remainder of this paper is organized as follows. Section II reviews the related work on ensemble clustering. Section III provides the formulation of the ensemble clustering problem. Section IV describes the construction of the cluster similarity graph and the random walk propagation for multiscale integration. Section V presents the ECA matrix by mapping cluster-wise similarities to object-wise similarities. Section VI proposes two novel consensus functions in our cluster-wise similarity propagation framework. Section VII reports the experimental results. Finally, Section VIII concludes this paper.

II. Related Work

Ensemble clustering aims to combine a set of multiple base clusterings into a better and more robust consensus clustering result [20]. In the past decade, many ensemble clustering algorithms have been proposed [24]–[26], [29]–[31], [35]–[44], which can be classified into three main categories, namely the pair-wise co-occurrence-based algorithms [20], [38], [41], the graph partitioning-based algorithms [35], [36], [40], and the median partition-based algorithms [24], [37], [39], [42].

The pair-wise co-occurrence-based algorithms [20], [38], [41] typically build a co-association matrix by considering the frequency that two objects occur in the same cluster among the multiple base clusterings. By treating the co-association matrix as the similarity matrix, the hierarchical agglomerative clustering algorithms [19] can be used to obtain the consensus result. Fred and Jain [20] for the first time presented the concept of co-association matrix and designed the evidence accumulation clustering (EAC) method. Then, Li *et al.* [38] extended the EAC method by presenting a new hierarchical agglomerative clustering algorithm that takes the sizes of clusters into consideration via normalized edges. Yi *et al.* [41] dealt with the uncertain entries in the co-association matrix by exploiting the matrix completion technique to improve the robustness of the consensus clustering.

The graph partitioning-based algorithms [35], [36], [40] formulate the clustering ensemble into a graph model and obtain the consensus clustering by segmenting the graph into a certain number of subsets. Strehl and Ghosh [35] treated each cluster in the set of base clusterings as a hyper-edge and proposed three graph partitioning-based ensemble clustering algorithms. Fern and Brodley [36] built a bipartite graph by treating both clusters and objects as graph nodes, which is then partitioned via the METIS algorithm to obtain the consensus clustering. Mimaroglu and Erdil [40] constructed a similarity graph between data objects and partitioned the graph by finding pivots and growing clusters.

The median partition-based algorithms cast the ensemble clustering problem into an optimization problem which aims to find a median partition (or clustering) such that the similarity between this clustering and the set of base clusterings is maximized [24], [37], [39], [42]. To deal with the median partition problem, which is NP-hard [37], Topchy *et al.* [37] utilized the EM algorithm to find an approximate solution for it. Li and Ding [39] formulated the ensemble clustering problem into a non-negative matrix factorization problem and proposed the weighted consensus clustering method. Franek and Jiang [42] cast the ensemble clustering problem into an Euclidean median problem and obtained an approximate solution via the Weiszfeld algorithm [45]. Huang *et al.* [24] formulated the ensemble clustering problem into a binary linear programming problem and solved it via the factor graph model [46].

Despite the fact that significant progress has been made in the ensemble clustering research in recent years [24]–[26], [35]–[42], [47]–[50], there are still two challenging issues in most of the existing algorithms. First, they mostly investigate the ensemble information at the object-level, but often fail to go beyond the object-level to explore the information at higher levels of granularity in the ensemble. Second, many of them only consider the direct connections in ensembles and lack the ability to incorporate the multiscale (indirect) connections for improving the consensus robustness. To (partially) address this, Iam-On *et al.* [33] proposed to refine the co-association matrix by considering the common neighborhood information between clusters, which in fact exploits the one-step indirect connections yet still neglects the multistep (or multiscale) indirect connections in ensembles. Further, Iam-On *et al.* [34] exploited the SRS to incorporate the multiscale neighborhood information in ensembles, which unfortunately suffers from its very high computational cost and is not feasible for large datasets. Huang *et al.* [25] proposed to explore the structural information in ensembles by conducting random walks on the data fragments that are generated by intersecting the cluster boundaries of multiple base clusterings. However, in one aspect, the number of fragments would increase dramatically as the number of base clusterings grows larger, which may bring in a very heavy computational burden [25]. In another aspect, the potentially imbalanced

nature of the fragments [as shown in Fig. 2(b) and (c)] also places an unstable factor on the robustness of the overall consensus clustering process. Moreover, while working at the fragment-level, the approach in [25] still lacks the desired ability to investigate the multiscale cluster-wise relationship in ensembles. Although considerable efforts have been made [25], [33], [34], it remains a very challenging task how to simultaneously tackle the aforementioned two issues effectively and efficiently for the ensemble clustering problem.

III. Problem Formulation

Ensemble clustering is the process of combining multiple base clusterings into a better consensus clustering result. Let $\mathcal{X} = \{x_1, \ldots, x_N\}$ denote a dataset with N objects, where x_i is the ith object. Let $\Pi = \{\pi^1, \ldots, \pi^M\}$ denote a set of M base clusterings for the dataset, where $\pi^m = \{C_1^m, \ldots, C_{n^m}^m\}$ is the mth base clustering, C_i^m is the ith cluster in π^m, and n^m is the number of clusters in π^m.

For clarity, the set of all clusters in the clustering ensemble Π is denoted as $\mathcal{C} = \{C_1, \ldots, C_{N_c}\}$, where C_i is the ith cluster and N_c is the total number of clusters in Π. Obviously, it holds that $N_c = \sum_{m=1}^{M} n^m$.

Formally, the objective of ensemble clustering is to integrate the information of the ensemble of multiple base clusterings in Π to build a better clustering result π^*.

IV. Propagation of Cluster-Wise Similarities

In ensemble clustering, each base clustering consists of a certain number of base clusters. To capture the base cluster information, a commonly used strategy is to map the base cluster labels to the object-level [20] (or fragment-level [25]) by building a co-association matrix, which reflects how many times two objects (or two fragments) are grouped in the same cluster among the multiple base clusterings. The straightforward mapping from the base cluster labels to the object-wise (or fragment-wise) co-association matrix implicitly assumes that different clusters are independent of each other, but fails to consider the potentially rich information hidden in the relationship between different clusters. In light of this, we aim to effectively and efficiently investigate the multiscale direct and indirect relationship between base clusters in the ensemble, so as to achieve better and more robust consensus clustering results. Toward this end, two subproblems here should first be solved.

1) How to define the initial similarity between clusters.
2) How to incorporate the multiscale information to construct more robust cluster-wise similarity.

Since a cluster is a set of data objects, the initial relationship between clusters can be investigated by the Jaccard coefficient [51], which measures the similarity between two sets by considering their intersection size and union size. Formally, the Jaccard coefficient between two clusters (or sets), say, C_i and C_j, is computed as [51]

$$\text{Jaccard}(C_i, C_j) = \frac{|C_i \bigcap C_j|}{|C_i \bigcup C_j|} \tag{1}$$

where $\bigcap$ denotes the intersection of two sets, $\bigcup$ denotes the union of two sets, and $|\cdot|$ denotes the number of objects in a set. By adopting the Jaccard coefficient as the similarity measure between clusters, an initial cluster similarity graph is constructed for the ensemble with each cluster treated as a graph node. That is,

$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\} \tag{2}$$

where $\mathcal{V} = \mathcal{C}$ is the node set and $\mathcal{E}$ is the edge set in the graph $\mathcal{G}$. The weight of an edge between two nodes $C_i, C_j \in \mathcal{V}$ is computed as

$$e_{ij} = \text{Jaccard}(C_i, C_j). \tag{3}$$

With the initial similarity graph constructed, the next step is to incorporate the multiscale information in the graph to enhance the cluster-wise similarity. In particular, the random walk process is performed on the graph, which is a dynamic process that transits from a node to one of its neighbors at each step with a certain probability [14], [25], [52]–[55]. It is a crucial task in random walk to construct the transition probability matrix, which decides the probability of the random walker transiting from a node to another one. In this paper, the transition probability matrix $P = \{p_{ij}\}_{N\times N}$ on the graph is computed as follows:

$$p_{ij} = \begin{cases} \frac{e_{ij}}{\sum_{C_k \neq C_i} e_{ik}}, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases} \tag{4}$$

where p_{ij} is the probability that a random walker transits from nodes C_i to C_j in one step, which is proportional to the edge weight between them. Based on the one-step transition probability matrix, we can obtain the multistep transition probability matrix $P^{(t)} = \{p_{ij}^{(t)}\}_{N\times N}$ for the random walkers on the graph. That is,

$$P^{(t)} = \begin{cases} P, & \text{if } t = 1 \\ P^{(t-1)} \cdot P, & \text{if } t > 1. \end{cases} \tag{5}$$

Note that the (i, j)th entry in $P^{(t)}$, i.e., $p_{ij}^{(t)}$, denotes the probability of a random walker transiting from node C_i to node C_j in t steps. We denote the ith row in $P^{(t)}$ as $P_{i:}^{(t)} = \{p_{i1}^{(t)}, p_{i2}^{(t)}, \ldots, p_{iN}^{(t)}\}$, which represents the probability distribution of a random walker transiting from C_i to all the other nodes in t steps. As different step-lengths of random walkers can reflect the graph structure information at different scales [25], [55], to capture the multiscale information in the graph $\mathcal{G}$, the random walk trajectories at different steps are exploited here to refine the cluster-wise similarity.

