K-Relations-Based Consensus Clustering With Entropy-Norm Regularizers

Liang Bai and Jiye Liang, *Senior Member, IEEE*

***Abstract*— Consensus clustering is to find a high quality and robust partition that is in agreement with multiple existing base clusterings. However, its computational cost is often very expensive and the quality of the final clustering is easily affected by uncertain consensus relations between clusters. In order to solve these problems, we develop a new k-type algorithm, called k-relations-based consensus clustering with double entropy-norm regularizers (KRCC-DE). In this algorithm, we build an optimization model to learn a consensus-relation matrix between final and base clusters and employ double entropy-norm regularizers to control the distribution of these consensus relations, which can reduce the impact of the uncertain consensus relations. The proposed algorithm uses an iterative strategy with strict updating formulas to get the optimal solution. Since its computation complexity is linear with the number of objects, base clusters, or final clusters, it can take low computational costs to effectively solve the consensus clustering problem. In experimental analysis, we compared the proposed algorithm with other k-type-based and global-search consensus clustering algorithms on benchmark datasets. The experimental results illustrate that the proposed algorithm can balance the quality of the final clustering and its computational cost well.**

***Index Terms*— Cluster analysis, consensus clustering, entropy-norm regularizer, k-type clustering.**

I. Introduction

CLUSTER analysis is an important field in machine learning [1]. The goal of clustering is to partition a dataset into several groups so that objects are highly similar within the same clusters but are dissimilar from different clusters. Various types of clustering algorithms [2], [3], [4] have been developed to achieve this goal. Since clustering algorithms work without label information, their clustering results are often different. Under an unsupervised scene, it is not easy to select a suitable clustering result for a dataset, although there are many clustering indices proposed to evaluate the quality of clustering results. Because these indices are defined based on different subjective assumptions. Besides, a clustering algorithm is often sensitive to parameter settings. A clustering algorithm with different input parameters often produces distinct clustering results on a dataset. It is difficult for users to determine which parameter setting would be the proper one, since the supervision information is lacking. Therefore, it is an extremely important task for cluster analysis to get a robust and stable clustering result on a dataset.

Many consensus clustering or cluster ensemble algorithms were proposed [5] to solve this problem. They try to compute the most consistency of multiple clustering results (which are called base clusterings) and then obtain a final clustering result with high robustness and stability. Consensus clustering can be used to overcome the limitations of a single clustering [6]. Different types of consensus clustering methods have been proposed, according to different scientific needs, such as *the pairwise-similarity approach* [7], [8], [9], [10], [11], *the graph-based approach* [12], [13], [14], [15], [16], [17], *the relabeling-based approach* [18], [19], [20], [21], [22], and *the feature-based approach* [23], [24], [25], [26], [27], [28].

In order to keep the final clustering having the most consistency of base clustering results, consensus clustering is often a combinatorial optimization problem, which has been shown to be nondeterministic polynomial (NP)-complete [24], even when the number of input clusterings is three. Therefore, we often need expensive computation costs for global search to obtain a final clustering with high consistency. A consensus clustering algorithm based on a global-search strategy is very difficult to deal with large-scale datasets. Graph-based [12], [14] and k-type-based consensus clustering methods [25], [26], [27], [28] are two good solutions for the challenge of high computational cost. In graph-based methods [12], [14], the clustering ensemble task is implemented on a set of all the base clusters. However, their computational costs are sensitive to the number of base clusters. They are often inefficient while the number of base clusters is very large. K-type-based methods were developed based on the classical k-means or its variants. They inherited the efficiency from k-means. Compared to other algorithms, k-type-based algorithms have linear computational complexity for the number of objects and base clusters. The representative methods include the k-modes-based [25], [26] and the k-means-based [27], [28] algorithms. However, existing k-type-based methods only simply implement one of k-means or its variants on the base clusterings which are seen as categorical or binary data. They did not fully consider the specific characteristics of the consensus clustering task. This lead to the following two important issues, which need to be addressed.

Manuscript received 20 January 2023; revised 20 July 2023; accepted 18 August 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2021ZD0113303 and in part by the National Natural Science Foundation of China under Grant 62022052 and Grant 62276159. *(Corresponding author: Jiye Liang.)*

The authors are with the School of Computer and Information Technology, Shanxi University, Taiyuan, Shanxi 030006, China (e-mail: bailiang@sxu.edu.cn; ljy@sxu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2023.3307158.

Digital Object Identifier 10.1109/TNNLS.2023.3307158

1) *It is a Lack of the Interpretability of Clustering Results, Such as the Meaning of Cluster Representation:* Each k-type algorithm needs to define and learn the cluster representation. For example, in k-means [3], "mean" of a cluster on numerical datasets is seen as its representation. In density-peak [4], an object with high local density in a cluster is seen as its representation. In k-modes [26], "mode" of a cluster on categorical datasets is seen as its representation. In improved k-modes [29], a cluster on categorical datasets is described by the frequency of each categorical value in the cluster. However, the existing representations are not suitable to explain the consensus between final and base clusters.
2) *The Uncertain Consensus-Relations Between Clusters are Not Considered:* The uncertainty of the consensus relation between clusters is mainly from the failure of the consensus evaluation index and the noisy labels in each base clustering. In most of the consensus clustering algorithms, the number of common objects between clusters is seen as an important index to evaluate their consensus. However, in some cases, the consensus index does not work. For example, there may be several base clusters which have the same numbers of common objects with a cluster. However, the same numbers of common objects do not represent the same consensus. Due to the fact that each base clustering includes partial incorrect labels, the consensus relation evaluated by the number of common objects may be uncertain. The uncertainty seriously affects the quality of the obtained final clustering results on many datasets.

In order to solve the above problems, we propose a new k-type algorithm for consensus clustering, called k-relations-based consensus clustering with double entropy-norm regularizers (KRCC-DE). Its main contributions are summarized as follows.

1) We construct an optimization model to learn a matrix of consensus relations between final and base clusters, which is seen as cluster representation. In this model, double entropy-norm regularizers are used to control the distribution of these consensus relations and reduce their uncertainty.
2) We derive an iterative method with strict updating formulas to solve the proposed optimization problem. The proposed method inherits high efficiency from the k-type algorithms.
3) The experimental analysis on several widely-used benchmark datasets illustrates that the proposed algorithm can well solve the problem of balancing the effectiveness and efficiency of cluster ensemble, compared to other consensus clustering algorithms.

The remainder of this article is organized as follows. Section II provides an overview of existing consensus clustering techniques. Section III introduces a new k-relations-based algorithm for consensus clustering. Section IV evaluates the performance of the proposed algorithm. Finally, Section V concludes the article with a discussion of the results.

II. Related Works

Various types of consensus clustering algorithms have been designed to solve the most consistency of base clusterings. In a consensus clustering algorithm, there are two important steps: representation of base clustering and integration technique. In some literature [5], [9], they can be classified into four categories, that is, pairwise similarity–based, graph-based, relabeling-based, and feature-based approaches, according to integration techniques. In this article, we review these existing consensus clustering algorithms, from the view of the representation of base clusterings. Currently, there are three types of representation methods, that is, consensus relations between objects, consensus relations between base clusters, and consensus relations between objects and base clusters, as seen in [30].

1) *Object-consensus-based approach* that represents base clusterings as an object graph or pairwise-similarity matrix to reflect the consistent relations between objects [7], [8], [9], [10], [11], [31]. At the early stage, Fred and Jain [7] constructed a co-association matrix for base clustering and proposed the evidence accumulation–based ensemble algorithm. Strehl and Ghosh [12] defined a hypergraph-based representation for base clusterings, where objects and clusters are seen as nodes and hyperedges, respectively. In [8] and [32], clustering validity functions were used to evaluate the importance of a base clustering and construct a weighted similarity matrix. In [9], a link-based similarity matrix was proposed, where the indirect similarity between clusters is computed. In [10] and [33], a pairwise-similarity matrix was proposed to reflect the label consistency on different subspace clusterings. In [14], a pairwise-similarity matrix for consensus clustering was learned by random walk. In [34], proposed a novel multidiversified ensemble clustering approach for integrating multiple similarity matrices. Lai et al. [11] defined a weighted co-association matrix based on prior information. Zhou et al. [35] built a graph learning model to learn multiple pairwise-similarity matrices for robust consensus clustering. In [36], a deep ensemble clustering method was proposed, which learns a final clustering to reconstruct the weighted pairwise-similarity matrices generated from base clusterings. In [37], they improved the co-association matrix by extracting highly confidence information to enhance the quality of consensus clustering. Besides, Shi et al. [38] proposed a co-association matrix optimization model to improve the co-association matrix by integrating abundant information from both label space and feature space.
2) *Cluster-consensus-based approach* that evaluates the consistent relations between base clusters to define a cluster graph or cluster-similarity matrix for consensus clustering. Relabeling-based ensemble algorithms are the representatives of the cluster consensus–based approach. They proposed different optimization models [18], [19], [21], [39] to solve the label alignment for base clusters. In [12], the MCLA algorithm

was proposed to construct a cluster graph where the nodes and edges denote clusters and the similarity between clusters, respectively. Fern and Brodley [13] extended MCLA to propose the HBGF algorithm, which is an integration of object consensus– and cluster consensus–based approaches. Its nodes represent both objects and clusters. In [40], [41], and [42], a consensus clustering problem is seen as the approximate spectral clustering problem and then different accelerated algorithms have been proposed. To reduce the computational costs, they constructed a cluster-similarity matrix, instead of the object-similarity matrix, to learn the graph representation.

