

# k-HyperEdge Medoids for Clustering Ensemble

Feijiang Li<sup>1,2</sup>, Jieting Wang<sup>1,2</sup>, Liuya Zhang<sup>1</sup>, Yuhua Qian<sup>1,2\*</sup>, Shuai Jin<sup>1</sup>, Tao Yan<sup>1,2</sup>, Liang Du<sup>1,2</sup>

<sup>1</sup>Institute of Big Data Science and Industry, Shanxi University

<sup>2</sup>Key Laboratory of Evolutionary Science Intelligence of Shanxi Province, Taiyuan, Shanxi, China  
jinchengqyh@126.com

## Abstract

Clustering ensemble has been a popular research topic in data science due to its ability to improve the robustness of the single clustering method. Many clustering ensemble methods have been proposed, most of which can be categorized into clustering-view and sample-view methods. The clustering-view method is generally efficient, but it could be affected by the unreliability that existed in base clustering results. The sample-view method shows good performance, while the construction of the pairwise sample relation is time-consuming. In this paper, the clustering ensemble is formulated as a k-HyperEdge Medoids discovery problem and a clustering ensemble method based on k-HyperEdge Medoids that considers the characteristics of the above two types of clustering ensemble methods is proposed. In the method, a set of hyperedges is selected from the clustering view efficiently, then the hyperedges are diffused and adjusted from the sample view guided by a hyperedge loss function to construct an effective k-HyperEdge Medoid set. The loss function is mainly reduced by assigning samples to the hyperedge with the highest degree of belonging. Theoretical analyses show that the solution can approximate the optimal, the assignment method can gradually reduce the loss function, and the estimation of the belonging degree is statistically reasonable. Experiments on artificial data show the working mechanism of the proposed method. The convergence of the method is verified by experimental analysis of twenty data sets. The effectiveness and efficiency of the proposed method are also verified on these data, with nine representative clustering ensemble algorithms as reference.

## Introduction

Data clustering is an interesting technique to discover the group structure inherent in a set of data samples without labels (Vega-Pons and Ruiz-Shulcloper 2011). The general objective of data clustering is discovering a set of clusters in which the samples in the same cluster share some kind of high similarity while the samples in different clusters share high diversity. With the complexity of the data development, many traditional clustering methods suffer from the robust problem. To handle this, the clustering ensemble technique that combines multiple diverse clustering results has been illustrated as an effective strategy to improve the robustness

of a single clustering result.

The clustering ensemble problem is first proposed by Strehl and Ghosh (Strehl and Ghosh 2002), which is described as combining multiple clustering results of a set of objects into a single consolidated clustering that shares most information with the ensemble base without accessing the features of the data samples or the algorithms that produce these clustering results. Following this description, many methods have been proposed to solve the clustering ensemble problem. Most of the existing clustering ensemble methods can be roughly categorized into clustering-view and sample-view.

As for the clustering-view methods, their main idea is to generate a clustering result that is similar to the clustering results in the ensemble base. To generate such a clustering result, a simple strategy is to first align the labels of the different clustering results and then merge them using a fusion strategy, such as voting. Another strategy is to give a clustering similarity measure and then use a dynamic programming method to produce the integrated result. Generally, due to that the number of clustering results is much less than that of samples, most of the clustering-view methods are efficient. However, the unreliability that existed in the base clustering results brings a negative effect on the ensemble performance (Zhou et al. 2022).

As for the sample-view methods, their main idea is to generate a clustering result that the samples in the same cluster share high similarity. There are two issues for this type of method, one is how to evaluate the similarity between samples and the other one is how to discover such a clustering result. As for the first issue, the co-association relation that evaluates the times that two samples appear in the same cluster is one of the most utilized sample pair similarity measures (Li, Qian, and Wang 2021). As for the second issue, many partition methods have been proposed, in which the clustering strategies (Liu et al. 2017), graph partition strategies (Strehl and Ghosh 2002), or heuristic strategies (Li et al. 2019) are utilized. This type of method has shown excellent effectiveness in handling the clustering ensemble problem. However, calculating the pairwise sample relationships and optimizing a clustering loss is often time-consuming.