Formally, for the random walker starting from a node C_i, its random walk trajectory from steps 1 to t is denoted as $P_{i:}^{(1:t)} = \{P_{i:}^{(1)}, P_{i:}^{(2)}, \ldots, P_{i:}^{(t)}\}$. Obviously, the t-step random walk trajectory (i.e., $P_{i:}^{(1:t)}$), starting from node C_i and having a step-length t, is an $N \cdot t$-tuple, which captures the multiscale (or multistep) structural information in the neighborhood of C_i. With the random walk trajectory of each node obtained, a new similarity measure can thereby be derived for every two nodes by considering the similarity of their random walk

trajectories. Specifically, the new similarity matrix between all of the clusters in Π is represented as

$$Z = \left\{z_{ij}\right\}_{N_c \times N_c} \tag{6}$$

where

$$z_{ij} = \mathrm{Sim}\left(P_{i:}^{(1:t)}, P_{j:}^{(1:t)}\right) \tag{7}$$

denotes the new similarity between two clusters C_i and C_j. Note that $\mathrm{Sim}(\cdot,\cdot)$ can be any similarity measure between two vectors. In this paper, the cosine similarity [56] is adopted. Thus, the new similarity measure between C_i and C_j can be computed as

$$z_{ij} = \frac{< P_{i:}^{(1:t)}, P_{j:}^{(1:t)} >}{\sqrt{< P_{i:}^{(1:t)}, P_{i:}^{(1:t)} > \cdot < P_{j:}^{(1:t)}, P_{j:}^{(1:t)} >}} \tag{8}$$

where $<\cdot,\cdot>$ outputs the dot product of two vectors. Since the entries in the transition probability matrix are always non-negative, it holds that $z_{ij} \in [0, 1]$ for any two clusters C_i and C_j in Π.

V. Enhanced Co-Association Matrix Based on Similarity Mapping

Having obtained the new cluster-wise similarity matrix Z, we proceed to map the new similarity matrix from the cluster-level to the object-level, and describe the enhanced co-association (ECA) representation in this section.

The conventional co-association matrix [20] is a widely used data structure to capture the object-wise similarity in the ensemble clustering problem. Given the clustering ensemble Π, the (direct) pair-wise relationship in the mth base clustering (i.e., π^m) can be represented by a connectivity matrix, which is computed as follows:

$$A^m = \left\{a_{ij}^m\right\}_{N \times N} \tag{9}$$

$$a_{ij}^m = \begin{cases} 1, & \text{if } \mathrm{Cls}^m(x_i) = \mathrm{Cls}^m\left(x_j\right) \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $\mathrm{Cls}^m(x_i)$ denotes the cluster in π^m that contains the object x_i. Obviously, if $C_j \in \pi^m$ and $x_i \in C_j$, then $\mathrm{Cls}^m(x_i) = C_j$. Then, the conventional co-association matrix $A = \{a_{ij}\}_{N \times N}$ for the entire ensemble is computed as follows:

$$A = \frac{1}{M}\sum_{m=1}^{M} A^m. \tag{11}$$

The conventional co-association matrix reflects the number of times that two objects appear in the same cluster among the multiple base clusterings. Although it is able to exploit the (direct) cluster-level information by investigating the object-wise co-occurrence relationship, it inherently treats each cluster as an independent entity and, however, neglects the potential relationship between *different* clusters, which may provide rich information for further refining the object-wise connections. In light of this, with the multiscale cluster-wise relationship explored by random walks in Section IV, the key problem in this section is how to map the multiscale cluster-wise relationship back to the object-level.

In particular, we present an ECA matrix to simultaneously capture the object-wise co-occurrence relationship and the multiscale cluster-wise similarity. Before the construction of the ECA matrix for the entire ensemble, we first take advantage of the newly designed cluster-wise similarity matrix Z to build the enhanced connectivity matrix for a single base clusterings, say, π^m. That is,

$$B^m = \left\{b_{ij}^m\right\}_{N \times N} \tag{12}$$

$$b_{ij}^m = \begin{cases} 1, & \text{if } \mathrm{Cls}^m(x_i) = \mathrm{Cls}^m\left(x_j\right) \\ z_{uv}, & \text{if } \mathrm{Cls}^m(x_i) \neq \mathrm{Cls}^m\left(x_j\right) \end{cases} \tag{13}$$

with

$$\mathrm{Cls}^m(x_i) = C_u^m, \mathrm{Cls}^m\left(x_j\right) = C_v^m. \tag{14}$$

Note that the (i,j)th entry in B^m and the (i,j)th entry in A^m will be the same *only when* x_i and x_j occur in the same cluster in π^m. The difference between B^m and A^m arises when two objects belongs to different clusters in a base clustering, in which situation the conventional connectivity matrix A^m lacks the ability to go beyond the direct co-occurrence relationship to exploit further cluster-wise connections. Different from the convectional connectivity matrix, when two objects belong to two different clusters in a base clustering, the enhanced connectivity matrix B^m is still able to capture their indirect relationship by investigating the correlation between the two clusters that these two objects, respectively, belong to.

With the enhanced connectivity matrix for each base clustering constructed, the ECA matrix, denoted as $B = \{b_{ij}\}_{N \times N}$, for the entire ensemble Π can be computed as follows:

$$B = \frac{1}{M}\sum_{m=1}^{M} B^m. \tag{15}$$

With $z_{ij} \in [0,\ 1]$, it is obvious that all entries in the ECA matrix are in the range of $[0,\ 1]$. By the construction of the ECA matrix, the cluster-wise similarity in Z is mapped from the cluster-level to the object-level. It is noteworthy that the ECA matrix can be utilized in any co-association matrix-based consensus functions. In particular, two new consensus functions will be designed in the next section.

VI. Two Types of Consensus Functions

In this section, we propose two consensus functions to obtain the final consensus clustering in the proposed ensemble clustering by propagating cluster-wise similarities (ECPCS) framework. The first consensus function is described in Section VI-A, which is based on hierarchical clustering (HC) and performs the partitioning process at the object-level, while the second consensus function is described in Section VI-B, which is based on meta-clustering (MC) and performs the partitioning process at the cluster-level.

A. ECPCS-HC

In this section, we describe our first consensus function termed ECPCS-HC. By treating the ECA matrix as the new object-wise similarity matrix, the hierarchical agglomerative clustering can be performed to obtain the consensus clustering

in an iterative region merging fashion. The original objects are viewed as the set of initial regions, that is,

$$\mathcal{R}^{(0)} = \left\{R_1^{(0)}, \ldots, R_N^{(0)}\right\} \tag{16}$$

where $R_i^{(0)} = \{x_i\}$ denotes the ith initial region that contains exactly one object x_i. The similarity matrix for the set of initial regions is defined as

$$S^{(0)} = \left\{s_{ij}^{(0)}\right\}_{N\times N} \tag{17}$$

$$s_{ij}^{(0)} = b_{ij}. \tag{18}$$

With the initial region set and its similarity matrix obtained, the region merging process is then performed iteratively. In each iteration, the two regions with the highest similarity are merged into a new and larger region, which will be followed by the update of the region set and the corresponding similarity matrix. Specifically, the updated region set after the qth iteration is denoted as

$$\mathcal{R}^{(q)} = \left\{R_1^{(q)}, \ldots, R_{|\mathcal{R}^{(q)}|}^{(q)}\right\} \tag{19}$$

where $R_i^{(q)}$ is the ith region and $|\mathcal{R}^{(q)}|$ is the number of regions in $\mathcal{R}^{(q)}$.

The similarity matrix after the qth iteration is updated according to the average-link. That is,

$$S^{(q)} = \left\{s_{ij}^{(q)}\right\}_{|\mathcal{R}^{(q)}|\times|\mathcal{R}^{(q)}|} \tag{20}$$

$$s_{ij}^{(q)} = \frac{1}{\left|R_i^{(q)}\right| \cdot \left|R_j^{(q)}\right|} \sum_{x_u\in R_i^{(q)}, x_v\in R_j^{(q)}} b_{uv} \tag{21}$$

where $|R_i^{(q)}|$ denotes the number of objects in $R_i^{(q)}$.

Note that in each iteration the number of regions decreases by one, i.e., $|\mathcal{R}^{(q+1)}| = |\mathcal{R}^{(q)}| - 1$. Since the number of the initial regions is N, it is obvious that all objects will be merged into a root region after totally $N-1$ iterations. As the result of the region merging process, a dendrogram (i.e., a HC tree) will be iteratively constructed. Each level of the dendrogram corresponds to a clustering result with a certain number of clusters. By choosing a level in the dendrogram, the final consensus clustering can thereby be obtained.

B. ECPCS-MC

In this section, we describe our second consensus function termed ECPCS-MC. Different from ECPCS-HC, the ECPCS-MC method performs the partitioning process at the cluster-level, which takes advantage of the enhanced cluster-wise similarity matrix Z and groups all the clusters in the ensemble into several subsets. Each subset of clusters is referred to as a meta-cluster. Then, each data object is assigned to one of the meta-clusters by majority voting to construct the final consensus clustering.

Specifically, by treating the clusters in the ensemble as graph nodes and using the cluster-wise similarity matrix Z to define the edge weights between them, a new cluster similarity graph can be constructed. That is,

$$\tilde{\mathcal{G}} = \left\{\mathcal{V}, \tilde{\mathcal{E}}\right\} \tag{22}$$

where $\mathcal{V} = \mathcal{C}$ is the node set and $\tilde{\mathcal{E}}$ is the edge set. The edge weights in the graph $\tilde{\mathcal{G}}$ are decided by the enhanced cluster-wise similarity matrix B. Given two clusters C_i and C_j, the weight between them is defined as

$$\tilde{e}_{ij} = b_{ij}. \tag{23}$$

Then, the normalized cut algorithm [11] can be used to partition the new graph into a certain number of meta-clusters, that is,

$$\mathcal{MC} = \{\mathrm{MC}_1, \mathrm{MC}_2, \ldots, \mathrm{MC}_k\} \tag{24}$$

where MC_i is the ith meta-cluster and k is the number of meta-clusters.