3) *Object-cluster-consensus-based approach* that learns the consensus relations between objects and base clusters and then converts a consensus clustering problem into a categorical or binary data clustering problem [24], [25], [26], [27], [30]. Topchy et al. [24] illustrated the equivalence between categorical data clustering and consensus clustering. In [25], the k-modes algorithm [26], as a representative of categorical data clustering, was used to solve the consensus clustering problem. In [30], from the view of categorical data clustering, information theory was employed to evaluate the quality of base clusterings and consensus of cluster ensemble. Wu et al. [27] used the k-means objective function as consensus function that is equal to the category utility function [43] for categorical data clustering, and then developed k-means-based consensus clustering algorithm. In [17], a self-paced consensus clustering algorithm was proposed to learn a structured bipartite graph from the multiple base clustering results, which reflects the relations between objects and base clusters. In [44], graph representation methods were employed to learn the vector representation of each label. In this case, base clusterings can be converted into numerical data, the consensus relations between objects and clusters were learned by k-means. Liu [45] proposed a simple multiple kernel k-means clustering algorithm which can be used to solve the consensus clustering problem by seeing the co-association matrix of each base clustering as a kernel matrix.

Although the existing consensus clustering methods already have good theoretical and practical contributions, it is still challenging for them to balance the effectiveness and efficiency of algorithms. Therefore, in this article, we focus on how to design an effective k-type clustering for consensus clustering.

III. K-Relations-Based Consensus Clustering

We first introduce the related symbols of consensus clustering. Let $X = \{\mathbf{x}_i\}_{i=1}^n$ be a set of n objects. On a given dataset X, we can run clustering algorithms to produce t different clustering results which are called "base clusterings." Base clusterings can be described by sets Π or matrices B. The set representation of base clusterings is defined as follows. Let $\Pi = \{\pi_1, \ldots, \pi_t\}$ be a set of t base clusterings and $\pi_h = \{C_{h_1}, \ldots, C_{h_{k_h}}\}$ be a set of all the clusters included by the hth base clustering, where k_h is the number of clusters and $C_{h_j} \in \pi_h$ is the jth base cluster in π_h, for $1 \le h \le t$. The matrix representation of base clusterings is defined as follows. Let $B = [B_1, \ldots, B_t]$ be a $n \times p$ matrix, where $p = \sum_{h=1}^t k_h$ is the number of all base clusters from Π, and B_i be a $n \times k_h$ membership matrix, where b_{ih_j} is the membership degree of object $\mathbf{x}_i$ to cluster C_{h_j}. $b_{ih_j} = 1$ if object $\mathbf{x}_i$ belongs to C_{h_j}, otherwise, 0. $\pi_* = \{C_1^*, \ldots, C_k^*\}$ denotes the final clustering including k final clusters, where $C_l^* \in \pi_*$ be the lth final cluster for $1 \le l \le k$, k is the number of the final clusters. $U = [u_{il}]$ is a $n \times k$ membership matrix of the final clustering and u_{il} is the membership degree of $\mathbf{x}_i$ to C_l^*. The task of the consensus clustering problem is to generate a final clustering π_* or U of dataset X based on the base clusterings, which is shown in Fig. 1.

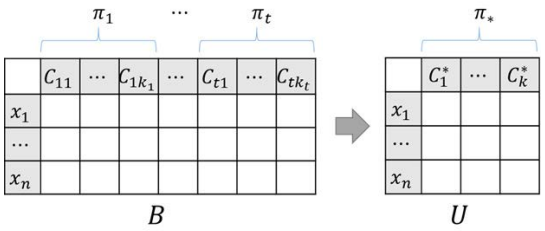


Fig. 1. Consensus clustering task.

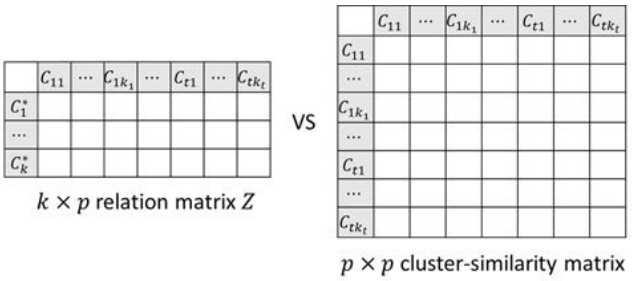


Fig. 2. Consensus-relation matrix versus cluster-similarity matrix.

In this article, we try to learn Z, which is a $k \times p$ consensus-relation matrix, where z_{lh_j} is an element of Z which reflects consensus (similarity) relation between final cluster C_l^* and base cluster C_{h_j}. We assume z_{lh_j} is proportional to the occurrence frequency of the common objects between C_l^* and C_{h_j} in C_l^*, that is,

$$z_{lh_j} \propto f_{lh_j}, \quad \text{where } f_{lh_j} = \frac{\left|C_l^* \cap C_{h_j}\right|}{\left|C_l^*\right|}. \tag{1}$$

Next, let us explain why learning Z can improve clustering speed. In some graph-based methods, a $p \times p$ cluster-similarity matrix is seen as the operation object, in order to fast obtain the final clustering U. However, if p is very large, these graph-based methods are not efficient. Compared to the cluster-similarity matrix, Z needs low computation cost (seen in Fig. 2). Given Z, we can directly compute the membership degree of objects to final clusters, according to a similarity measure.

In this, we define the similarity measure as follows:

$$s_{il} = \mathbf{b}_i \mathbf{z}_l^T \tag{2}$$

where $\mathbf{b}_i$ is the lth row of B and $\mathbf{z}_l$ is the lth row of Z. s_{il} uses the dot product between vectors $\mathbf{b}_i$ and $\mathbf{z}_l$ to reflect the membership degree of object $\mathbf{x}_i$ to final cluster C_l^*. We can see that

$$\mathbf{b}_i \mathbf{z}_l^T = \sum_{\mathbf{x}_i \in C_{h_j}, 1 \le h \le t, 1 \le h_j \le k_h} z_{lh_j} \tag{3}$$

is the sum of the similarity between the lth final cluster and the base clusters that object $\mathbf{x}_i$ belongs to. The higher the similarity value is, the more possibly $\mathbf{x}_i$ is assigned to C_l^*. Based on the similarity measure, a consensus function Φ is defined as

$$\Phi(U, Z) = \sum_{i=1}^{n} \sum_{l=1}^{k} u_{il} s_{il}. \tag{4}$$

It is used to evaluate the consensus degree of final clustering with base clusterings. We have

$$\Phi(U, Z) = \sum_{l=1}^{k} \left| C_l^* \right| \sum_{h=1}^{t} \sum_{j=1}^{k_h} f_{lh_j} z_{lh_j}. \tag{5}$$

Since $z_{lh_j} \propto f_{lh_j}$, $f_{lh_j} z_{lh_j}$ can be used to reflect the consensus between C_l^* and C_{h_j}. The larger $f_{lh_j} z_{lh_j}$ is, the more consensus they have. In this case, $\Phi(U, Z)$ can be seen as the sum of the consensus between all the final and base clusters. For example, if we set $z_{lh_j} = f_{lh_j}$

$$\Phi(U, Z) = \sum_{l=1}^{k} \left| C_l^* \right| \sum_{h=1}^{t} \left(\sum_{j=1}^{k_h} f_{lh_j}^2 \right). \tag{6}$$

We have $0 \le \sum_{j=1}^{k_h} f_{lh_j}^2 \le 1$. The more consistent the base-cluster labels of π_h in the final cluster C_l^* are, the closer $\sum_{j=1}^{k_h} f_{lh_j}^2$ to be 1. Thus, maximizing Φ is used to find out a final clustering with the high consensus of base clusterings.

If we add a constraint $\sum_{j=1}^{k_h} z_{lh_j} = 1$ to $\Phi(U, Z)$, we can obtain the following equation:

$$z_{lh_j} = \begin{cases} 1, & j = \arg\max_{j=1}^{k_h} f_{lh_j} \\ 0, & \text{otherwises} \end{cases} \tag{7}$$

to maximize Φ given U. In this case, we can see that each final cluster has consensus relation with only a cluster from each base clustering. Other consensus relations are omitted. In order to solve this problem, we add a regularizer term Ω to stimulate more consensus relations. A consensus function with regularizer is defined as follows:

$$\max_{U,Z} \; F(U, Z) = \Phi(U, Z) + \Omega(U, Z) \tag{8}$$

where

$$\Omega(U, Z) = -\alpha \sum_{l=1}^{k} \sum_{i=1}^{n} u_{il} \mathbf{z}_l \ln \mathbf{z}_l^T. \tag{9}$$

$\Omega(U, Z)$ is an entropy-norm regularizer, which makes use of information entropy to control the distribution of the vector $\mathbf{z}_{lh} = [z_{lh_1}, \ldots, z_{lh_{k_h}}]$, for $1 \le l \le k$ and $1 \le h \le t$. By maximizing $\Omega(U, Z)$, we can assign such base clusters that have high overlapping degrees with the final cluster to the high within-cluster-consensus values and reduce the roles of the base clusters that have low overlapping degrees with the final cluster. α is a parameter used to control the sparsity of Z. The smaller the parameter value, the more sparse Z is.