With consideration of the characteristics of the above two types of clustering ensemble methods, in this paper, a method based on k-HyperEdge Medoids is developed,

which integrates clustering-view and sample-view. First, the description of k-HyperEdge Medoids is given. For a hypergraph, the k-HyperEdge Medoids are  $k$  nonoverlapping hyperedges that each hyperedge in the hypergraph could find a similar hyperedge in the k-HyperEdge Medoids. Then the clustering ensemble problem is formulated as a k-HyperEdge Medoids discovering problem. We propose a method to solve the problem and conduct theoretical and experimental analysis about the method.

The main work of this paper in terms of methods, theory, and experiments includes:

**Method.** A k-HyperEdge Medoids discovery method is proposed<sup>1</sup>. This method mainly contains three steps, which are k-HyperEdge initialization, k-HyperEdge diffusion, and k-HyperEdge adjustment. The k-HyperEdge initialization step is from the clustering view, which utilizes the k-medoids algorithm to discover a set of sub-hyperedges medoids. The k-medoids algorithm is suitable for arbitrary distance measurement and robust to noise (Tiwari et al. 2020). The k-HyperEdge diffusion and k-HyperEdge adjusting improve the quality of the initial hyperedges methods from the sample view. The improvement is realized by updating the sample assignment guided by a hyperedge loss function based on cluster quality. The loss function is mainly reduced by assigning samples to the hyperedge with the highest degree of belonging.

**Theory.** Three main issues in the proposed method are theoretically analyzed. Firstly, given the initial hyperedge set, the solution has been proven to be able to approximate to the optimal, indicating its ability to guide model learning. Secondly, assigning samples to the hyperedge with the highest belonging degree has been proven to be able to reduce the value of the loss function. Finally, the method for estimating the belonging degree has been proven to be statistically reasonable.

**Experiment.** The proposed method is experimentally analyzed from three aspects. The working mechanism of the proposed method is illustrated on artificial data. The effectiveness of the method in solving the clustering ensemble problem is illustrated on twenty real-world data sets compared with the other nine representative clustering ensemble methods. The time complexity of the proposed method is analysed and the time costs of the compared methods on real data sets are reported. The results show the effectiveness and efficiency of the proposed method.

## Preliminaries and Related Work

### Clustering Ensemble

As for clustering ensemble, what we can obtain is a set of clustering results  $\Pi = \{\pi^1, \pi^2, \dots, \pi^l\}$  on a data set  $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ , where  $l$  is number of base clustering result and  $n$  is the number of data point. The  $j$ th clustering result contains  $k_j$  clusters  $\pi^j = \{c_1^j, c_2^j, \dots, c_{k_j}^j\}$  and  $\mathcal{X} = \bigcup_{i=1}^{k_j} c_i^j$ . The cluster label of  $x_i$  in  $\pi^j$  is  $\pi^j(x_i)$ . For  $\Pi$ , there exists  $n_c = \sum_{i=1}^h k_i$  base clusters. The task

<sup>1</sup>Demo code is at <https://github.com/FeijiangLi/Code-k-HyperEdge-Medoids-for-Clustering-Ensemble-AAAI>

of the clustering ensemble is to discover a clustering result  $\pi^* = \{c_1^*, c_2^*, \dots, c_k^*\}$  that shares the most information with  $\Pi$ , where  $k$  is the expected number of clusters. In the original description of the clustering ensemble, the generation of  $\pi^*$  does not access the features or algorithms that determine these clustering results (Strehl and Ghosh 2002). This paper follows this description.

Many clustering ensemble methods have been proposed, accompanied by various classification schemes for these methods (Vega-Pons and Ruiz-Shulcloper 2011). From the view of processing, most of the clustering ensemble methods can be roughly categorized into clustering-view and sample-view.