Note that a meta-cluster consists of a certain number of clusters. Given an object x_i and a meta-cluster MC_j, the object x_i may appear in *zero or more* clusters inside MC_j. Specifically, the voting score of x_i with respect to the meta-cluster MC_j can be defined as the proportion of the clusters in MC_j that contain x_i. That is,

$$\begin{aligned}\mathrm{Score}\big(x_i, \mathrm{MC}_j\big) &= \frac{1}{|\mathrm{MC}_j|} \sum_{C_l\in\mathrm{MC}_j} \mathbf{1}(x_i \in C_l)\\ \mathbf{1}(\text{statement}) &= \begin{cases} 1, & \text{if statement is true}\\ 0, & \text{otherwise}\end{cases}\end{aligned} \tag{25}$$

where $|\mathrm{MC}_j|$ denotes the number of clusters in MC_j.

Then, by majority voting, each object is assigned to the meta-cluster in which it appears most frequently (i.e., with the highest voting score). That is,

$$\mathrm{MetaCls}(x_i) = \arg\max_{\mathrm{MC}_j\in\mathcal{MC}} \mathrm{Score}\big(x_i, \mathrm{MC}_j\big). \tag{26}$$

If an object obtains the same highest voting score from two or more different meta-clusters (which in practice rarely happens), then the object will be randomly assigned to one of the winning meta-clusters. By assigning each object to a meta-cluster via majority voting and treating the objects in the same meta-cluster as a consensus cluster, the final consensus clustering result can therefore be obtained.

VII. Experiments

In this section, we conduct experiments on a variety of benchmark datasets to evaluate the performance of the proposed ECPCS-HC and ECPCS-MC algorithms against several state-of-the-art ensemble clustering algorithms.

A. Datasets and Evaluation Measures

In our experiments, ten benchmark datasets are used, i.e., *Breast Cancer* (*BC*), *Cardiotocography* (*CTG*), *Ecoli*, *Gisette*, *LR*, *Landsat Satellite* (*LS*), *MNIST*, *Pen Digits* (*PD*), *Wine*, and *Yeast*. The *MNIST* dataset is from [57], while the other nine datasets are from the UCI machine learning repository [58]. The detailed information of the benchmark datasets is given in Table I.

To quantitatively evaluate the clustering results, two widely used evaluation measures are adopted, namely normalized mutual information (NMI) [35] and adjusted Rand index

TABLE I
DESCRIPTION OF THE BENCHMARK DATASETS

Dataset	#Object	#Class	Dimension
BC	683	2	9
CTG	2,126	10	21
Ecoli	336	8	7
Gisette	7,000	2	5,000
LR	20,000	26	16
LS	6,435	6	36
MNIST	5,000	10	784
PD	10,992	10	16
Wine	178	3	13
Yeast	1,484	10	8

(ARI) [59]. Note that large values of NMI and ARI indicate better clustering results.

The NMI evaluates the similarity between two clusterings from an information theory perspective [35]. Let π' be a test clustering and π^G be the ground-truth clustering. The NMI between π' and π^G is computed as follows [35]:

$$\mathrm{NMI}\left(\pi', \pi^G\right) = \frac{\sum_{i=1}^{n'} \sum_{j=1}^{n^G} n_{ij} \log \frac{n_{ij} n}{n'_i n^G_j}}{\sqrt{\sum_{i=1}^{n'} n'_i \log \frac{n'_i}{n} \sum_{j=1}^{n^G} n^G_j \log \frac{n^G_j}{n}}} \quad (27)$$

where n' is the cluster number in π', n^G is the cluster number in π^G, n'_i is the number of objects in the cluster i of π', n^G_j is the number of objects in the cluster j of π^G, and n_{ij} is the size of the intersection of the cluster i of π' and the cluster j of π^G.

The ARI is an evaluation measure that takes into consideration the number of object-pairs upon which two clusterings agree (or disagree) [59]. Formally, the ARI between two clusterings π' and π^G is computed as follows [59]:

$$\mathrm{ARI}\left(\pi', \pi^G\right) = \frac{2(N_{00}N_{11} - N_{01}N_{10})}{(N_{00} + N_{01})(N_{01} + N_{11}) + (N_{00} + N_{10})(N_{10} + N_{11})} \quad (28)$$

where N_{11} is the number of object-pairs that belong to the same cluster in both π' and π^G, N_{00} is the number of object-pairs that belong to different clusters in both π' and π^G, N_{10} is the number of object-pairs that belong to the same cluster in π' while belonging to different clusters in π^G, and N_{01} is the number of object-pairs that belong to different clusters in π' while belonging to the same cluster in π^G.

B. Baseline Methods and Experimental Settings

Our proposed ECPCS-HC and ECPCS-MC methods will be compared against eight baseline ensemble clustering methods, which are listed as follows.

1) *EAC [20]:* Evidence accumulation clustering.
2) *MCLA [35]:* MC algorithm.
3) *SRS [34]:* SimRank similarity-based method.
4) *WCT [33]:* Weighted connected triple method.
5) *KCC [23]: k*-means-based consensus clustering.
6) *PTGP [25]:* Probability trajectory-based graph partitioning.
7) *ECC [44]:* Entropy-based consensus clustering.
8) *SEC [30]:* Spectral ensemble clustering.

The parameters in the baseline methods are set as suggested by their corresponding papers [20], [23], [25], [30], [33]–[35], [44]. The step-length parameter t in the proposed methods is set to 20 for the experiments on all datasets, whose sensitivity will be further evacuated in Section VII-E.

To provide a fair comparison, we run each of the test methods 20 times on each dataset, and report their average NMI and ARI scores over multiple runs. At each run, an ensemble of $M = 20$ base clusterings is constructed by the k-means clustering with initial cluster centers randomly initialized and the number of clusters in each base clustering randomly selected in the range of $[K, \min(\sqrt{N},100)]$, where K is the number of classes and N is the number of objects in the dataset. Moreover, the performances of these test methods using different ensemble size M will be further evaluated in Section VII-D.

C. Comparison With Other Ensemble Clustering Methods

In this section, we compare the proposed ECPCS-HC and ECPCS-MC methods against the baseline ensemble clustering methods on the ten benchmark datasets. For the experiment on each benchmark dataset, two criteria are adopted in terms of the number of clusters, that is, true-k and best-k. In the true-k criterion, the true number of classes in a dataset is used as the cluster number for all the test methods. In the best-k criterion, the cluster number that leads to the best performance is used for each test method.

Table II reports the average NMI scores (over 20 runs) by different ensemble clustering methods. As shown in Table II, our ECPCS-HC method obtains the best performance with respect to NMI in terms of both true-k and best-k on the *BC*, *Ecoli*, *Wine*, and *Yeast* datasets, whereas ECPCS-MC achieves the best NMI scores in terms of both true-k and best-k on the *CTG*, *LR*, *LS*, and *PD* datasets. Note that, with two comparisons (with respect to true-k and best-k, respectively) on each of ten datasets, there are totally 20 comparisons in Table II. As shown in Fig. 3(a), our ECPCS-HC and ECPCS-MC methods are ranked in the first position in ten and nine comparisons, respectively, out of the totally 20 comparisons, while the third best method (i.e., PTGP) is ranked in the first position in only two comparisons. Similarly, as shown in Fig. 3(b), ECPCS-HC and ECPCS-MC are ranked in the top three in 17 and 20 comparisons, respectively, out of the totally 20 comparisons, while the third best method PTGP is only able to be ranked in the top three in eight comparisons.

Table III reports the average ARI scores (over 20 runs) by different ensemble clustering methods. As shown in Table III, the highest ARI scores are achieved by either ECPCS-HC or ECPCS-MC in sixteen comparisons out of the totally 20 comparisons. Specifically, as shown in Fig. 4(a), with respect to the average ARI scores, ECPCS-HC and ECPCS-MC are ranked in the first position in nine and seven comparisons, respectively, out of the totally 20 comparisons, while the third best methods is ranked in the first position in only two comparisons. As shown in Fig. 4(b), both ECPCS-HC and ECPCS-MC are

TABLE II
AVERAGE NMI (%) SCORES (OVER 20 RUNS) BY DIFFERENT ENSEMBLE CLUSTERING METHODS. THE BEST THREE SCORES IN EACH COMPARISON ARE HIGHLIGHTED IN BOLD, WHILE THE BEST ONE IN [BOLD AND BRACKETS]