Given Φ and Ω, the optimization problem of F becomes

$$\max_{U,Z} \; F = \sum_{i=1}^{n} \sum_{l=1}^{k} u_{il} \left[\mathbf{b}_i \mathbf{z}_l^T - \alpha \mathbf{z}_l \ln \mathbf{z}_l^T \right] \tag{10}$$

$$\text{s.t.} \sum_{j=1}^{k_h} z_{lh_j} = 1, \quad z_{lh_j} \in [0, 1]. \tag{11}$$

Given U, we use the Lagrangian multiplier technique to compute

$$\frac{\partial F'}{\partial z_{lh_j}} = \left| C_l^* \cap C_{h_j} \right| - \alpha \left| C_l^* \right| (1 + \ln z_{lh_j}) + \lambda \tag{12}$$

where

$$F' = F + \lambda \sum_{l=1}^{k} \sum_{h=1}^{t} \left(\sum_{j=1}^{k_h} z_{lh_j} - 1 \right).$$

According to (12), we can solve the maximization problem of F by the following equation:

$$z_{lh_j} = \frac{\exp\left(\frac{f_{lh_j}}{\alpha}\right)}{\sum_{r=1}^{k_h} \exp\left(\frac{f_{lh_r}}{\alpha}\right)}. \tag{13}$$

In this, $\mathbf{z}_{lh}$ reflects the distribution of the cluster labels of base clustering π_h in final cluster C_l^*. Based on the equation, we can see that z_{lh_j} is directly proportional to f_{lh_j} and inversely proportional to other f_{lh_r} for $1 \le r \ne j \le k_h$. In this case, Z reflects the consensus of base-cluster labels within final clusters. However, it ignores the consensus of a base-cluster label among final clusters. A base-cluster label may have high occurrence frequency in more than one final cluster. The consensus between final cluster C_l^* and base cluster C_{h_j} is strong when the label of C_{h_j} has low frequencies in other final clusters. Thus, z_{lh_j} should be inversely proportional to f_{qh_j} for $1 \le q \ne l \le k$. Therefore, we need to consider its distribution in all the final clusters.

We take an example in Fig. 3 for this problem. There is a base clustering $\pi_h = \{C_{h_1}, C_{h_2}, C_{h_3}, C_{h_4}\}$ and a final clustering $\pi_* = \{C_1^*, C_2^*, C_3^*, C_4^*\}$. The figure first shows the frequencies of these base clusters within the final cluster C_1^*. We can see that both f_{1h_1} and f_{1h_2} are equal to 40%. This indicates that if we only consider the frequencies within final clusters to evaluate the consensus relations, the consensus degree between C_1^* and C_{1h_1} is the same as that between C_1^* and C_{1h_2}. However, if we consider the frequencies of these base clusters within other final clusters, we may get different conclusion. According to the figure, we can observe that base cluster C_{1h_1} only has high frequency within final cluster C_1^*. However, base cluster C_{1h_2} has higher frequency within final cluster C_2^* than final cluster C_1^*. Therefore, we can conclude that the consensus degree between C_1^* and C_{1h_1} should be higher than that between C_1^* and C_{1h_2}.

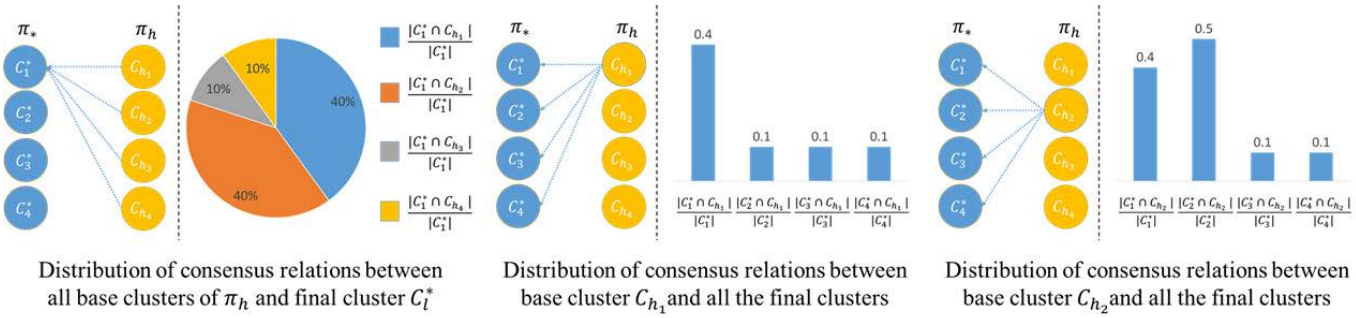


Fig. 3. Distributions of within-cluster and between-cluster-consensus relations.

Based on the above motivation, we split Z into two $k \times p$ variable matrices V and W, that is,

$$Z = V \odot W \tag{14}$$

where $\odot$ is an element-wise multiplication and $z_{lh_j} = v_{lh_j} w_{lh_j}$, for $1 \le l \le k$, $1 \le h \le t$, and $1 \le j \le k_h$. We employ v_{lh_j} and w_{lh_j} to reflect *within-cluster consensus* and *between-cluster consensus* of z_{lh_j}, respectively. The definitions of V and W are formalized as follows.

1) V is a $k \times p$ within-cluster matrix of consensus relations, where $\mathbf{v}_l$ is the lth row of V, v_{lh_j} is the h_jth component of $\mathbf{v}_l$ reflecting the within-cluster consensus of base cluster C_{h_j} to final cluster C_l^*. We assume

$$v_{lh_j} \propto f_{lh_j} \quad \text{and} \quad v_{lh_j} \propto -f_{lh_r}, \quad 1 \le r \ne j \le k_h.$$

Therefore, we add the constraint $\sum_{j=1}^{k_h} v_{lh_j} = 1$, for $1 \le h \le t$, $1 \le l \le k$ to V.

2) W is a $k \times p$ between-cluster matrix of consensus relations, where $\mathbf{w}_l$ is the lth row of W, w_{lh_j} is the h_jth component of $\mathbf{w}_l$ reflecting the between-cluster consensus of base cluster C_{h_j} to final cluster C_l^*. We assume

$$w_{lh_j} \propto f_{lh_j} \quad \text{and} \quad w_{lh_j} \propto -f_{rh_j}, \quad 1 \le r \ne l \le k.$$

Therefore, we add the constraint $\sum_{l=1}^{k} w_{lh_j} = 1$, for $1 \le h \le t$, $1 \le j \le k_h$, to W.

Based on the new description of Z, Ω can be redefined as follows:

$$\Omega(U, V, W) = -\alpha \sum_{l=1}^{k} \sum_{i=1}^{n} u_{il} \mathbf{v}_l \ln \mathbf{v}_l^T - \beta \sum_{l=1}^{k} \sum_{i=1}^{n} u_{il} \mathbf{w}_l \ln \mathbf{w}_l^T. \tag{15}$$

According to the definition, we can see that Ω uses two entropy-norm regularizers to control the distributions of consensus relations between final and base clusters. The term $-\mathbf{v}_l \ln \mathbf{v}_l^T$ with the constraint $\sum_{j=1}^{k_h} v_{lh_j} = 1$ is used to control the distribution of vector $\mathbf{v}_{lh} = [v_{lh_1}, \ldots, v_{lh_{k_h}}]$, for $1 \le l \le k$ and $1 \le h \le t$. By maximizing it, we can assign such base clusters that have high overlapping degrees with the final cluster to the high within-cluster-consensus values and reduce the roles of the base clusters that have low overlapping degrees with the final cluster. The term $-\mathbf{w}_l \ln \mathbf{w}_l^T$ with the constraint $\sum_{l=1}^{k} w_{lh_j} = 1$ is used to control the distribution of the column vector $\mathbf{w}_{h_j} = [w_{1h_j}, \ldots, w_{kh_j}]^T$, for $1 \le h \le t$ and $1 \le j \le k_h$. By maximizing it, we wish each base cluster has high consensus with few final clusters rather than all the final clusters. α and β are two important parameters that are used to control the distributions of V and W, respectively.

TABLE I
DESCRIPTION OF DATASETS

Data set	n	m	k
Soybean	47	21	4
Zoo	101	16	7
Voting	435	16	2
Breastcancers	699	9	2
ORL	400	1024	40
Isolet	1,560	617	26
OpticalDigits	5,620	64	10
Statlog	6,435	36	6
COIL100	7,200	1024	100
Mushroom	8,124	22	2
PenDigits	10,992	36	10
USPS	11,000	256	10
Letters	20,000	16	26
Shuttle	57,756	9	5
MNIST	70,000	784	10

When using V and W, instead of Z, the optimization problem of F is redescribed as follows:

$$\max_{U,V,W} \; F(U, V, W) = \Phi(U, V, W) + \Omega(U, V, W) \tag{16}$$

$$\text{s.t.} \begin{cases} \sum_{l=1}^{k} u_{il} = 1, & u_{il} \in \{0, 1\} \\ \sum_{j=1}^{k_h} v_{lh_j} = 1, & v_{lh_j} \in [0, 1] \\ \sum_{j=1}^{k_h} w_{lh_j} = 1, & w_{lh_j} \in [0, 1]. \end{cases} \tag{17}$$

Maximization of the objective function F with the constraint (17) is a constrained non-linear optimization problem. In order to rapidly solve the optimization problem, we need to iteratively solve the following three subproblems.