### Clustering-view Clustering Ensemble

The clustering ensemble methods from the clustering-view are mainly guided by the objective that the integrated clustering should be similar to the base clustering results. Then, the objective can be expressed as follows,

$$\pi^* = \arg \max_{\pi} \sum_{\pi' \in \Pi} \text{sim}(\pi, \pi'), \quad (1)$$

where  $\text{sim}(\pi, \pi')$  is a similarity between  $\pi$  and  $\pi'$ .

The optimization of the objective is an NP-hard problem, most of the proposed methods generally obtain an approximate solution. One direct approach is to first align the cluster labels and then merge them using a voting strategy (Zhou and Tang 2006). Ayad et al. (Ayad and Kamel 2008, 2010) introduced an idea of cumulative voting that utilizes a probabilistic mapping method for cluster label alignment, and computed empirical probability summarizing the ensemble. Carpineto et al. (Carpineto and Romano 2012) proposed a probabilistic rand index and cast clustering ensemble as an optimization problem of the PRI between a target result and the ensemble base. Li et al. (Li et al. 2017) proposed a Dempster-Shafer evidence theory-based clustering ensemble method that substitutes voting with evidence theory. Huang et al. (Huang, Gao, and Akhavan 2023) proposed an ensemble hierarchical clustering algorithm based on merits at the clustering level, the main idea of which is to re-cluster the selected clusters to generate a consensus clustering. Guilbert et al. (Guilbert et al. 2022) proposed an anchored constrained clustering ensemble method that constructs an allocation matrix and uses an integer linear programming model to find the consensus partition.

Generally, these methods are simple and efficient. However, the performance of these methods relies on the reliability of the underlying clustering ensemble. Due to the diversity among the base clustering results, low-reliability results exist in the ensemble inevitably, which could affect the performance of many clustering-view clustering ensemble methods.

### Sample-view Clustering Ensemble

The clustering ensemble methods from the sample view are mainly guided by the objective that the intra-cluster samples in the integrated clustering should be similar to each other. This objective is equivalent to finding a clustering result with

high cluster quality, where cluster quality is measured by the consistency among intra-cluster samples. The objective can be expressed as follows

$$\pi^* = \arg \max_{\pi} \sum_{c_i \in \pi} Q(c_i), \quad (2)$$

where  $Q(c_i)$  indicates the quality of  $c_i$ . From the sample view,  $Q(c_i)$  is generally evaluated by the consistency between samples in  $c_i$ .

This type of methods are generally realized by optimizing a clustering loss function. Fred et al. (Fred and Jain 2005) first proposed the concept of the co-association matrix (CA matrix) and proposed an evidence accumulation method that utilizes the idea of hierarchical clustering. The spectral ensemble clustering method proposed by Liu utilizes the spectral clustering method to optimize the clustering loss (Liu et al. 2017). Iam-On proposed a type of link-based approach to refine the cluster-based matrix, and k-means, PAM, and graph partitioning algorithms are used to generate the final result. Graph partitioning techniques have been used in many clustering ensemble methods to optimize the clustering loss, such as CSPA, HGPA, MCLA (Strehl and Ghosh 2002), HBGF (Fern and Brodley 2004), PTGP (Huang, Lai, and Wang 2016), U-SENC (Huang et al. 2020), SCCBG (Zhou, Du, and Li 2020), etc. Hao et al. (Hao et al. 2024) proposed a clustering ensemble method with attentional representation that jointly optimizes the clustering loss and the reconstruction loss. In addition, many enriching methods are proposed to improve the relation between samples, such as the probability trajectory proposed by Huang et al. (Huang, Lai, and Wang 2016), the propagating cluster-wise similarities with meta-cluster proposed by Huang et al. (Huang et al. 2021), co-association matrix self-enhancement proposed by Jia et al. (Jia et al. 2023), dense representation proposed by Zhou et al. (Zhou, Zheng, and Pan 2019), stable clustering ensemble method based on evidence theory proposed by Fu et al. (Fu et al. 2022), etc.