Dataset		EAC	MCLA	SRS	WCT	KCC	PTGP	ECC	SEC	ECPCS-MC	ECPCS-HC
BC	True-k	73.00$_{\pm 6.57}$	77.63$_{\pm 2.39}$	72.46$_{\pm 5.32}$	76.32$_{\pm 5.43}$	76.18$_{\pm 10.22}$	76.09$_{\pm 4.14}$	**79.07**$_{\pm 1.86}$	45.26$_{\pm 26.26}$	**77.89**$_{\pm 1.47}$	[**79.46**$_{\pm 3.47}$]
	Best-k	73.34$_{\pm 5.64}$	77.63$_{\pm 2.39}$	72.56$_{\pm 5.01}$	76.33$_{\pm 5.42}$	76.59$_{\pm 8.20}$	76.09$_{\pm 4.14}$	**79.07**$_{\pm 1.86}$	54.58$_{\pm 16.93}$	**77.89**$_{\pm 1.47}$	[**79.46**$_{\pm 3.47}$]
CTG	True-k	**26.16**$_{\pm 0.85}$	24.71$_{\pm 0.92}$	25.85$_{\pm 0.77}$	26.15$_{\pm 1.00}$	23.28$_{\pm 1.73}$	25.15$_{\pm 1.11}$	23.58$_{\pm 1.27}$	24.41$_{\pm 1.80}$	[**26.87**$_{\pm 0.91}$]	**26.42**$_{\pm 1.42}$
	Best-k	26.47$_{\pm 0.80}$	25.63$_{\pm 0.66}$	26.27$_{\pm 0.81}$	**26.66**$_{\pm 0.85}$	25.07$_{\pm 0.83}$	26.38$_{\pm 0.81}$	24.78$_{\pm 0.65}$	25.44$_{\pm 0.85}$	[**27.60**$_{\pm 0.76}$]	**26.99**$_{\pm 0.87}$
Ecoli	True-k	58.33$_{\pm 2.91}$	49.17$_{\pm 2.92}$	56.79$_{\pm 2.42}$	**62.20**$_{\pm 3.80}$	49.64$_{\pm 2.84}$	50.28$_{\pm 2.23}$	50.61$_{\pm 1.97}$	51.76$_{\pm 3.81}$	**59.54**$_{\pm 1.75}$	[**70.48**$_{\pm 2.51}$]
	Best-k	68.02$_{\pm 3.36}$	52.68$_{\pm 1.58}$	67.40$_{\pm 2.79}$	**70.98**$_{\pm 1.86}$	54.68$_{\pm 3.19}$	60.63$_{\pm 2.99}$	57.63$_{\pm 2.30}$	55.01$_{\pm 3.73}$	**69.93**$_{\pm 2.01}$	[**71.45**$_{\pm 0.96}$]
Gisette	True-k	27.02$_{\pm 13.60}$	**41.69**$_{\pm 12.52}$	35.09$_{\pm 9.52}$	37.79$_{\pm 8.35}$	17.26$_{\pm 12.74}$	[**47.13**$_{\pm 1.94}$]	29.15$_{\pm 10.08}$	12.10$_{\pm 7.97}$	**47.01**$_{\pm 2.23}$	40.42$_{\pm 8.25}$
	Best-k	31.18$_{\pm 8.78}$	**43.13**$_{\pm 8.37}$	35.77$_{\pm 8.39}$	38.37$_{\pm 7.17}$	23.13$_{\pm 7.57}$	[**47.13**$_{\pm 1.94}$]	30.41$_{\pm 7.64}$	17.83$_{\pm 5.70}$	**47.01**$_{\pm 2.23}$	41.00$_{\pm 7.05}$
LR	True-k	38.30$_{\pm 0.90}$	38.60$_{\pm 1.17}$	38.40$_{\pm 1.10}$	38.48$_{\pm 1.09}$	34.87$_{\pm 0.95}$	**39.16**$_{\pm 1.17}$	35.72$_{\pm 0.83}$	33.13$_{\pm 1.44}$	[**39.30**$_{\pm 0.74}$]	**38.73**$_{\pm 1.44}$
	Best-k	41.64$_{\pm 0.54}$	40.53$_{\pm 0.62}$	42.14$_{\pm 0.62}$	**42.49**$_{\pm 0.58}$	38.78$_{\pm 0.66}$	41.85$_{\pm 0.60}$	39.22$_{\pm 0.69}$	38.88$_{\pm 0.76}$	[**42.85**$_{\pm 0.55}$]	[**42.85**$_{\pm 0.84}$]
LS	True-k	60.86$_{\pm 3.73}$	53.58$_{\pm 3.16}$	62.00$_{\pm 3.80}$	62.13$_{\pm 2.59}$	48.46$_{\pm 3.67}$	**62.45**$_{\pm 1.33}$	52.39$_{\pm 4.21}$	43.57$_{\pm 6.97}$	[**63.90**$_{\pm 2.36}$]	**63.18**$_{\pm 2.55}$
	Best-k	62.17$_{\pm 2.17}$	54.50$_{\pm 2.30}$	62.96$_{\pm 1.59}$	**63.79**$_{\pm 1.61}$	51.35$_{\pm 2.41}$	63.09$_{\pm 1.18}$	53.55$_{\pm 3.33}$	49.75$_{\pm 3.58}$	[**65.02**$_{\pm 1.86}$]	**64.86**$_{\pm 1.34}$
MNIST	True-k	61.94$_{\pm 1.81}$	58.26$_{\pm 3.53}$	**62.63**$_{\pm 1.82}$	62.44$_{\pm 1.73}$	50.90$_{\pm 2.77}$	**63.59**$_{\pm 2.51}$	50.02$_{\pm 2.68}$	45.74$_{\pm 4.32}$	[**63.81**$_{\pm 2.15}$]	60.26$_{\pm 1.62}$
	Best-k	62.47$_{\pm 1.84}$	58.60$_{\pm 3.11}$	62.99$_{\pm 1.84}$	63.73$_{\pm 1.71}$	54.13$_{\pm 1.90}$	**64.84**$_{\pm 1.94}$	53.54$_{\pm 1.33}$	55.12$_{\pm 1.89}$	**64.40**$_{\pm 1.81}$	[**65.00**$_{\pm 1.36}$]
PD	True-k	73.63$_{\pm 2.16}$	70.20$_{\pm 3.37}$	74.57$_{\pm 2.67}$	74.61$_{\pm 3.13}$	60.77$_{\pm 3.74}$	**74.80**$_{\pm 3.38}$	62.36$_{\pm 2.67}$	51.75$_{\pm 7.58}$	[**76.41**$_{\pm 2.28}$]	**74.91**$_{\pm 3.24}$
	Best-k	76.87$_{\pm 1.24}$	71.01$_{\pm 2.75}$	77.75$_{\pm 1.38}$	78.51$_{\pm 1.72}$	67.63$_{\pm 1.94}$	**79.11**$_{\pm 1.54}$	67.55$_{\pm 1.50}$	67.44$_{\pm 2.11}$	[**79.79**$_{\pm 1.24}$]	**78.84**$_{\pm 1.65}$
Wine	True-k	86.34$_{\pm 2.60}$	82.25$_{\pm 3.16}$	**88.05**$_{\pm 2.88}$	87.33$_{\pm 3.11}$	86.01$_{\pm 3.69}$	86.85$_{\pm 2.51}$	83.29$_{\pm 7.10}$	86.10$_{\pm 4.13}$	**87.85**$_{\pm 2.36}$	[**88.82**$_{\pm 2.82}$]
	Best-k	86.34$_{\pm 2.60}$	82.25$_{\pm 3.16}$	**88.05**$_{\pm 2.88}$	87.33$_{\pm 3.11}$	86.06$_{\pm 3.41}$	86.85$_{\pm 2.51}$	83.68$_{\pm 6.19}$	86.13$_{\pm 4.01}$	**87.85**$_{\pm 2.36}$	[**88.84**$_{\pm 2.79}$]
Yeast	True-k	26.21$_{\pm 1.33}$	22.12$_{\pm 1.15}$	26.03$_{\pm 1.04}$	**28.36**$_{\pm 1.39}$	21.54$_{\pm 2.59}$	23.42$_{\pm 1.21}$	19.53$_{\pm 0.72}$	22.02$_{\pm 2.01}$	**27.71**$_{\pm 1.09}$	[**29.62**$_{\pm 1.21}$]
	Best-k	28.44$_{\pm 1.64}$	23.15$_{\pm 1.01}$	28.05$_{\pm 1.66}$	**29.83**$_{\pm 1.09}$	23.37$_{\pm 0.98}$	27.76$_{\pm 1.36}$	24.30$_{\pm 0.91}$	23.49$_{\pm 1.01}$	**29.30**$_{\pm 0.87}$	[**30.05**$_{\pm 0.97}$]
Avg. score	True-k	53.18	51.82	54.19	55.58	46.89	54.89	48.57	41.58	57.03	57.23
	Best-k	55.69	52.91	56.39	57.80	50.08	57.37	51.37	47.37	59.16	58.93
Avg. rank	True-k	5.70	6.60	5.10	3.90	8.50	4.30	7.90	8.80	1.90	2.30
	Best-k	5.70	7.20	5.20	3.60	8.50	4.30	7.90	8.70	2.10	1.70

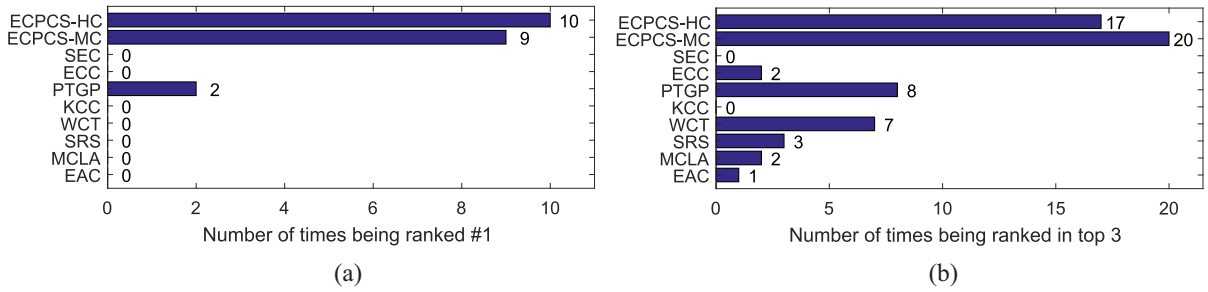


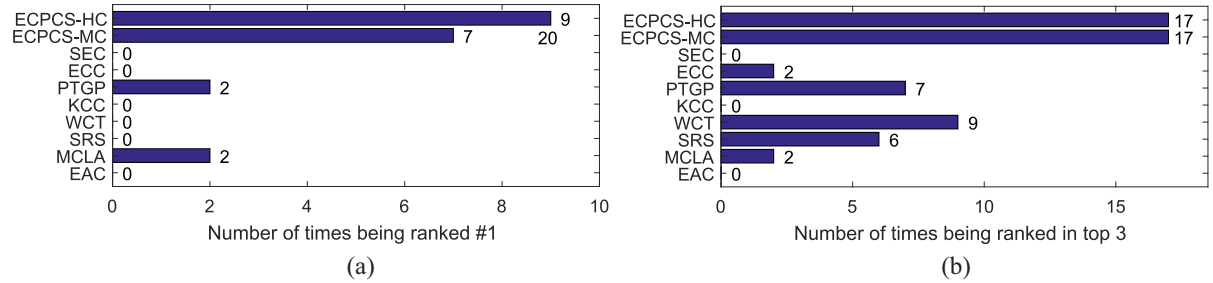
Fig. 3. Number of times that each method is ranked in the (a) first position and (b) top three with respect to Table II.

ranked in the top three in 17 comparisons out of the totally 20 comparisons, while the third best method WCT is ranked in the top three in only nine comparisons.