1) *Problem P_1:* Fix $U = \hat{U}$ and $V = \hat{V}$, compute W to maximize $F(\hat{U}, \hat{V}, W)$.
2) *Problem P_2:* Fix $U = \hat{U}$ and $W = \hat{W}$, compute V to maximize $F(\hat{U}, V, \hat{W})$.
3) *Problem P_3:* Fix $V = \hat{V}$ and $W = \hat{W}$, compute U to maximize $F(U, \hat{V}, \hat{W})$.

Next, we provide the following theorems to solve these subproblems.

TABLE II

ADJUSTED RAND INDEX (ARI) VALUES OF DIFFERENT ALGORITHMS ON THE BENCHMARK DATASETS WITH FIXED k

Data	KModes	KMeans	KCC	WKCC	KRCC-E	KRCC-DE
ORL	0.4203±0.04	0.4068±0.03	0.3661±0.03	0.4058±0.03	0.4314±0.03	0.4527±0.03
Isolet	0.5251±0.04	0.5263±0.03	0.4892±0.04	0.5263±0.04	0.5385±0.03	0.5543±0.02
OpticalDigits	0.6247±0.08	0.6168±0.07	0.5751±0.07	0.543±0.09	0.6357±0.06	0.6439±0.04
Statlog	0.4855±0.07	0.4698±0.06	0.4555±0.08	0.4254±0.09	0.4972±0.06	0.5010±0.04
COIL100	0.4585±0.02	0.4592±0.02	0.4214±0.02	0.4493±0.02	0.4690±0.02	0.4677±0.02
PenDigits	0.3990±0.07	0.4355±0.06	0.3978±0.07	0.3831±0.07	0.4258±0.05	0.4966±0.06
USPS	0.2866±0.03	0.2903±0.03	0.2916±0.03	0.2852±0.03	0.2890±0.03	0.3062±0.02
Letters	0.1254±0.01	0.1241±0.01	0.1210±0.01	0.1164±0.01	0.1273±0.01	0.1399±0.01
Shuttle	0.5310±0.09	0.5147±0.12	0.5432±0.16	0.5738±0.20	0.2351±0.09	0.5950±0.04
MNIST	0.4036±0.03	0.3949±0.04	0.4079±0.04	0.3435±0.04	0.4080±0.03	0.4282±0.02

TABLE III

NORMALIZED MUTUAL INFORMATION (NMI) VALUES OF DIFFERENT ALGORITHMS ON THE BENCHMARK DATASETS WITH FIXED k

Data	KModes	KMeans	KCC	WKCC	KRCC-E	KRCC-DE
ORL	0.7692±0.01	0.7650±0.01	0.7378±0.02	0.7642±0.01	0.7744±0.01	0.7845±0.01
Isolet	0.7598±0.01	0.7617±0.02	0.7396±0.02	0.7603±0.02	0.7644±0.01	0.7699±0.01
OpticalDigits	0.7397±0.04	0.7384±0.03	0.7141±0.03	0.7004±0.05	0.7454±0.03	0.7500±0.02
Statlog	0.5788±0.04	0.5715±0.04	0.5531±0.06	0.5395±0.06	0.5902±0.04	0.5935±0.03
COIL100	0.7702±0.01	0.7739±0.01	0.7531±0.01	0.7699±0.01	0.7739±0.01	0.7733±0.01
PenDigits	0.6641±0.04	0.6691±0.03	0.6314±0.04	0.6379±0.04	0.6779±0.03	0.6996±0.02
USPS	0.4677±0.02	0.4672±0.02	0.4633±0.02	0.4620±0.03	0.4695±0.02	0.4809±0.01
Letters	0.3329±0.01	0.3338±0.01	0.3262±0.01	0.3270±0.01	0.3374±0.01	0.3423±0.01
Shuttle	0.6867±0.09	0.6680±0.13	0.6680±0.14	0.6615±0.19	0.6865±0.09	0.6943±0.06
MNIST	0.5224±0.02	0.5219±0.02	0.5293±0.02	0.4946±0.03	0.5255±0.02	0.5414±0.01

TABLE IV

ARI VALUES OF DIFFERENT ALGORITHMS ON THE BENCHMARK DATASETS WITH RANDOM k

Data	KModes	KMeans	KCC	WKCC	KRCC-E	KRCC-DE
ORL	0.4074±0.03	0.3955±0.03	0.3487±0.03	0.3953±0.03	0.4124±0.02	0.4421±0.03
Isolet	0.5203±0.03	0.5207±0.03	0.4823±0.03	0.5076±0.04	0.5279±0.03	0.5410±0.03
OpticalDigits	0.5928±0.06	0.5921±0.07	0.5622±0.06	0.5746±0.08	0.5968±0.06	0.6494±0.05
Statlog	0.4416±0.07	0.4383±0.07	0.3988±0.09	0.4389±0.08	0.4454±0.07	0.4920±0.05
COIL100	0.4588±0.02	0.4594±0.02	0.414±0.02	0.4524±0.02	0.4709±0.01	0.2074±0.11
PenDigits	0.4080±0.07	0.4268±0.05	0.4325±0.06	0.3836±0.06	0.4227±0.07	0.5200±0.04
USPS	0.2891±0.03	0.2907±0.03	0.3007±0.03	0.2916±0.03	0.2937±0.03	0.3005±0.02
Letters	0.1175±0.01	0.1211±0.01	0.1185±0.01	0.1107±0.01	0.1187±0.01	0.1353±0.01
Shuttle	0.6125±0.10	0.6218±0.14	0.6133±0.16	0.6283±0.16	0.6302±0.08	0.6355±0.10
MNIST	0.4008±0.04	0.4043±0.05	0.4117±0.05	0.4140±0.04	0.4030±0.04	0.4298±0.03

TABLE V

NMI VALUES OF DIFFERENT ALGORITHMS ON THE BENCHMARK DATASETS WITH RANDOM k

Data	KModes	KMeans	KCC	WKCC	KRCC-E	KRCC-DE
ORL	0.7606±0.01	0.7568±0.01	0.7262±0.01	0.7553±0.01	0.7623±0.01	0.7769±0.01
Isolet	0.7544±0.01	0.757±0.01	0.7351±0.02	0.7544±0.02	0.7579±0.01	0.7655±0.01
OpticalDigits	0.7231±0.03	0.7252±0.03	0.7086±0.03	0.7192±0.04	0.7253±0.03	0.7606±0.03
Statlog	0.5593±0.05	0.5534±0.05	0.5127±0.07	0.5481±0.05	0.5630±0.05	0.5855±0.04
COIL100	0.7672±0.01	0.7724±0.01	0.7488±0.01	0.7697±0.01	0.7714±0.01	0.6101±0.14
PenDigits	0.6625±0.04	0.6573±0.03	0.6472±0.03	0.639±0.03	0.6684±0.03	0.6955±0.02
USPS	0.4735±0.02	0.4744±0.03	0.4796±0.02	0.4744±0.02	0.4772±0.02	0.4793±0.02
Letters	0.3246±0.01	0.3336±0.01	0.3285±0.01	0.3213±0.01	0.3281±0.01	0.3417±0.01
Shuttle	0.6712±0.07	0.6747±0.11	0.6753±0.14	0.6722±0.13	0.6821±0.06	0.6918±0.07
MNIST	0.5205±0.03	0.5327±0.03	0.5360±0.03	0.5383±0.02	0.5229±0.02	0.5409±0.02

TABLE VI

ARI VALUES OF DIFFERENT ALGORITHMS ON THE CATEGORICAL DATASETS

Data	KModes	KMeans	KCC	WKCC	KRCC-E	KRCC-DE
Soybean	0.6615±0.18	0.7316±0.22	0.3515±0.15	0.4836±0.19	0.7652±0.21	0.8372±0.18
Zoo	0.6804±0.13	0.6651±0.12	0.5831±0.15	0.6176±0.15	0.6830±0.11	0.8096±0.08
Voting	0.5126±0.08	0.5549±0.11	0.5094±0.15	0.3746±0.20	0.5693±0.08	0.5780±0.01
Breastcancer	0.5546±0.31	0.7642±0.16	0.7679±0.22	0.2261±0.06	0.7818±0.01	0.8688±0.01
Mushroom	0.2589±0.24	0.4123±0.26	0.2846±0.26	0.3507±0.26	0.4271±0.25	0.4937±0.22