This type of method has shown excellent ensemble performance. However, for such methods, the construction of the co-association relationship is time-consuming, so as with many enriching methods, the time cost of which is generally at least  $O(n^2)$ , where  $n$  is the number of samples. Although some methods utilize representative point selection or down-sampling strategies for acceleration, this affects the stability and effectiveness of the methods.

In this paper, considering the characteristics of the above two types of clustering ensemble methods, we propose a clustering ensemble method based on k-HyperEdge Medoids.

## k-HyperEdge Medoids

It is natural to represent a set of clustering results as a hypergraph (Zhou et al. 2022). In this section, we introduce the k-HyperEdge Medoids.

A hypergraph  $\mathcal{G}$  can be represented as  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$ , where  $\mathcal{V}$  is the node set,  $\mathcal{E}$  is the hyperedge set, and  $\mathcal{W}$  is the set of hyperedge weight. Mainly different from the traditional edge in the graph, the hyperedge links a set of nodes

in the graph that can represent complex multivariate relationships more flexibly. For example, in a hypergraph with a node set  $\mathcal{V} = \{x_1, x_2, x_3, x_4, x_5\}$ , a hyperedge  $e$  could connect multiple nodes, such as  $e = \{x_1, x_2, x_5\}$ , representing a common relationship or connection among  $x_1$ ,  $x_2$ , and  $x_5$ .

Given a set of clustering results  $\Pi = \{\pi^1, \pi^2, \dots, \pi^l\}$  on a data set  $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ , from cluster view,  $\Pi$  can be expressed as  $\Pi_c = \{c_1^1, c_2^1, \dots, c_{k_1}^1, \dots, c_1^l, c_2^l, \dots, c_{k_l}^l\}$ . For convenience, we denote

$$\Pi_c = \{c_1, c_2, \dots, c_{n_c}\}. \quad (3)$$

The hypergraph representation of  $\Pi_c$  is  $\mathcal{G}(\Pi_c) = \{\mathcal{V}(\Pi_c), \mathcal{E}(\Pi_c), \mathcal{W}(\Pi_c)\}$ , where  $\mathcal{V}$  is the set of data points  $\mathcal{V}(\Pi_c) = \mathcal{X}$ ,  $\mathcal{E}$  is the set of clusters  $\mathcal{E}(\Pi_c) = \Pi_c$ , and  $\mathcal{W}$  is the weight of each cluster. The cluster weight can be quantified by cluster quality evaluation. In this paper, the cluster weight is not a focus and is set equally.

**Definition 1. (k-HyperEdge Medoids)** For a hypergraph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$ , the k-HyperEdge Medoids  $\mathcal{E}_m(\mathcal{G})$  are k hyperedges

$$\mathcal{E}_m(\mathcal{G}) = \arg \max_{\mathcal{E}'} \sum_{e \in \mathcal{E}} \max_{e' \in \mathcal{E}'} \text{sim}(e, e'), \quad (4)$$

$$\text{s.t. } \bigcup_{e \in \mathcal{E}_m} e = \mathcal{V}, \quad \forall e_i, e_j \in \mathcal{E}_m, i \neq j, e_i \cap e_j = \emptyset.$$

From Definition 1, it is known that the k-HyperEdge Medoids are k representative hyperedges that maximize overall similarity with all hyperedges in the hypergraph. Additionally, the k-HyperEdge Medoids are non-overlapping and collectively cover all nodes. From the view of clustering ensemble, each cluster can be treated as a hyperedge, and multiple clusters form a hypergraph. Then the k-HyperEdge Medoids of  $\mathcal{G}(\Pi_c)$  can be treated as a clustering ensemble result of  $\Pi_c$ , and the clustering ensemble problem can be formulated as a k-HyperEdge Medoids discovering problem.