Additionally, the summary statistics (i.e., average score and average rank) of the experimental results are also provided in the bottom rows of Tables II and III. The average score is computed by averaging the NMI (or ARI) scores of each method across the ten benchmark datasets, whereas the average rank is obtained by averaging the ranking positions of each method across the ten benchmark datasets. As shown in Table II, in terms of true-k, our ECPCS-HC method achieves the highest average NMI (%) score of 57.23, across the ten datasets, while ECPCS-MC achieves the second highest average score of 57.03. In terms of best-k, the highest two average NMI scores across the ten datasets are also obtained by the proposed ECPCS-MC and ECPCS-HC methods, respectively. When considering the average rank, ECPCS-MC and ECPCS-HC achieve the best and the second best average ranks of 1.90 and 2.30, respectively, in terms of true-k, which are significantly better than the third best method (i.e., WCT), whose average rank in terms of true-k is 3.90. In terms of best-k, ECPCS-HC and ECPCS-MC are also the best two methods with respect to the average rank across the ten datasets. Besides the performance with respect to NMI, similar advantages can also be observed in terms of average score and average rank with respect to ARI (as shown in Table III). Moreover, it is interesting to compare ECPCS-HC against EAC and to compare ECPCS-MC against MCLA. Since EAC is a classical method based on the conventional co-association matrix, the comparison between the proposed ECPCS-HC method (which typically incorporates the multiscale cluster-level information via the ECA matrix) and the EAC method provides a straightforward view as to how the proposed ECA matrix improves the consensus performance when compared to the original co-association matrix. Specifically, the average NMI (%) and ARI (%) scores (in terms of true-k) of EAC are, respectively, 53.18 and 46.13, whereas that of ECPCS-HC are, respectively, 57.23 and 52.60. Similar improvements can also be observed when considering the best-k situation (as shown in Tables II and III). Besides ECPCS-HC versus EAC, the ECPCS-MC versus MCLA comparison also provides a view as to what influence the multiscale cluster-level information has upon the conventional meta-cluster-based method. Note that both ECPCS-MC and MCLA are meta-cluster-based methods, the integration of cluster-wise similarity propagation is able to bring in significant improvements for the ECPCS-MC method

TABLE III
AVERAGE ARI (%) SCORES (OVER 20 RUNS) BY DIFFERENT ENSEMBLE CLUSTERING METHODS. THE BEST THREE SCORES IN EACH COMPARISON ARE HIGHLIGHTED IN BOLD, WHILE THE BEST ONE IN [BOLD AND BRACKETS]

Dataset		EAC	MCLA	SRS	WCT	KCC	PTGP	ECC	SEC	ECPCS-MC	ECPCS-HC
BC	True-k	83.11$_{\pm 5.29}$	87.09$_{\pm 1.57}$	82.92$_{\pm 4.19}$	85.55$_{\pm 3.73}$	84.42$_{\pm 14.33}$	85.70$_{\pm 2.90}$	**87.68**$_{\pm 1.27}$	46.48$_{\pm 35.22}$	**87.20**$_{\pm 1.06}$	[**87.81**$_{\pm 2.25}$]
	Best-k	84.95$_{\pm 3.36}$	87.09$_{\pm 1.57}$	83.19$_{\pm 3.67}$	**87.58**$_{\pm 1.38}$	85.50$_{\pm 8.99}$	85.70$_{\pm 2.90}$	**87.68**$_{\pm 1.27}$	62.21$_{\pm 19.35}$	87.20$_{\pm 1.06}$	[**88.55**$_{\pm 0.84}$]
CTG	True-k	12.23$_{\pm 0.81}$	11.55$_{\pm 0.74}$	12.45$_{\pm 0.76}$	**12.66**$_{\pm 1.02}$	10.64$_{\pm 1.39}$	11.93$_{\pm 0.96}$	11.10$_{\pm 1.06}$	11.58$_{\pm 1.54}$	[**13.05**$_{\pm 0.90}$]	**12.68**$_{\pm 1.19}$
	Best-k	12.95$_{\pm 1.20}$	13.18$_{\pm 0.68}$	13.13$_{\pm 1.17}$	13.40$_{\pm 1.12}$	11.69$_{\pm 0.73}$	**13.97**$_{\pm 1.11}$	11.69$_{\pm 0.75}$	12.24$_{\pm 0.95}$	[**15.58**$_{\pm 0.99}$]	**13.79**$_{\pm 0.94}$
Ecoli	True-k	49.15$_{\pm 5.73}$	35.37$_{\pm 4.00}$	45.58$_{\pm 5.24}$	**57.39**$_{\pm 8.79}$	34.87$_{\pm 3.89}$	35.94$_{\pm 4.16}$	37.60$_{\pm 4.16}$	39.94$_{\pm 7.64}$	**51.44**$_{\pm 2.94}$	[**75.75**$_{\pm 5.35}$]
	Best-k	74.43$_{\pm 2.52}$	45.08$_{\pm 2.01}$	74.55$_{\pm 1.89}$	**76.78**$_{\pm 1.65}$	46.88$_{\pm 8.17}$	69.81$_{\pm 3.38}$	52.93$_{\pm 8.89}$	46.53$_{\pm 7.87}$	**75.44**$_{\pm 1.64}$	[**77.43**$_{\pm 1.10}$]
Gisette	True-k	28.10$_{\pm 17.20}$	51.31$_{\pm 15.00}$	38.99$_{\pm 13.63}$	43.65$_{\pm 11.32}$	19.54$_{\pm 14.41}$	[**58.56**$_{\pm 2.09}$]	34.63$_{\pm 11.49}$	10.98$_{\pm 8.86}$	**57.61**$_{\pm 2.38}$	**47.75**$_{\pm 9.47}$
	Best-k	35.42$_{\pm 10.97}$	52.95$_{\pm 10.03}$	42.01$_{\pm 9.78}$	45.15$_{\pm 8.56}$	25.58$_{\pm 9.78}$	[**58.56**$_{\pm 2.09}$]	36.14$_{\pm 8.57}$	18.73$_{\pm 9.25}$	**57.61**$_{\pm 2.38}$	**48.79**$_{\pm 7.30}$
LR	True-k	14.97$_{\pm 0.76}$	[**17.71**$_{\pm 1.27}$]	15.22$_{\pm 1.04}$	14.69$_{\pm 0.90}$	14.02$_{\pm 1.04}$	**16.16**$_{\pm 1.36}$	14.29$_{\pm 0.66}$	11.74$_{\pm 1.79}$	**15.44**$_{\pm 0.73}$	15.28$_{\pm 0.78}$
	Best-k	16.73$_{\pm 0.62}$	[**18.44**$_{\pm 0.88}$]	**17.85**$_{\pm 0.70}$	17.07$_{\pm 0.55}$	16.49$_{\pm 0.80}$	17.55$_{\pm 0.73}$	16.88$_{\pm 0.72}$	16.12$_{\pm 1.29}$	16.77$_{\pm 0.39}$	**17.68**$_{\pm 0.81}$
LS	True-k	56.07$_{\pm 6.52}$	46.27$_{\pm 4.90}$	57.09$_{\pm 5.95}$	**60.07**$_{\pm 5.68}$	36.28$_{\pm 5.60}$	52.68$_{\pm 2.88}$	40.24$_{\pm 5.89}$	26.57$_{\pm 10.16}$	**61.49**$_{\pm 5.25}$	[**61.62**$_{\pm 5.11}$]
	Best-k	60.72$_{\pm 4.17}$	52.23$_{\pm 5.37}$	62.40$_{\pm 3.61}$	**63.07**$_{\pm 3.42}$	41.29$_{\pm 3.76}$	60.76$_{\pm 3.28}$	45.87$_{\pm 4.79}$	37.10$_{\pm 5.62}$	[**65.43**$_{\pm 2.60}$]	**64.77**$_{\pm 3.02}$
MNIST	True-k	49.53$_{\pm 2.73}$	46.54$_{\pm 5.24}$	**51.62**$_{\pm 2.45}$	51.17$_{\pm 2.06}$	36.23$_{\pm 4.11}$	**52.88**$_{\pm 4.27}$	35.27$_{\pm 3.73}$	26.99$_{\pm 6.46}$	[**53.17**$_{\pm 3.13}$]	49.61$_{\pm 1.58}$
	Best-k	51.55$_{\pm 2.53}$	47.64$_{\pm 4.30}$	**54.01**$_{\pm 2.12}$	52.81$_{\pm 2.30}$	42.08$_{\pm 2.61}$	**55.43**$_{\pm 3.01}$	41.56$_{\pm 1.78}$	41.37$_{\pm 2.89}$	[**55.59**$_{\pm 2.66}$]	53.08$_{\pm 2.30}$
PD	True-k	62.21$_{\pm 3.75}$	58.29$_{\pm 5.71}$	63.23$_{\pm 4.22}$	62.85$_{\pm 5.21}$	43.79$_{\pm 6.21}$	**63.38**$_{\pm 5.39}$	45.09$_{\pm 5.11}$	29.90$_{\pm 9.99}$	[**65.42**$_{\pm 4.23}$]	**63.54**$_{\pm 5.22}$
	Best-k	71.13$_{\pm 2.06}$	60.72$_{\pm 4.20}$	**73.80**$_{\pm 0.87}$	**73.68**$_{\pm 1.09}$	56.88$_{\pm 3.24}$	72.40$_{\pm 2.24}$	56.53$_{\pm 3.05}$	54.77$_{\pm 3.78}$	73.10$_{\pm 0.99}$	[**74.61**$_{\pm 0.98}$]
Wine	True-k	89.56$_{\pm 2.40}$	84.50$_{\pm 3.35}$	**90.78**$_{\pm 2.82}$	90.26$_{\pm 3.04}$	88.18$_{\pm 4.09}$	90.02$_{\pm 2.37}$	84.76$_{\pm 10.11}$	88.47$_{\pm 5.75}$	**90.62**$_{\pm 2.25}$	[**91.29**$_{\pm 2.96}$]
	Best-k	89.76$_{\pm 2.19}$	84.50$_{\pm 3.35}$	**90.96**$_{\pm 2.60}$	90.59$_{\pm 2.72}$	88.28$_{\pm 3.56}$	90.02$_{\pm 2.37}$	86.52$_{\pm 5.99}$	88.83$_{\pm 4.31}$	**90.66**$_{\pm 2.21}$	[**91.70**$_{\pm 2.53}$]
Yeast	True-k	16.38$_{\pm 1.58}$	12.32$_{\pm 1.03}$	16.32$_{\pm 1.31}$	**19.01**$_{\pm 1.74}$	11.89$_{\pm 2.59}$	13.36$_{\pm 1.40}$	9.89$_{\pm 0.75}$	11.90$_{\pm 2.40}$	**16.94**$_{\pm 1.33}$	[**20.62**$_{\pm 1.51}$]
	Best-k	20.46$_{\pm 2.53}$	13.77$_{\pm 1.12}$	20.21$_{\pm 2.77}$	**21.40**$_{\pm 1.57}$	13.44$_{\pm 1.25}$	18.82$_{\pm 1.70}$	14.53$_{\pm 0.75}$	14.17$_{\pm 1.67}$	[**21.57**$_{\pm 1.66}$]	**21.48**$_{\pm 1.19}$
Avg. score	True-k	46.13	45.10	47.42	49.73	37.99	48.06	40.05	30.45	51.24	52.60
	Best-k	51.81	47.56	53.21	54.15	42.81	54.30	45.03	39.21	55.90	55.19
Avg. rank	True-k	5.80	6.30	4.70	4.10	8.70	4.40	7.90	8.70	2.20	2.20
	Best-k	6.40	6.40	4.30	3.70	8.50	4.20	7.40	9.10	2.70	2.20