TABLE VII

NMI VALUES OF DIFFERENT ALGORITHMS ON THE CATEGORICAL DATASETS

Data	KModes	KMeans	KCC	WKCC	KRCC-E	KRCC-DE
Soybean	0.7809±0.13	0.8458±0.13	0.4963±0.14	0.6179±0.16	0.8679±0.12	0.9076±0.10
Zoo	0.7752±0.05	0.7689±0.05	0.7049±0.07	0.7554±0.08	0.7855±0.04	0.7953±0.05
Voting	0.4433±0.07	0.4750±0.1	0.4506±0.12	0.2928±0.14	0.4890±0.07	0.5045±0.01
Breastcancer	0.4932±0.23	0.6939±0.14	0.6932±0.19	0.2292±0.04	0.7097±0.01	0.7832±0.01
Mushroom	0.2559±0.22	0.3969±0.22	0.2611±0.24	0.3476±0.24	0.4136±0.22	0.4423±0.20

TABLE VIII

COMPARISONS WITH GLOBAL-SEARCH ALGORITHMS

Data	ARI				NMI			
	USENC	MCLA	WCT	KRCC+DE	USENC	MCLA	WCT	KRCC+DE
ORL	0.4647	0.5018	0.4429	0.5143	0.7873	0.7986	0.7875	0.8060
Isolet	0.5560	0.5611	0.5190	0.5760	0.7728	0.7773	0.8047	0.7818
OpticalDigits	0.6601	0.7430	0.5935	0.7164	0.7457	0.7861	0.7268	0.7785
Statlog	0.5062	0.4296	0.5229	0.6490	0.5941	0.5195	0.6039	0.6694
COIL100	0.5118	0.5001	0.4231	0.4758	0.7923	0.7903	0.7789	0.7700
PenDigits	0.6528	0.7264	0.6036	0.6213	0.7455	0.7768	0.7316	0.7027
USPS	0.3076	0.3055	0.3255	0.3569	0.4660	0.4524	0.4840	0.5034
Letters	0.1361	0.1363	0.1062	0.1445	0.3508	0.3458	0.3325	0.3619
Shuttle	0.6441	0.6734	0.0000	0.6975	0.6924	0.7046	NA	0.8095
MNIST	0.4023	0.3404	0.0000	0.4453	0.5243	0.4729	NA	0.5409

TABLE IX

RUNNING TIME (SECONDS) OF DIFFERENT ALGORITHMS

Data	KModes	KMeans	KCC	WKCC	USENC	MCLA	WCT	KRCC-E
ORL	0.06	0.11	0.10	0.12	9.13	3.78	39.87	0.23
Isolet	0.09	0.43	0.38	0.27	1.79	2.71	27.59	0.27
OpticalDigits	0.06	0.43	1.25	0.44	0.33	3.01	69.49	0.13
Statlog	0.04	0.28	0.85	0.41	0.15	1.84	75.87	0.08
COIL100	6.17	16.13	18.14	14.58	157.06	50.16	1639.68	12.00
PenDigits	0.06	0.42	1.20	0.32	0.31	2.82	69.58	0.12
USPS	0.17	1.26	5.51	0.96	0.68	5.26	255.75	0.37
Letters	1.38	10.27	21.03	5.60	5.03	18.41	1115.84	4.88
Shuttle	0.07	0.54	2.89	1.62	0.50	6.49	NA	0.13
MNIST	0.74	9.11	45.05	5.68	3.17	22.02	NA	2.02

TABLE X

DIFFERENT INITIALIZATION

Data	ARI			NMI		
	Random	CU	k-means++	Random	CU	k-means++
ORL	0.4452	0.4562	0.4566	0.7807	0.7829	0.7856
Isolet	0.5541	0.5623	0.5798	0.7673	0.7731	0.7736
OpticalDigits	0.6462	0.7022	0.7022	0.7510	0.7684	0.7684
Statlog	0.5063	0.5263	0.5263	0.5924	0.6124	0.6124
COIL100	0.4648	0.4708	0.4765	0.7715	0.7726	0.7763
PenDigits	0.4989	0.5989	0.5993	0.6940	0.6940	0.6968
USPS	0.3034	0.3348	0.3476	0.4866	0.4825	0.4866
Letters	0.1334	0.1363	0.1386	0.3411	0.3588	0.3571
Shuttle	0.5983	0.6383	0.6383	0.6900	0.7321	0.7478
MNIST	0.4236	0.4294	0.4289	0.5401	0.5426	0.5438

Theorem 1: Let $U = \hat{U}$ and $W = \hat{W}$ be fixed. $F(\hat{U}, V, \hat{W})$ is maximized iff

$$v_{lh_j} = \frac{\exp\left(\frac{w_{lh_j} f_{lh_j}}{\alpha}\right)}{\sum_{r=1}^{k_h} \exp\left(\frac{w_{lh_r} f_{lh_r}}{\alpha}\right)} \tag{18}$$

for $1 \le h \le t$, $1 \le l \le k$, $1 \le j \le k_h$.

Proof: Let $\kappa_{lh} = \sum_{j=1}^{k_h} |C_l^* \cap C_{h_j}| v_{lh_j} w_{lh_j} - \alpha |C_l^*| \sum_{j=1}^{k_h} v_{lh_j} \ln v_{lh_j}$, for $1 \le h \le t$ and $1 \le l \le k$. We have the following equation:

$$F(U, V, W) = \sum_{h=1}^{t} \sum_{l=1}^{k} \kappa_{lh} - \beta \sum_{l=1}^{k} \sum_{i=1}^{n} u_{il} \mathbf{w}_l \ln \mathbf{w}_l^T.$$

Each κ_{lh} is independent of each other. Given U and W, $|C_l^*|$, $|C_l^* \cap C_{h_j}|$, and $\mathbf{w}_l \ln \mathbf{w}_l^T$ are constants. As κ_{lh} is a strictly convex function, the well-known Karush–Kuhn–Tucker (K-K-T) necessary optimization condition is also sufficient to ensure an optimal solution. Consequently, $\mathbf{v}_{lh} = [v_{lh_1}, \ldots, v_{lh_{k_h}}]$ is an optimal solution if and only if there exists λ together with $\mathbf{v}_{lh}$ that satisfies the following system of equations:

$$\tilde{\kappa}_{lh}(\mathbf{v}_{lh}, \lambda) = \kappa_{lh} + \lambda \left(\sum_{j=1}^{k_h} v_{lh_j} - 1 \right)$$

$$\nabla_{\mathbf{v}_{lh}} \tilde{\kappa}_{lh}(\mathbf{v}_{lh}, \lambda) = 0, \quad \sum_{j=1}^{k_h} v_{lh_j} = 1. \tag{19}$$

We have

$$\frac{\partial \tilde{\kappa}_{lh}(\mathbf{v}_{lh}, \lambda)}{\partial v_{lh_j}} = \left|C_l^* \cap C_{h_j}\right| w_{lh_j} - \alpha \left|C_l^*\right| (1 + \ln v_{lh_j}) + \lambda. \tag{20}$$

From (19) and (20), we obtain the optimal solution

$$v_{lh_j} = \frac{\exp\left(\frac{w_{lh_j}}{\alpha}\frac{\left|C_l^* \cap C_{h_j}\right|}{\left|C_l^*\right|}\right)}{\sum_{r=1}^{k_h}\exp\left(\frac{w_{lh_r}}{\alpha}\frac{\left|C_l^* \cap C_{h_r}\right|}{\left|C_l^*\right|}\right)}.$$

This completes the proof. ∎

Theorem 2: Let $U = \hat{U}$ and $V = \hat{V}$ be fixed. $F(\hat{U}, \hat{V}, W)$ is maximized iff

$$w_{lh_j} = \frac{\exp\left(\frac{v_{lh_j} f_{lh_j}}{\beta}\right)}{\sum_{r=1}^{k}\exp\left(\frac{v_{rh_j} f_{rh_j}}{\beta}\right)} \tag{21}$$

for $1 \le h \le t$, $1 \le l \le k$.

Proof: Let $\theta_{h_j} = \sum_{l=1}^{k}|C_l^* \cap C_{h_j}|v_{lh_j}w_{lh_j} - \beta|C_l^*|w_{lh_j}\ln w_{lh_j}$, for $1 \le h \le t$ and $1 \le j \le k_h$. We have the following equation:

$$F(U, V, W) = \sum_{h=1}^{t}\sum_{j=1}^{k_h}\theta_{h_j} - \alpha\sum_{l=1}^{k}\sum_{i=1}^{n}u_{il}\mathbf{v}_l\ln\mathbf{v}_l^T.$$

Each θ_{h_j} is independent of each other. Given U and V, $|C_l^*|$, $|C_l^* \cap C_{h_j}|$, v_{lh_j}, and $\mathbf{v}_l\ln\mathbf{v}_l^T$ are constants. Thus, minimizing the objective function F is equivalent to minimizing each θ_{h_l}. Since θ_{h_j} is a strictly convex function, it follows that the K-K-T necessary optimization condition is also sufficient. Therefore, the vector $\mathbf{w}_{h_j} = [w_{1h_j}, \ldots, w_{kh_j}]^T$ is an optimal solution if and only if there exists a scalar λ such that the following system of equations is satisfied:

$$\begin{aligned}\tilde{\theta}_{h_j}(\mathbf{w}_{h_j}, \lambda) &= \theta_{h_j} + \lambda\left(\sum_{l=1}^{k}w_{lh_j} - 1\right)\\ \nabla_{\mathbf{w}_{h_j}}\tilde{\theta}_{h_j}(\mathbf{w}_{h_j}, \lambda) &= 0, \quad \sum_{l=1}^{k}w_{lh_j} = 1.\end{aligned} \tag{22}$$