## A k-HyperEdge Medoids Discovery Method

In this section, a heuristic k-HyperEdge Medoids discovery method is proposed to approximatively solve Formula (4). The method mainly contains three steps, which are k-hyperedge initialization, k-hyperedge diffusion, and k-hyperedge adjustment. The main framework of the k-HyperEdge Medoids discovers method is shown as Figure 1. Then processes of each step are introduced.

### k-HyperEdge Initialization

The objective of the k-hyperedge initialization step is finding a set of  $k$  hyperedges from  $\mathcal{E}(\Pi_c) = \{c_1, c_2, \dots, c_{n_c}\}$ , i.e.  $\mathcal{E}_i = \{e_{i_1}, e_{i_2}, \dots, e_{i_k}\} \subset \mathcal{E}$ , that minimizes the following loss:

$$\mathcal{L}_{\text{ini}}(\mathcal{E}_i) = \sum_{i=1}^{n_c} \min_{e_i \in \mathcal{E}_i} (1 - \text{sim}(e_i, c_i)), \quad (5)$$

where  $\text{sim}(e_i, c_i)$  is a similarity measure between  $e_i$  and the  $i$ th hyperedge  $c_i$ . To prevent the discovery of an excessively large hyperedge, in this step, we use the following similarity measure

$$\text{sim}(e_i, c_i) = \frac{|e_i \cap c_i|}{|c_i|}. \quad (6)$$

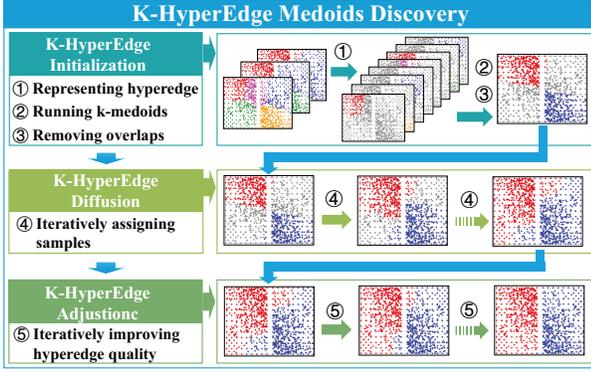


Figure 1: The framework of a k-HyperEdge Medoids discovery method

As shown by Equation (5), what we want to find is  $k$  hyperedges, in which each hyperedge in  $\mathcal{E}(\Pi_c)$  could find a similar representation. Equation (5) is equivalent to the loss function of the k-medoids algorithm. Then the k-medoids algorithm can be directly utilized to solve (5). Meanwhile, the k-medoids algorithm has several advantages in this step. Firstly, the medoids are real points, which are interpretable. Secondly, k-medoids are not sensitive to noise, which could mitigate the negative effects of unreliability clusters. Thirdly, the distance or consistency function in k-medoids can be arbitrary, which makes it possible to adopt Equation (6) (Tiwari et al. 2020).

After run k-medoids algorithm on  $\mathcal{E}(\Pi_c)$ ,  $k$  hyperedges are obtained, which is noted as  $\mathcal{E}i = \{ei_1, ei_2, \dots, ei_k\}$ . Due to the selected hyperedges could come from different clusterings, which may not satisfy  $\bigcup_{ei \in \mathcal{E}i} ei = \mathcal{V}$  and  $\forall ei_i, ei_j \in \mathcal{E}i, i \neq j, ei_i \cap ei_j = \emptyset$ . To handle the first condition, we remove the intersection regions between different hyperedges and regenerate  $\mathcal{E}i = \{ei_1, ei_2, \dots, ei_k\}$ . To handle the second one, we conduct the following hyperedge diffusion step. The processes of the k-HyperEdge initialization step are shown as Algorithm 1 in Appendix A.1<sup>2</sup>.