(a) (b)

Fig. 4. Number of times that each method is ranked in the (a) first position and (b) top three with respect to Table III.

when compared to the classical MCLA method, as shown by their average scores and average ranks across ten datasets. To summarize, as shown in Tables II and III and Figs. 3 and 4, the proposed ECPCS-HC and ECPCS-MC methods exhibit overall better performances (with respect to NMI and ARI) than the baseline methods on the benchmark datasets.

D. Robustness to Ensemble Size M

In this section, we evaluate the performances of the proposed methods and the baseline methods using different ensemble sizes M. As shown in Fig. 5, ECPCS-HC obtains the best performance with respect to NMI on the *BC*, *Ecoli*, *LR*, *Wine*, and *Yeast* datasets, whereas ECPCS-MC obtains the best performance on the *CTG* and *PD* datasets, as the ensemble size goes from 10 to 50. Similarly, as shown in Fig. 6, ECPCS-HC obtains the best performance (with respect to ARI) on the *BC*, *Ecoli*, *PD*, and *Wine* datasets, whereas ECPCS-MC obtains the best performance (with respect to ARI) on the *CTG* and *MNIST* datasets, with varying ensemble sizes M. Although the MCLA method shows better ARI scores than the proposed methods on the *LR* dataset, yet on all of the other nine datasets our methods consistently outperform MCLA with different ensemble sizes. As can be seen in Figs. 5 and 6, the proposed ECPCS-HC and ECPCS-MC methods exhibit overall the best performance with respect to NMI and ARI on the benchmark datasets.

E. Sensitivity of Parameter t

In this section, we evaluate the performances of the proposed ECPCS-HC and ECPCS-MC methods with varying parameter t.

Table IV reports the average NMI scores (over 20 runs) of ECPCS-HC and ECPCS-MC when the parameter t takes different values. Note that the parameter t controls the number of steps of the random walkers during the propagation of cluster-wise similarities (as described in Section IV). As shown in Table IV, the proposed methods yield consistently good performances (with respect to NMI) with varying parameter t. Generally, using a larger parameter t (e.g., larger than 10) can lead to better clustering results than using a very small one, which is probably due to the fact that a random walker with adequate number of steps can better capture the multiscale structure information of the graph. Also, the performances (with respect to ARI) by the proposed ECPCS-HC and ECPCS-MC methods are reported in Table V. From the experimental results in Tables IV and V, it can be

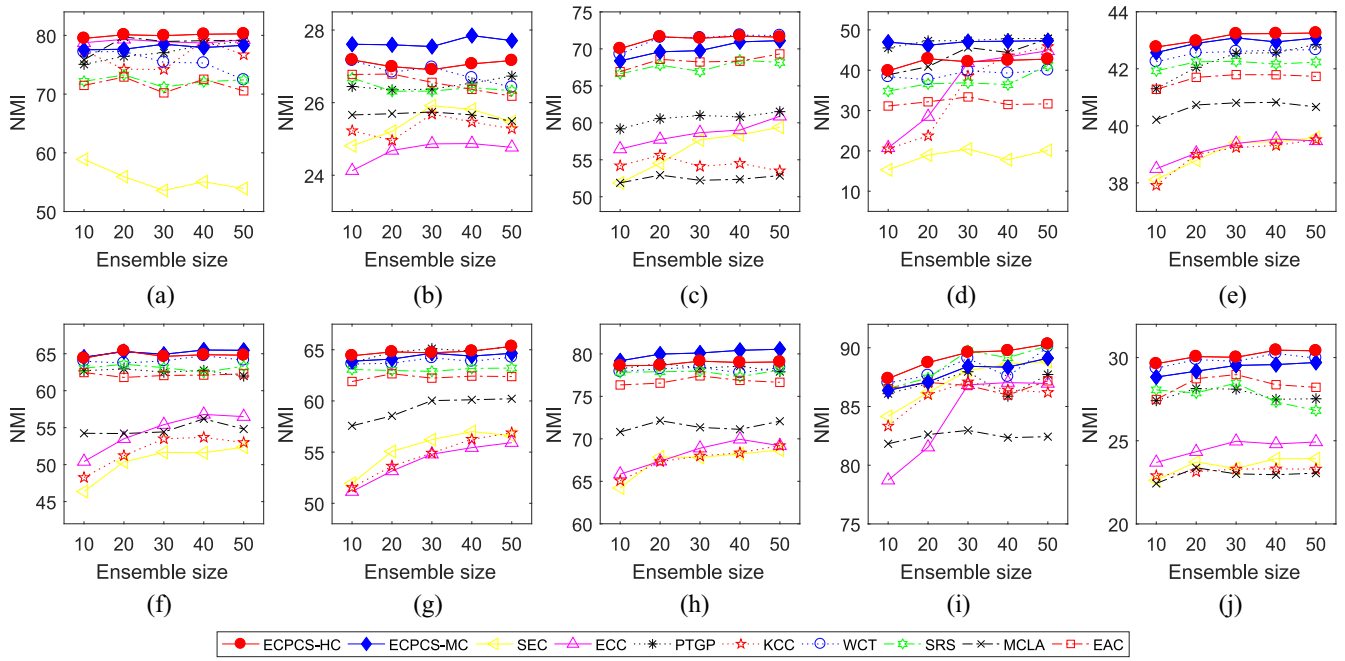


Fig. 5. Average performances with respect to NMI (%) over 20 runs by different ensemble clustering methods with varying ensemble sizes M. (a) *BC*. (b) *CTG*. (c) *Ecoli*. (d) *Gisette*. (e) *LR*. (f) *LS*. (g) *MNIST*. (h) *PD*. (i) *Wine*. (j) *Yeast*.

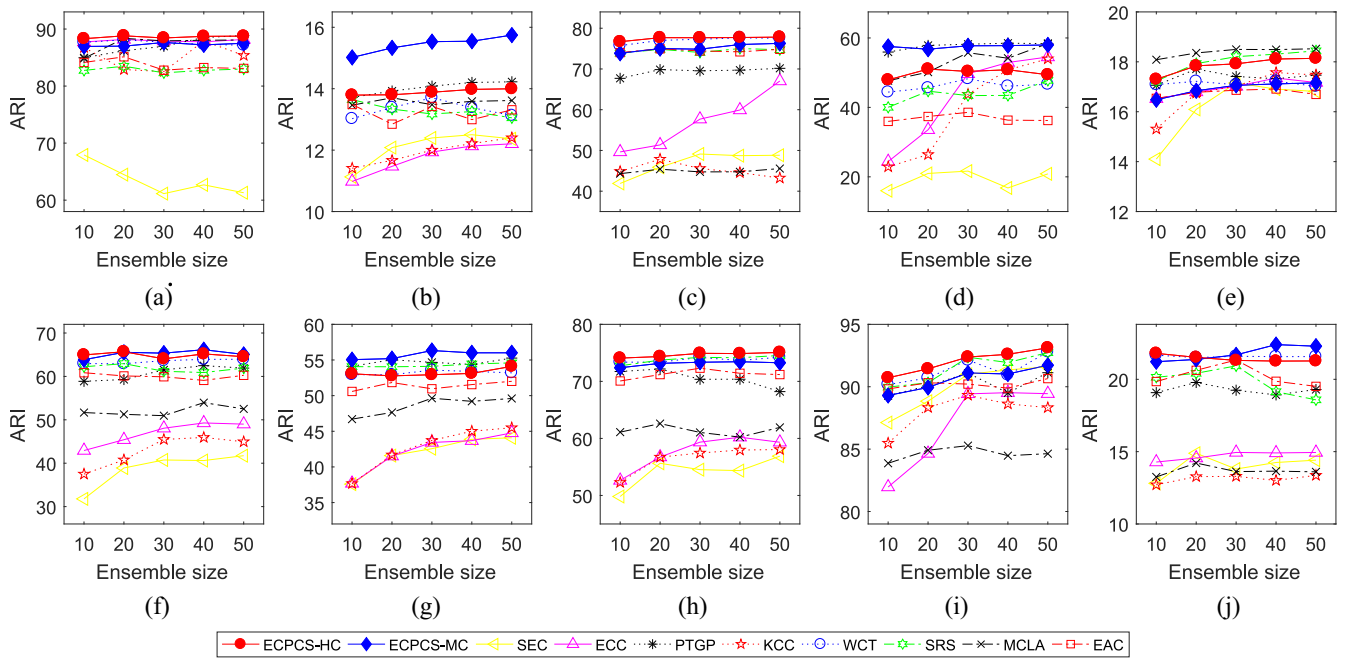


Fig. 6. Average performances with respect to ARI (%) over 20 runs by different ensemble clustering methods with varying ensemble sizes M. (a) *BC*. (b) *CTG*. (c) *Ecoli*. (d) *Gisette*. (e) *LR*. (f) *LS*. (g) *MNIST*. (h) *PD*. (i) *Wine*. (j) *Yeast*.

observed that the proposed methods exhibit robust consensus clustering performances with different values of the parameter.