We have

$$\frac{\partial\tilde{\theta}_{h_j}(\mathbf{w}_{h_j}, \lambda)}{\partial w_{lh_j}} = \left|C_l^* \cap C_{h_j}\right|v_{lh_j} - \beta\left|C_l^*\right|(1 + \ln w_{lh_j}) + \lambda. \tag{23}$$

From (22) and (23), we obtain the optimal solution

$$w_{lh_j} = \frac{\exp\left(\frac{v_{lh_j}}{\beta}\frac{\left|C_l^* \cap C_{h_j}\right|}{\left|C_l^*\right|}\right)}{\sum_{r=1}^{k}\exp\left(\frac{v_{hrj}}{\beta}\frac{\left|C_r^* \cap C_{h_j}\right|}{\left|C_r^*\right|}\right)}.$$

This completes the proof. ∎

Theorem 3: Let $V = \hat{V}$ and $W = \hat{W}$ be fixed. $F(U, \hat{V}, \hat{W})$ is maximized iff

$$u_{il} = \begin{cases}1, & l = \arg\max_l \ \mathbf{b}_i\mathbf{z}_l^T - \alpha\mathbf{v}_l\ln\mathbf{v}_l{}^T - \beta\mathbf{w}_l\ln\mathbf{w}_l{}^T\\ 0, & \text{otherwise}\end{cases} \tag{24}$$

for $1 \le l \le k$, $1 \le i \le n$.

Proof: Let $\psi_{il} = \mathbf{b}_i\mathbf{z}_l^T - \alpha\mathbf{v}_l\ln\mathbf{v}_l{}^T - \beta\mathbf{w}_l\ln\mathbf{w}_l{}^T$ and $\varphi_i = \sum_{l=1}^{k}\psi_{il}$. For a given V and W, $F(U, V, W) = \sum_{i=1}^{n}\varphi_i$. Since each φ_i is independent of each other. Maximizing F is equivalent to maximizing each φ_i. When $u_{il} = 1$, we have $u_{ij} = 0, 1 \le j \le k, j \ne l$ and $\varphi_i = \psi_{il}$. It is clear that φ_i is maximized iff $l = \arg\max_l \psi_{il}$. The result follows. ∎

Based on [26], we can prove the convergence of the proposed algorithm by Theorem 4 as follows.

Theorem 4: The KRCC-DE algorithm converges in a finite number of iterations.

Proof: We first observe that there are only a finite number of possible partitions U. We then demonstrate that each of these partitions U appears at most once in the sequence generated by the algorithm. Suppose that $U^{(\tau_1)} = U^{(\tau_2)}$, where $\tau_1 \ne \tau_2$. Since we know $U^{(\tau)}$, we can compute the minimizer $V^{(\tau)}$ independently of $W^{(\tau)}$. We have the maximizers $V^{(\tau_1)}$ and $V^{(\tau_2)}$ for $U^{(\tau_1)}$ and $U^{(\tau_2)}$, respectively. Using $U^{(\tau_1)}$ and $V^{(\tau_1)}$ and $U^{(\tau_2)}$ and $V^{(\tau_2)}$, according to Theorem 3, we can compute the maximizers $W^{(\tau_1)}$ and $W^{(\tau_2)}$, respectively. Since $W^{(\tau_1)} = W^{(\tau_2)}$, we obtain that

$$F(U^{(\tau_1)}, V^{(\tau_1)}, W^{(\tau_1)}) = F(U^{(\tau_2)}, V^{(\tau_2)}, W^{(\tau_2)}).$$

However, we know that the sequence $F(\cdot, \cdot, \cdot)$ generated by the algorithm is non-decreasing. Thus, the proof is complete. ∎

Based on Theorems 1, 2, and 3, an iterative optimization algorithm is proposed to maximize the objective function F with the constraint, which is described in Algorithm 1. It is called K-relations-based consensus clustering with double entropy-norm regularizers (KRCC-DE). Based on Theorem 4, we conclude that the proposed algorithm can converge in a finite number of iterations.

Algorithm 1 KRCC-DE Algorithm

Input: Π, k, α, β
Output: U
Initialize U and W;
Repeat
Fixed U and W, solve Problem P_1 to compute V by Theorem 1;
Fixed U and V, solve Problem P_2 to compute W by Theorem 2;
Fixed V and W, solve Problem P_3 to compute U by Theorem 3;
Until The objective function F is not changed.

Before implementing this algorithm, we need to provide the initialization of W and U. For initial between-cluster relation matrix W, we initially set each w_{lh_j} to $1/k$, for $1 \le l \le k$, $1 \le h \le t$, and $1 \le j \le k_h$. Compared to W, the proposed algorithm is affected by initial U, which is a common shortcoming for k-type algorithms. In order to overcome this shortcoming, we can employ one of the existing initialization methods of k-type algorithms for U, such as k-means++ [46], or we can use one of internal clustering indices, such as category utility (CU) [47] measure, to select the best base clustering to initialize U.

Besides, we need to input the number of final clusters k, and the parameters α and β. In general, k is set according

to prior knowledge of users. We assume α and β should be not less than 0. If $\alpha = 0$ and $\beta = 0$, the regularizers do not work in the optimization of F. In this case, Z becomes the most sparse, which leads to the omission of many important consensus relations. If $\alpha > 0$ and $\beta = 0$, the regularizer of W do not work and Z is equivalent to V. Similarly, if $\alpha = 0$ and $\beta > 0$, Z is equivalent to W. In this article, we hope to set small positive values for V and W to make them sparse. This setting can reduce many of the uncertain consensus relations.

Similar to other k-type algorithms, the time complexity of the proposed algorithm is $O(nkp\tau)$, where τ is the number of iterations. The time complexity is linear with the number of objects n, base clusters p, or final clusters k. We know that the computational costs of object-consensus clustering and cluster-consensus clustering are $O(n^2)$ and $O(np^2)$ with respect to n and p, respectively. According to the comparison of the computational complexities, we can see that if n and p are very large in a dataset, the k-type clustering is suitable, compared to other types of consensus clustering. The storage complexity of the proposed algorithm is $O(np + nk + 2pk)$, which is necessary to hold the set of t base clusterings, the final partition matrix U, the cluster representation matrices V and W. Thus, the complexity analysis reveals that the proposed algorithm inherits the efficiency of k-type algorithms for large-scale datasets.

IV. Relation Between KRCC-DE and Other K-Type Clustering

If we see B as an input data matrix and Z as cluster prototype matrix, the objective function of k-type clustering is generally described as follows:

$$P(U, Z) = \sum_{i=1}^{n}\sum_{l=1}^{k} u_{il} d(\mathbf{b}_i, \mathbf{z}_l) \tag{25}$$

where d is a dissimilarity or distance measure to evaluate the similarity between an object and a cluster prototype. In k-means, Euclidean distance is used to define d, that is, $\|\mathbf{b}_i - \mathbf{z}_l\|^2$. In this case, the objective function becomes

$$\begin{aligned} P(U, Z) &= \sum_{i=1}^{n}\sum_{l=1}^{k} u_{il}\|\mathbf{b}_i - \mathbf{z}_l\|^2 \\ &= \sum_{i=1}^{n}\sum_{l=1}^{k} u_{il}\left(\mathbf{b}_i^2 + \mathbf{z}_l^2 - 2\mathbf{b}_i\mathbf{z}_l\right) \\ &= kn + \sum_{i=1}^{n}\sum_{l=1}^{k} u_{il}\mathbf{z}_l^2 - 2\Phi(U, Z). \end{aligned} \tag{26}$$

If fix U to minimize $P(U, Z)$, by the Lagrangian multiplier technique to compute

$$\frac{\partial P}{\partial \mathbf{z}_l} = 2\sum_{i=1}^{n}\sum_{l=1}^{k} u_{il}(\mathbf{z}_l - \mathbf{b}_i) = 0 \tag{27}$$

we can obtain $z_{lh_j} = f_{lh_j}$. In this case, we have

$$\sum_{i=1}^{n}\sum_{l=1}^{k} u_{il}\mathbf{z}_l^2 = \Phi(U, Z). \tag{28}$$

Thus, we can see the following relation between the objective function of k-means and Φ defined in this article:

$$P(U, Z) = kn - \Phi(U, Z). \tag{29}$$

According to the equation, we can conclude that if we do not consider the entropy-norm regularizers $\Omega(U, Z)$, the objective function of KRCC-DE is equal to k-means. Besides, we also can see the role of the entropy-norm regularizers, which make Z become sparse, compared to directly computing Z by f_{lh_j}.

V. Experimental Analysis

A. Datasets

In order to verify the performance of the proposed algorithm, we perform experiments on several widely used benchmark datasets, which can be found from https://cs.nyu.edu/ roweis/data.html and https://archive.ics.uci.edu/ml. These datasets include a variety of different types, such as face image (ORL), spoken letter recognition (Isolet), satellite image (Statlog), handwritten digits (COIL100, OpticalDigits, USPS and MNIST), shuttle information data (Shuttle), handwritten letters (Letters), categorical data (Soybean, Zoo, Voting, Breastcancer, Mushroom). Details of the tested datasets can be found in Table I. For each non-categorical datasets, we set the number of base clusterings $t = 100$ and use classical k-means as base clusterer to produce 100 different base clusterings. For a categorical dataset, each of its features is seen as a base clustering. The number of clusters k is set to the number of real clusters on each dataset.