### k-HyperEdge Diffusion

The purpose of the k-HyperEdge diffusion step is to make each sample in  $\mathcal{X}$  belong to a hyperedge. This step is realized by iteratively assigning a sample to the hyperedge which it is most likely to belong. Firstly, in this step, we initialize  $\mathcal{E}d = \mathcal{E}i$ . In each iteration, a set of samples that with high belonging confidence will be reassigned to the k initialization hyperedges  $\mathcal{E}d = \{ed_1, ed_2, \dots, ed_k\}$ . With the progress of iteration, the number of samples to be assigned gradually increases. The diffusion step stops when the number of the assigned sample is equal to the number of data points. The calculation of the sample confidence and the number of samples to be assigned are then introduced.

Firstly, given  $\mathcal{E}(\Pi_c)$ , the belonging degree of sample  $x_i$  to a hyperedge  $ed$  is defined as the average similarity between  $ed$  and the hyperedges that  $x_i$  belongs to in  $\mathcal{E}(\Pi_c)$ .

<sup>2</sup>The Appendix is in (Li et al. 2024)

The equation is

$$b(x, ed) = \sum_{c_x \in \mathcal{E}(\Pi_c)} |c_x \cap ed|, \quad (7)$$

where  $c_x \in \mathcal{E}(\Pi_c)$ .

Based on Equation (7), the belongings of a sample to all the hyperedges in  $\mathcal{E}d$  can be obtained. Then, the hyperedge to which sample  $x_i$  belongs is obtained by

$$ed^*(x_i) = \arg \max_{ed \in \mathcal{E}d} b(x_i, ed). \quad (8)$$

The sample  $x_i$  will be assigned to  $ed^*$  with a confidence degree of

$$m(x_i) = b(x_i, ed^*(x_i)) - \max_{\substack{ed_j \neq ed^*(x_i) \\ ed_j \in \mathcal{E}d}} b(x_i, ed_j), \quad (9)$$

which is the difference between the maximum and the second maximum degree of belonging.

With Equation (9), the confidence of all the samples in  $\mathcal{X}$  can be obtained, and the samples with high confidence will be selected and re-assigned to the hyperedge set in hand. Assuming  $\mathcal{E}d$  is the present hyperedge set, the number of samples in  $\mathcal{E}d$  is

$$n_a = \left| \bigcup_{ed \in \mathcal{E}d} ed \right|. \quad (10)$$

With the overall consideration of  $n_a$  and  $n$ , the number of samples  $n_s$  to be selected is calculated as

$$n_s = \min\{(n_a + \max\{\lceil \frac{n - n_a}{k} \rceil, \lceil \sqrt{n} \rceil\}), n\}. \quad (11)$$

For an edge  $ed$ , the diffusion is realized by

$$ed = \{x | ed^*(x) = ed, m(x) \geq m_{(n_s)}(\mathcal{X})\}, \quad (12)$$

where  $m_{(n_s)}(\mathcal{X})$  is the  $n_s$ -th largest element in the set  $\{m(x_1), m(x_2), \dots, m(x_n)\}$ .

Repeating the above diffusion process, the number of samples in  $\mathcal{E}d$  will increase. When  $n_a = n$ , the k-HyperEdge diffusion process stops, and the obtained hyperedge set after this step is noted as  $\mathcal{E}d = \{ed_1, ed_2, \dots, ed_k\}$ , which satisfies  $\bigcup_{ed \in \mathcal{E}d} ed = \mathcal{V}$  and  $\forall ed_i, ed_j \in \mathcal{E}d, i \neq j, ed_i \cap ed_j = \emptyset$ . The processes of the k-HyperEdge diffusion step are shown as Algorithm 2 in Appendix A.1. Further, the quality of  $\mathcal{E}d$  is improved by the following k-HyperEdge adjustment step.

### k-HyperEdge Adjustment

After assigning all the samples in  $\mathcal{X}$ , we further improve the quality of the hyperedge set from the sample view according to Equation (2). Assuming the hyperedge set is  $\mathcal{E}a = \{ea_1, ea_2, \dots, ea_k\}$ , the quality of  $ea_i$  is

$$Q(ea_i) = \sum_{x \in ea_i} b(x, ea_i). \quad (13)$$

The loss function of the k-HyperEdge adjustment step is

$$\mathcal{L}_{\text{adj}}(\mathcal{E}a) = \sum_{i=1}^k (n|ea_i| - Q(ea_i)) \quad (14)$$

$$= \sum_{i=1}^n (n - b(x_i, ea^*(x_i))).$$

The loss value can be iteratively reduced by updating  $\mathcal{E}a$  until convergence.