F. Execution Time

In this section, we evaluate the execution times of different ensemble clustering methods. The experiments are conducted on the *LR* dataset with the data size varying from 0 to 20 000. As shown in Fig. 7, ECPCS-MC is the fastest method, which requires 1.28 s to process the entire *LR* dataset with 20 000 objects, while SEC and MCLA are the second and third fastest ones, which requires 1.56 and 2.00 s, respectively, to process the entire *LR* dataset. The time efficiency of ECPCS-HC is comparable to that of the ECC method, and is better than the PTGP, WCT, and SRS methods. To summarize, the proposed ECPCS-MC and ECPCS-HC methods consistently outperform the baseline methods in terms of clustering quality (as shown in Tables II and III and Figs. 3–6) while exhibiting competitive time efficiency (as shown in Fig. 7).

TABLE IV
AVERAGE PERFORMANCE WITH RESPECT TO NMI (%) OVER 20 RUNS BY OUR ECPCS-HC AND ECPCS-MC METHODS USING VARYING PARAMETER t

Dataset	ECPCS-HC							ECPCS-MC						
	$t=1$	2	4	8	16	32	64	$t=1$	2	4	8	16	32	64
BC	75.30	76.28	76.86	78.23	79.16	79.69	79.99	77.19	77.48	77.80	77.82	77.80	77.81	77.74
CTG	26.65	26.69	26.82	26.90	26.99	26.97	27.10	27.45	27.63	27.63	27.66	27.66	27.69	27.73
Ecoli	70.61	70.98	71.44	71.66	71.49	71.34	71.26	69.63	69.75	70.11	70.13	70.09	70.04	70.08
Gisette	37.58	38.60	40.34	41.01	40.95	41.00	39.49	46.51	46.77	46.99	47.04	46.97	46.88	46.78
LR	42.42	42.35	42.45	42.50	42.75	43.20	43.56	42.36	42.75	42.96	42.90	42.85	42.78	42.77
LS	63.43	63.78	64.22	64.50	64.95	64.93	65.00	65.19	65.31	65.32	65.29	65.18	65.16	65.03
MNIST	63.36	63.43	63.94	64.32	64.88	64.82	64.50	63.63	63.75	63.90	64.01	64.30	64.34	64.22
PD	77.98	78.35	78.53	78.80	78.73	78.78	78.49	79.66	79.81	79.80	79.84	79.77	79.70	79.80
Wine	87.22	88.24	88.52	88.65	88.98	88.95	89.06	87.41	87.69	87.91	87.87	87.91	87.91	87.86
Yeast	29.25	29.35	29.48	29.91	30.04	30.15	29.97	28.73	28.89	29.03	29.20	29.22	29.36	29.44

TABLE V
AVERAGE PERFORMANCE WITH RESPECT TO ARI (%) OVER 20 RUNS BY OUR ECPCS-HC AND ECPCS-MC METHODS USING VARYING PARAMETER t

Dataset	ECPCS-HC							ECPCS-MC						
	$t=1$	2	4	8	16	32	64	$t=1$	2	4	8	16	32	64
BC	86.75	87.34	87.56	88.12	88.52	88.67	88.74	87.03	87.12	87.14	87.15	87.14	87.14	87.09
CTG	13.39	13.47	13.65	13.65	13.49	13.49	13.61	15.52	15.56	15.73	15.66	15.62	15.65	15.62
Ecoli	76.40	76.75	77.17	77.41	77.41	77.35	77.33	75.34	75.44	75.67	75.59	75.57	75.53	75.63
Gisette	43.75	45.15	47.22	48.81	48.76	48.82	47.18	57.09	57.36	57.58	57.63	57.57	57.47	57.37
LR	17.05	17.02	17.07	17.24	17.43	17.35	16.93	16.55	16.59	16.63	16.62	16.65	16.75	16.75
LS	62.89	63.11	64.16	64.51	64.82	64.54	63.94	65.89	66.06	66.29	66.04	65.74	65.23	64.89
MNIST	52.37	52.61	53.12	53.08	53.20	52.67	52.16	53.63	54.00	54.46	54.83	55.45	55.45	55.34
PD	73.03	73.69	74.11	74.47	74.59	74.68	74.82	73.03	73.09	73.17	73.12	73.11	73.20	73.20
Wine	90.57	91.36	91.54	91.63	91.81	91.78	91.99	90.69	90.69	90.70	90.67	90.70	90.69	90.66
Yeast	20.99	21.15	21.46	21.72	21.61	21.41	21.04	21.12	21.19	21.29	21.34	21.39	21.70	21.89

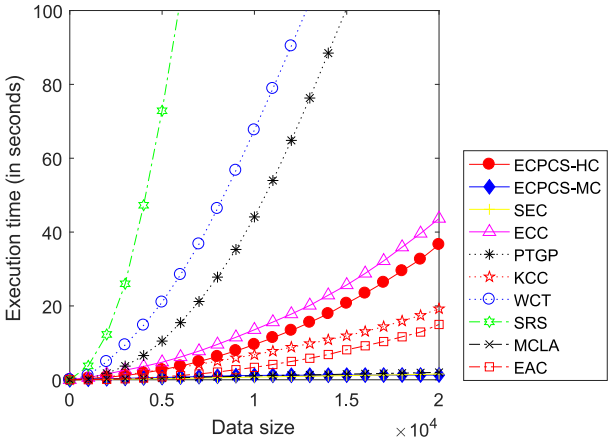


Fig. 7. Execution times of different ensemble clustering methods on the *LR* datasets with the data size varying from 0 to 20 000.

All experiments were conducted in MATLAB 2016b on a PC with an Intel i7-6700K CPU and 64 GB of RAM.

VIII. CONCLUSION

In this paper, we propose a new ensemble clustering approach based on fast propagation of cluster-wise similarities via random walks. By treating the base clusters as nodes and using the Jaccard coefficient to build weighted edges, a cluster similarity graph is first constructed. With a new transition probability matrix defined on the graph, the random walk process is performed with each node treated as a starting node. Then, a new cluster-wise similarity matrix can be derived from the original graph by investigating the propagating trajectories of the random walkers starting from different nodes (i.e., clusters). Then, we construct an ECA matrix by mapping the new cluster-wise similarity from the cluster-level to the object-level, and propose two novel consensus functions, i.e., ECPCS-HC and ECPCS-MC, to achieve the final consensus clustering result. Extensive experiments are conducted on ten real-world datasets, which have shown the advantage of our ensemble clustering approach over the state-of-the-art in terms of both clustering quality and efficiency.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions that help enhance this paper significantly.

Our MATLAB source code is available for download at: www.researchgate.net/publication/328581758.

REFERENCES

[1] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, Feb. 2007.
[2] S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp. 218–237, Jan. 2008.
[3] C.-D. Wang, J.-H. Lai, C. Y. Suen, and J.-Y. Zhu, "Multi-exemplar affinity propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2223–2237, Sep. 2013.
[4] C.-D. Wang, J.-H. Lai, D. Huang, and W.-S. Zheng, "SVStream: A support vector-based algorithm for clustering data streams," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1410–1424, Jun. 2013.
[5] Y. Yang, Z. Ma, Y. Yang, F. Nie, and H. T. Shen, "Multitask spectral clustering by exploring intertask correlation," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1083–1094, May 2015.

[6] C.-D. Wang, J.-H. Lai, and P. S. Yu, "Multi-view clustering based on belief propagation," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 4, pp. 1007–1021, Apr. 2016.

[7] Y. Chen *et al.*, "DHeat: A density heat-based algorithm for clustering with effective radius," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 4, pp. 649–660, Apr. 2018.

[8] Y. Zhang, F.-L. Chung, and S. Wang, "Fast reduced set-based exemplar finding and cluster assignment," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[9] H. He and Y. Tan, "Pattern clustering of hysteresis time series with multivalued mapping using tensor decomposition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 6, pp. 993–1004, Jun. 2018.

[10] J.-S. Wu, W.-S. Zheng, J.-H. Lai, and C. Y. Suen, "Euler clustering on large-scale dataset," *IEEE Trans. Big Data*, to be published.

[11] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[12] D. Huang, J.-H. Lai, C.-D. Wang, and P. C. Yuen, "Ensembling over-segmentations: From weak evidence to strong segmentation," *Neurocomputing*, vol. 207, pp. 416–427, Sep. 2016.

[13] Z. Wang *et al.*, "Discovering and profiling overlapping communities in location-based social networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 4, pp. 499–509, Apr. 2014.

[14] C.-D. Wang, J.-H. Lai, and P. S. Yu, "NEIWalk: Community discovery in dynamic content-based networks," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1734–1748, Jul. 2014.

[15] D. Rafailidis and P. Daras, "The TFC model: Tensor factorization and tag clustering for item recommendation in social tagging systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 3, pp. 673–688, May 2013.

[16] P. Symeonidis, "ClustHOSVD: Item recommendation by combining semantically enhanced tag clustering with tensor HOSVD," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 9, pp. 1240–1251, Sep. 2016.