B. Compared Methods

In order to properly examine the performance of the KRCC-DE algorithm, we compare it with the following k-type-based consensus clustering algorithms.

1) *k-modes algorithm* [26] sees multiple base clusterings as categorical data and uses the k-modes algorithm to produce the final clustering.
2) *k-means algorithm* [3] sees multiple base clusterings as numerical data and uses the k-means algorithm to produce the final clustering.
3) *k-means-based algorithm (KCC)* [27] was proposed by Wu and Liu et al., which extends k-means objective function to build the optimization model for consensus clustering.
4) *Weighted k-means-based algorithm (WKCC)* [28] was a weighted version of KCC.
5) *k-relations-based algorithm with single entropy-norm regularizer (KRCC-E)* is equal to KRCC-DE with the parameter $\beta = 0$.

Besides, we also compare the proposed algorithm with the following global-search consensus clustering algorithms.

1) *The USENC algorithm* [14] is an approximated spectral clustering algorithm for consensus clustering.
2) *The MCLA algorithm* [12] is a kind of graph-based consensus clustering algorithm.
3) *The WCT algorithm* [9] is a kind of pairwise similarity–based consensus clustering algorithm.

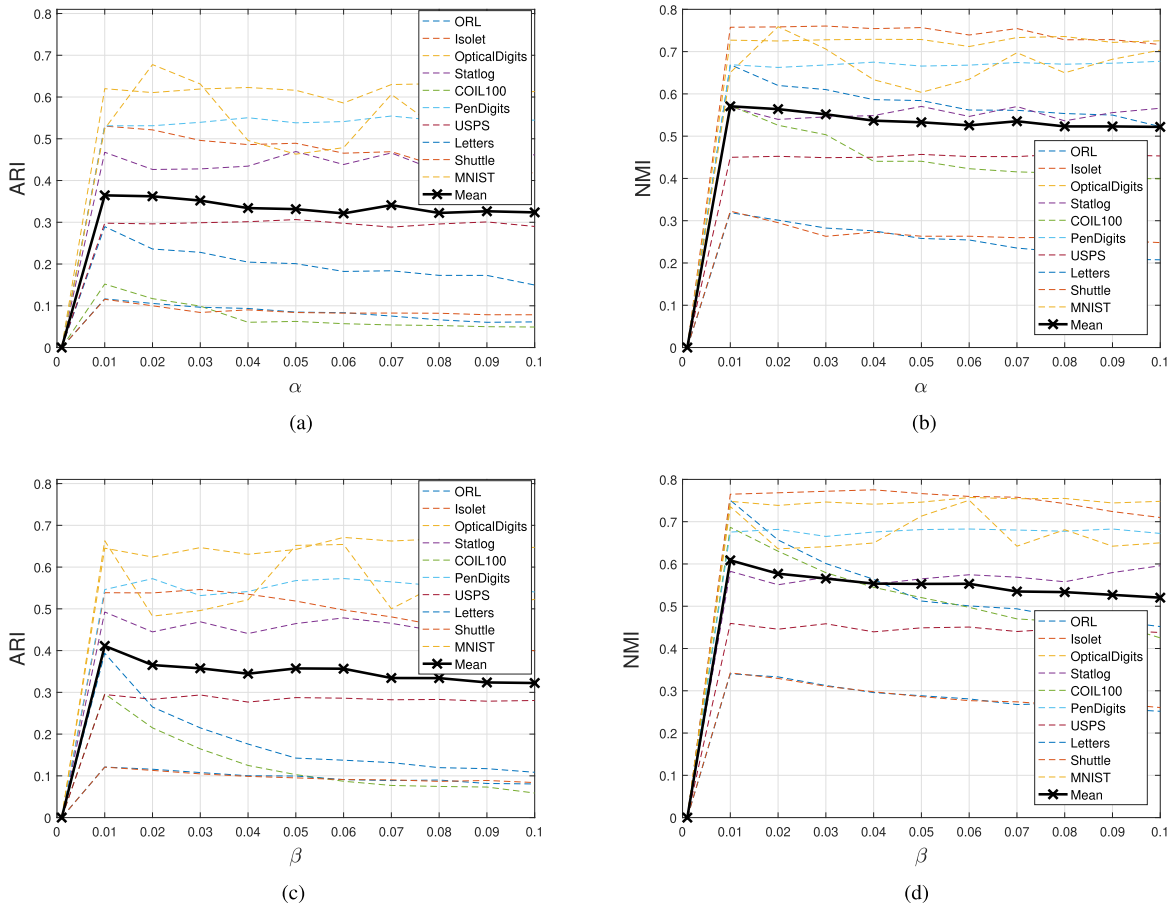


Fig. 4. Effect of parameters on the proposed algorithm. (a) ARI against α. (b) NMI against α. (c) ARI against β. (d) NMI against β.

The comparisons are conducted on a personal computer (Intel i7@3.60 GHz) with 16G RAM and MATLAB 2016b.

C. Evaluation Criteria

In the experiments, the clustering indices ARI [48] and NMI [49] are used to evaluate the clustering result. If a clustering result is highly similar to the ground truth on a dataset, it will yield a high ARI and NMI score.

D. Effectiveness Analysis

We first compare the proposed algorithm with other k-type-based consensus clustering algorithms on all the benchmark datasets. For the KRCC-DE algorithm, we fix $\alpha = 0.01$ and $\beta = 0.01$ and for the KRCC-E algorithm, we set $\alpha = 0.01$ and $\beta = 0$. We run each of these algorithms 50 times to compute the mean and standard deviation of their ARI and NMI on each dataset. Tables II and III show the comparison results on non-categorical datasets, where the number of base clusters k_h for each base clustering is fixed to the number of true clusters k of each dataset. Tables IV and V show the comparison results on non-categorical datasets, where k_h in each base clustering is randomly selected in the interval $[k/2, 2k]$. Tables VI and VII show the comparison results on categorical datasets. According to these tables, we can see that the proposed algorithm is significantly better than other k-type-based consensus clustering algorithms. Moreover, we compared the proposed algorithm with double entropy-norm regularizers (KRCC-DE) and single entropy-norm regularizers (KRCC-E). We found that the double entropy-norm regularizers can further improve the clustering accuracy.

Next, we compare the proposed algorithm with three global-search consensus clustering algorithms on non-categorical datasets, where k_h in each base clustering is randomly selected in the interval $[k/2, 2k]$. Compared to k-type algorithms based on local search, the global-search algorithms need expensive computation costs to get more robust clustering results. However, the local-search performance of k-type algorithms can be improved by an appropriate selection of the initial value U, which makes the average local search results to not reach the same level as the global-search result in clustering accuracy. Therefore, in this comparison, we try to verify whether the proposed algorithm with a good initial value can achieve global-search results. Consequently, we compare the highest ARI and NMI values of the proposed algorithm with 50 different initial U to those of the global-search algorithms. Table VIII shows the comparison results on benchmark datasets. It is noted that while WCT is used on Shuttle and MNIST datasets, it needs a very large size of memory. Thus, we cannot get its results on the two datasets. Thus, in this case, we use "NA" instead of the real values of ARI and NMI in the tables. According to the table, we can

conclude that the proposed algorithm with an appropriate initial value can reach or even exceed the global-search results.

E. Efficiency Analysis

We compare the computation costs of different algorithms on the benchmark datasets, as shown in Table IX. We can see from the table that the running time of the k-type algorithms is less than the global-search algorithms (USENC, MCLA, and WCT). Among these algorithms, WCT needs the highest computation costs. We also see that USENC is very efficient, compared to MCLA and WCT. However, it is not suitable to deal with datasets with a large number of clusters and base clusterings, since the time complexity of USENC is $O(p^3)$. As seen in Table IX, USENC is highly time consuming for the COIL100 dataset. According to the table, we can see that the proposed algorithm (KRCC-DE) has the excellent balance between the effectiveness and efficiency of clustering consensus. In terms of clustering accuracy, the KRCC-DE algorithm is obviously better than other k-type algorithms. In terms of clustering efficiency, the KRCC-DE algorithm is far faster than USENC, MCLA, and WCT. It is worth noting that the proposed algorithm requires additional computations for the entropy-norm regularizers, compared to other k-type algorithms. However, according to the tests, the proposed algorithm is still scalable. Therefore, according to the experiment analysis, we conclude that the proposed algorithm can take low computational costs to effectively solve the consensus clustering problem.

F. Effect of Initialization

The performance of the proposed algorithm is effected by the initialization of U. To evaluate the effect, we test three different initialization methods: (1) randomly selecting a base clustering as initial U; (2) using CU measure [47] to evaluate base clusterings and select one with the maximum CU value; (3) employing k-means++ to compute the initial U. The comparison results are shown in Table X. We can see that using CU measure and k-means++ are two good initialization methods for U. They can enhance the performance of the proposed algorithm, compared to the random selection.