Assuming the hyperedge set in the  $t$ -th iteration is  $\mathcal{E}a^t = \{ea^t_1, ea^t_2, \dots, ea^t_k\}$ , the belongings of a sample  $x_i$  to a hyperedge  $ea^t_j$  in  $\mathcal{E}a^t$  can be calculated by Equation (7), noted as  $b(x_i, ea^t_j)$ . Then, the hyperedge that  $x_i$  belongs to in the  $(t+1)$ -th iteration is

$$ea^{(t+1)*}(x_i) = \arg \max_{ea^t \in \mathcal{E}a^t} b(x_i, ea^t). \quad (15)$$

The  $i$ -th hyperedge is updated by

$$ea^{(t+1)}_i = \{x | ea^{(t+1)*}(x) = ea^t_i, x \in \mathcal{X}\}. \quad (16)$$

The  $k$  updated hyperedges constitute the hyperedge set  $\mathcal{E}a^{(t+1)}$ . The adjution will stops when  $\mathcal{E}a^{(t+1)} = \mathcal{E}a^t$ . The processes of the k-HyperEdge adjution step are shown as Algorithm 3 in Appendix A.1.

The final hyperedge set is the discovered k-HyperEdge Medoids and noted as  $\mathcal{E}_m = \{e_1, e_2, \dots, e_k\}$ . The final corresponding clustering ensemble result is

$$\pi^* = \{c_1 = e_1, c_2 = e_2, \dots, c_k = e_k\}. \quad (17)$$

We note the above clustering ensemble method based on the construction of k-HyperEdge Medoids as CEHM. The processes of CECH is shown as Algorithm 4 in Appendix A.1.

## Theoretical Analysis

In this section, we give three theorems about the key element in the method to show the rationality.

First, given a  $g$ -approximate initial hyperedge set  $\mathcal{E}a^0$ , we prove that the solution can approximate to the optimal, which is reflected by the following theorem.

**Theorem 1.** *let  $\mathcal{E}a^0 = \{ea^0_1, ea^0_2, \dots, ea^0_k\}$  be a set of hyperedges produced by a  $g$ -approximate algorithm based on a base cluster set  $\Pi_{c_x}$ , and Let*

$$\text{dist}(ea_i, ea_j) = n - |ea_i \cap ea_j|, \quad (18)$$

$$\begin{aligned} \phi(ea_i; ea_j) &= \sum_{x \in ea_j} (n - b(x, ea_i)) \\ &= \sum_{x \in ea_j} \left( n - \frac{1}{l} \sum_{x \in c_x} |c_x \cap ea_i| \right), \end{aligned} \quad (19)$$

then for any hyperedge set, denoted by  $\mathcal{E}a = \{ea_1, ea_2, \dots, ea_k\}$ , we have for any  $ea_i$ , there exists  $ed^0_i$  satisfies  $\text{dist}(ea_i, ea^0_i) \leq \frac{(g+1)\phi(ea_i; *)}{|ea_i|}$ .

Secondly, we prove that assigning samples to the hyperedge with the highest belonging degree can reduce the value of the loss function.