[17] Q. Zhao, C. Wang, P. Wang, M. Zhou, and C. Jiang, "A novel method on information recommendation via hybrid similarity," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 3, pp. 448–459, Mar. 2018.

[18] D. G. Rajpathak and S. Singh, "An ontology-based text mining method to develop D-matrix from unstructured text," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 7, pp. 966–977, Jul. 2014.

[19] A. K. Jain, "Data clustering: 50 years beyond *K*-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[20] A. L. N. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 835–850, Jun. 2005.

[21] D. Huang, J.-H. Lai, and C.-D. Wang, "Combining multiple clusterings via crowd agreement estimation and multi-granularity link analysis," *Neurocomputing*, vol. 170, pp. 240–250, Dec. 2015.

[22] Z. Yu *et al.*, "Hybrid clustering solution selection strategy," *Pattern Recognit.*, vol. 47, no. 10, pp. 3362–3375, 2014.

[23] 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.

[24] D. Huang, J. Lai, and C.-D. Wang, "Ensemble clustering using factor graph," *Pattern Recognit.*, vol. 50, pp. 131–142, Feb. 2016.

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

[26] 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.

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

[28] Q. Kang, S. Liu, M. Zhou, and S. Li, "A weight-incorporated similarity-based clustering ensemble method based on swarm intelligence," *Knowl. Based Syst.*, vol. 104, pp. 156–164, Jul. 2016.

[29] D. Huang, C.-D. Wang, and J.-H. Lai, "LWMC: A locally weighted meta-clustering algorithm for ensemble clustering," in *Proc. Int. Conf. Neural Inf. Process. (ICONIP)*, 2017, pp. 167–176.

[30] H. Liu, J. Wu, T. Liu, D. Tao, and Y. Fu, "Spectral ensemble clustering via weighted K-means: Theoretical and practical evidence," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 5, pp. 1129–1143, May 2017.

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

[32] Z. Yu *et al.*, "Distribution-based cluster structure selection," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3554–3567, Nov. 2017.

[33] N. Iam-On, T. Boongoen, S. Garrett, and C. 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.

[34] N. Iam-On, T. Boongoen, and S. Garrett, "Refining pairwise similarity matrix for cluster ensemble problem with cluster relations," in *Proc. Int. Conf. Disc. Sci. (ICDS)*, 2008, pp. 222–233.

[35] A. Strehl and J. Ghosh, "Cluster ensembles: A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Mar. 2003.

[36] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2004, p. 36.

[37] A. Topchy, A. K. Jain, and W. Punch, "Clustering ensembles: Models of consensus and weak partitions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1866–1881, Dec. 2005.

[38] Y. Li, J. Yu, P. Hao, and Z. Li, "Clustering ensembles based on normalized edges," in *Proc. Pac.–Asia Conf. Knowl. Disc. Data Min. (PAKDD)*, 2007, pp. 664–671.

[39] T. Li and C. Ding, "Weighted consensus clustering," in *Proc. SIAM Int. Conf. Data Min. (SDM)*, 2008, pp. 798–809.

[40] S. Mimaroglu and E. Erdil, "Combining multiple clusterings using similarity graph," *Pattern Recognit.*, vol. 44, no. 3, pp. 694–703, 2011.

[41] J. Yi, T. Yang, R. Jin, A. K. Jain, and M. Mahdavi, "Robust ensemble clustering by matrix completion," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, 2012, pp. 1176–1181.

[42] L. Franek and X. Jiang, "Ensemble clustering by means of clustering embedding in vector spaces," *Pattern Recognit.*, vol. 47, no. 2, pp. 833–842, 2014.

[43] C. Zhong, X. Yue, Z. Zhang, and J. Lei, "A clustering ensemble: Two-level-refined co-association matrix with path-based transformation," *Pattern Recognit.*, vol. 48, no. 8, pp. 2699–2709, 2015.

[44] H. Liu *et al.*, "Entropy-based consensus clustering for patient stratification," *Bioinformatics*, vol. 33, no. 17, pp. 2691–2698, 2017.

[45] E. Weiszfeld and F. Plastria, "On the point for which the sum of the distances to *n* given points is minimum," *Ann. Oper. Res.*, vol. 167, no. 1, pp. 7–41, 2009.

[46] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[47] Y. Ren, C. Domeniconi, G. Zhang, and G. Yu, "Weighted-object ensemble clustering," in *Proc. Int. Conf. Data Min. (ICDM)*, 2013, pp. 627–636.

[48] Z. Yu, L. Li, J. Liu, J. Zhang, and G. Han, "Adaptive noise immune cluster ensemble using affinity propagation," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 12, pp. 3176–3189, Dec. 2015.

[49] Z. Yu *et al.*, "Adaptive fuzzy consensus clustering framework for clustering analysis of cancer data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 12, no. 4, pp. 887–901, Jul./Aug. 2015.

[50] Y. Ren, C. Domeniconi, G. Zhang, and G. Yu, "Weighted-object ensemble clustering: Methods and analysis," *Knowl. Inf. Syst.*, vol. 51, no. 2, pp. 661–689, 2017.

[51] M. Levandowsky and D. Winter, "Distance between sets," *Nature*, vol. 234, pp. 34–35, Nov. 1971.

[52] L. Lovász, "Random walks on graphs: A survey," *Combinatorics Paul Erdös Eighty*, vol. 2, no. 1, pp. 1–46, 1993.

[53] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, Feb. 2004, Art. no. 026113.

[54] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *Proc. Int. Symp. Comput. Inf. Sci. (ISCIS)*, 2005, pp. 284–293.

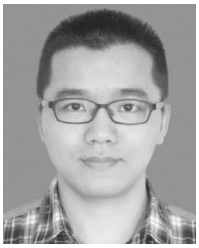
[55] D. Lai, H. Lu, and C. Nardini, "Enhanced modularity-based community detection by random walk network preprocessing," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 81, pp. 264–323, Jun. 2010.

[56] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Boston, MA, USA: Addison-Wesley, 2005.

[57] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[58] K. Bache and M. Lichman. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[59] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, no. 11, pp. 2837–2854, 2010.



Dong Huang (M'15) received the B.S. degree from the South China University of Technology, Guangzhou, China, in 2009, and the M.Sc. and Ph.D. degrees from Sun Yat-sen University, Guangzhou, in 2011 and 2015, respectively, all in computer science.

He joined South China Agricultural University, Guangzhou, in 2015, where he is currently an Associate Professor with the College of Mathematics and Informatics. From 2017 to 2018, he was a Visiting Fellow with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His current research interests include data mining and pattern recognition.

Chang-Dong Wang (M'13) received the Ph.D. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2013.

He was a visiting student with the University of Illinois at Chicago, Chicago, IL, USA, in 2012, for 11 months. He joined Sun Yat-sen University, in 2013, as an Assistant Professor, where he is currently an Associate Professor with the School of Data and Computer Science. He has published over 90 papers in international journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, *Pattern Recognition*, KAIS, ICDM, CIKM, and SDM. His current research interests include machine learning and data mining.

Dr. Wang was a recipient of the Honorable Mention for Best Research Paper Awards for his ICDM 2010, the 2012 Microsoft Research Fellowship Nomination Award, and the 2015 Chinese Association for Artificial Intelligence Outstanding Dissertation.

Hongxing Peng received the Ph.D. degree from South China Agricultural University, Guangzhou, China, in 2014, where he is currently an Associate Professor with the College of Mathematics and Informatics.

From 2016 to 2017, he was a Post-Doctoral Researcher with Washington State University, Pullman, WA, USA. His current research interests include pattern recognition and computer vision.

Jianhuang Lai (SM'13) received the M.Sc. degree in applied mathematics and the Ph.D. degree in mathematics from Sun Yat-sen University, Guangzhou, China, in 1989 and 1999, respectively.

He joined Sun Yat-sen University, in 1989, as an Assistant Professor, where he is currently a Professor with the School of Data and Computer Science. He has published over 200 scientific papers in the international journals and conferences on image processing and pattern recognition, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, *Pattern Recognition*, ICCV, CVPR, IJCAI, ICDM, and SDM. His current research interests include digital image processing, pattern recognition, multimedia communication, and wavelet and its applications.

Prof. Lai serves as a Standing Member of the Image and Graphics Association of China, and also serves as a Standing Director of the Image and Graphics Association of Guangdong.

Chee-Keong Kwoh (SM'18) received the bachelor's degree (First Class) in electrical engineering and the master's degree in industrial system engineering from the National University of Singapore, Singapore, in 1987 and 1991, respectively, and the Ph.D. degree from the Imperial College of Science, Technology and Medicine, University of London, London, U.K., in 1995.

He has been with the School of Computer Science and Engineering, Nanyang Technological University, Singapore, since 1993, where he is the Programme Director of the M.Sc. in Bioinformatics Programme. His current research interests include data mining, soft computing, graph-based inference, bioinformatics, and biomedical engineering. He has done significant research work in the above research areas and has published several quality international conferences and journal papers.

Dr. Kwoh was a recipient of the Public Service Medal by the President of Singapore in 2008. He is an Editorial Board Member of the *International Journal of Data Mining and Bioinformatics*, *Scientific World Journal*, *Network Modeling and Analysis in Health Informatics and Bioinformatics*, *Theoretical Biology Insights*, and *Bioinformation*. He has been a Guest Editor for several journals, such as the *Journal of Mechanics in Medicine and Biology*, *International Journal on Biomedical and Pharmaceutical Engineering*, and others. He has often been invited as an Organizing Member or a Referee and a Reviewer for a number of premier conferences and journals, including GIW, the IEEE BIBM, RECOMB, PRIB, BIBM, ICDM, and iCBBE. He has provided several services to professional bodies in Singapore. He is a member of the Association for Medical and Bio-Informatics, Imperial College Alumni Association of Singapore.