G. Parameter Analysis

Parameters α and β are two important factors that affect the performance of the proposed algorithm. To evaluate their effect on the datasets, we fixed one of the parameters to 0.1 and then tested the other parameter in the interval $[0, 0.1]$ with a step size of 0.01. The results of this analysis are demonstrated in Fig. 4. It is evident that the effects of the parameters vary across datasets, which implies that it is challenging for the proposed algorithm to select α and β values that are general and appropriate for each dataset. To further analyze the impact, we computed the mean ARI and NMI of the proposed algorithm on all the tested datasets for each α and β in Fig. 4. According to the mean lines, we can see that the average performance of the proposed algorithm in the interval $[0, 0.1]$ is relatively stable. Besides, we can see that setting α and β to 0.01 is a good choice for most datasets. In this setting, we can learn a sparse Z which can reduce the uncertain consensus relations and enhance the quality of the final clustering result.

VI. Conclusion

In this article, we present a novel k-relations consensus clustering algorithm developed under the k-type clustering paradigm. We define a cluster representation using consensus relations between the final and base clusters. Furthermore, we propose a new objective function composed of a consensus function to evaluate the consensus between the final and base clusterings, and two entropy-norm regularizers to control the distributions of consensus relations. We design an iterative optimization approach to minimize the objective function. The algorithm can rapidly and accurately capture a good final clustering. To demonstrate the effectiveness and efficiency of the proposed algorithm, we conducted experiments on benchmark datasets and compared it against other k-type-based and several global-search consensus clustering algorithms. The comparison results showed the superiority of the proposed algorithm.

References

[1] A. K. Jain, "Data clustering: 50 years beyond k-means," in *Machine Learning and Knowledge Discovery in Databases*, W. Daelemans, B. Goethals, and K. Morik, Eds. Berlin, Germany: Springer, 2008, pp. 3–4.

[2] C. C. Aggarwal and C. K. Reddy, *Data Clustering: Algorithms and Applications*. Boca Raton, FL, USA: CRC Press, 2014.

[3] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.* Berkeley, CA, USA: Univ. California Press, 1967, pp. 281–297.

[4] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.

[5] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, 1st ed. Boca Raton, FL, USA: CRC Press, 2012.

[6] V. Y. Kiselev et al., "SC3: Consensus clustering of single-cell RNA-seq data," *Nature Methods*, vol. 14, no. 5, pp. 483–486, May 2017.

[7] A. L. N. Fred and A. K. Jain, "Data clustering using evidence accumulation," in *Proc. Int. Conf. Pattern Recognit.*, vol. 4, Aug. 2002, pp. 276–280.

[8] Y. Yang and K. Chen, "Temporal data clustering via weighted clustering ensemble with different representations," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 2, pp. 307–320, Feb. 2011.

[9] 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.

[10] Z. Yu, X. Zhu, H.-S. Wong, J. You, J. Zhang, and G. Han, "Distribution-based cluster structure selection," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3554–3567, Nov. 2017.

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

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

[13] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, pp. 1–10.

[14] 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.

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

[16] U. Endriss and U. Grandi, "Graph aggregation," *Artif. Intell.*, vol. 245, pp. 86–114, Apr. 2017.

[17] P. Zhou, L. Du, and X. Li, "Self-paced consensus clustering with bipartite graph," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 1–10.

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

[19] A. Topchy, B. Minaei-Bidgoli, A. K. Jain, and W. F. Punch, "Adaptive clustering ensembles," in *Proc. 17th Int. Conf. Pattern Recognit.*, vol. 1, Aug. 2004, pp. 272–275.

[20] P. Hore, L. O. Hall, and D. B. Goldgof, "A scalable framework for cluster ensembles," *Pattern Recognit.*, vol. 42, no. 5, pp. 676–688, May 2009.

[21] B. Long, Z. Zhang, and P. S. Yu, "Combining multiple clusterings by soft correspondence," in *Proc. 5th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2005, pp. 1–8.

[22] C. Boulis and M. Ostendorf, "Combining multiple clustering systems," in *Proc. Eur. Conf. Princ. Pract. Knowl. Discovery Databases*, 2004, pp. 1–10.

[23] D. Cristofor and D. Simovici, "Finding median partitions using information theoretical based genetic algorithms," *J. Universal Comput. Sci.*, vol. 8, no. 2, pp. 153–172, 2002.

[24] 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.

[25] N. Nguyen and R. Caruana, "Consensus clusterings," in *Proc. 7th IEEE Int. Conf. Data Mining*, Oct. 2007, pp. 607–612.

[26] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining Knowl. Discovery*, vol. 2, no. 3, pp. 283–304, Sep. 1998.

[27] 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.

[28] 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.

[29] L. Bai, J. Liang, C. Dang, and F. Cao, "The impact of cluster representatives on the convergence of the K-modes type clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1509–1522, Jun. 2013.

[30] L. Bai, J. Liang, H. Du, and Y. Guo, "An information-theoretical framework for cluster ensemble," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 8, pp. 1464–1477, Aug. 2019.

[31] G. He, W. Jiang, R. Peng, M. Yin, and M. Han, "Soft subspace based ensemble clustering for multivariate time series data," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 14, 2022, doi: 10.1109/TNNLS.2022.3146136.

[32] Y. Yang and J. Jiang, "Adaptive bi-weighting toward automatic initialization and model selection for HMM-based hybrid meta-clustering ensembles," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1657–1668, May 2019.

[33] Z. Yu, D. Wang, X.-B. Meng, and C. L. P. Chen, "Clustering ensemble based on hybrid multiview clustering," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 6518–6530, Jul. 2022.

[34] D. Huang, C.-D. Wang, J.-H. Lai, and C.-K. Kwoh, "Toward multi-diversified ensemble clustering of high-dimensional data: From subspaces to metrics and beyond," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 12231–12244, Nov. 2022.

[35] P. Zhou, L. Du, Y.-D. Shen, and X. Li, "Tri-level robust clustering ensemble with multiple graph learning," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 12, pp. 11125–11133.

[36] Z. Hao, Z. Lu, G. Li, F. Nie, R. Wang, and X. Li, "Ensemble clustering with attentional representation," *IEEE Trans. Knowl. Data Eng.*, early access, Jul. 5, 2023, doi: 10.1109/TKDE.2023.3292573.

[37] Y. Jia, S. Tao, R. Wang, and Y. Wang, "Ensemble clustering via co-association matrix self-enhancement," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 6, 2023, doi: 10.1109/TNNLS.2023.3249207.

[38] Y. Shi, Z. Yu, C. L. P. Chen, and H. Zeng, "Consensus clustering with co-association matrix optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 9, 2022, doi: 10.1109/TNNLS.2022.3201975.

[39] P. Rathore, J. C. Bezdek, S. M. Erfani, S. Rajasegarar, and M. Palaniswami, "Ensemble fuzzy clustering using cumulative aggregation on random projections," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 3, pp. 1510–1524, Jun. 2018.

[40] L. Bai, J. Liang, and Y. Guo, "An ensemble clusterer of multiple fuzzy k-means clusterings to recognize arbitrarily shaped clusters," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 6, pp. 3524–3533, Dec. 2018.

[41] L. Bai and J. Liang, "A three-level optimization model for nonlinearly separable clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 3211–3218.

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

[43] D. H. Fisher, "Knowledge acquisition via incremental conceptual clustering," *Mach. Learn.*, vol. 2, no. 2, pp. 139–172, Sep. 1987.

[44] L. Bai and J. Liang, "A categorical data clustering framework on graph representation," *Pattern Recognit.*, vol. 128, Aug. 2022, Art. no. 108694.

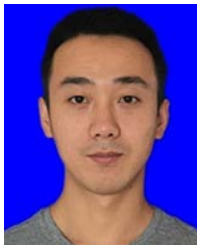
[45] X. Liu, "SimpleMKKM: Simple multiple kernel K-means," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5174–5186, Apr. 2023.

[46] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2007, pp. 1–12.

[47] B. Mirkin, "Reinterpreting the category utility function," *Mach. Learn.*, vol. 45, no. 2, pp. 219–228, Nov. 2001.

[48] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Stat. Assoc.*, vol. 66, no. 336, pp. 846–850, Dec. 1971.

[49] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Conditional entropy and mutual information," in *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. New York, NY, USA: Cambridge Univ. Press., 2007.



Liang Bai received the Ph.D. degree in computer science from Shanxi University, Taiyuan, China, in 2012.

He is currently a Professor with the School of Computer and Information Technology, Shanxi University. He has published several papers in his research fields, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON FUZZY SYSTEMS, International Conference on Machine Learning (ICML), ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), and AAAI Conference on Artificial Intelligence (AAAI). His research interests are in the areas of cluster analysis.

Jiye Liang (Senior Member, IEEE) received the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2000.

He is a Professor with the Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan, China. He has published more than 200 international papers in his research fields, including the *Journal of Artificial Intelligence*, *Journal of Machine Learning Research*, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Data Mining and Knowledge Discovery*, ICML, KDD, and AAAI. His research interests include artificial intelligence, granular computing, data mining, and machine learning.