**Theorem 2.** *Assuming the optimal hyperedge of sample  $x$  is the  $i$ -th hyperedge:  $ea^*(x) = ea_i$ ; assuming a hyperedge sets  $\mathcal{E}a^1 = \{ea^1_1, ea^1_2, \dots, ea^1_k\}$  and  $x$  is in the*

*$i$ -th hyperedge in  $\mathcal{E}a^1$ :  $x \in ea^1_i$ ; assuming another hyperedge set  $\mathcal{E}a^2$  based on  $\mathcal{E}a^1$ , and in  $\mathcal{E}a^2$ ,  $x$  is moved from the  $i$ -th hyperedge into the  $j$ -th hyperedge: for  $\forall p \in \{1, 2, \dots, k\} / \{i, j\}$ ,  $ea^2_i = ea^1_i/x$ ,  $ea^2_j = ea^1_j \cup x$ . Then, we have  $\mathcal{L}_{\text{adj}}(\mathcal{E}a^1) \leq \mathcal{L}_{\text{adj}}(\mathcal{E}a^2)$ .*

Thirdly, we prove that using neighbors to estimate the belonging degree is statistically reasonable. To evaluate the similarity between  $x$  and  $ea$ , Equation (7) uses the number of samples in the intersection set of the cluster  $c_x$  and  $ea$ . Indeed, to estimate the posterior probability  $\eta(X) = \mathbb{P}(x \in ea | X = x)$ , the essence of Equation (7) is to use the empirical conditional posterior probability

$$\hat{\eta}_N(x) = \frac{1}{N_{c_x}} \sum_{i: X_i \in c_x} \mathbb{I}\{X_i \in ea\}, \quad (20)$$

where  $N_{c_x}$  is the number of samples in  $c_x$ .

To evaluate of the estimator  $\hat{\eta}_N(x)$ , we explore the upper bound of the estimation error  $\mathbb{E}|\hat{\eta}_N(X) - \eta(X)|$ , which is reflected by the following theorem.

**Theorem 3.** *Consider an estimator  $\hat{\eta}_N(X)$  of  $\eta(X)$  as defined above. Suppose that the second moment of  $\eta(X)$  of  $ea$  exists:  $\mathbb{E}_{x \in ea} \eta(X)^2 < +\infty$  and the difference in probability density between set  $ea$  and the whole set  $\mathcal{X}$  is bounded: that is, there exists  $C$ , such that  $\mathbb{E}_{x \in ea} \left( \frac{\mathbf{m}(x|x \in \mathcal{X})}{\mathbf{m}(x|x \in ea)} \right)^2 < C$ . Then*

$$\mathbb{E}|\hat{\eta}_N(X) - \eta(X)| \rightarrow 0 \text{ if}$$

$$N_{c_x} \rightarrow \infty, \quad (21)$$

$$\mathbb{E}_{x \in ea} \left( \frac{\mathbf{m}(x|x \in c_x)}{\mathbf{m}(x|x \in ea)} - 1 \right)^2 \rightarrow 0, \quad (22)$$

$$\mathbb{V}_{x \in ea}(\eta(X)) \rightarrow 0, \quad (23)$$

where  $\mathbf{m}(\cdot)$  is the probability density function and  $\mathbb{V}$  is the variance operator.

Theorem 3 give three intuitively reasonable and principled rules to design a good estimator:

- The ensemble generation method should guarantee each  $c_x$  has many samples, as suggested by Formula (21).
- The sets  $c_x$  and  $ea$  should have a high similarity, as suggested by Formula (22). In our method, according to Equation (7) and Equation (15), there are many samples in the intersection of  $ea$  and  $c_x$ , which follows this suggestion.
- For an ideal k-HyperEdge Medoids discovering problem, the posterior probabilities in  $ea$  should be concentrated, as suggested by Formula (23).

The proofs of the above theorems are given in the Appendix A.2 to A.4, respectively.

## Experimental Analyses

The CEHM is experimentally analyzed from three aspects: the demonstration of working mechanisms on artificial data, convergence demonstration on real data, and ensemble performance comparison with representative methods. The experimental analyses are conducted on a PCWIN 64 computer with 64G memory.

Number	Data name	n	d	k
1	iris	150	4	3
2	wine	178	13	3
3	seeds	210	7	3
4	heart	294	12	2
5	soybean-train	307	35	18
6	ecoli	336	7	8
7	dermatology	366	34	6
8	low-res-spect	531	100	9
9	breast-cancer-wisc-diag	569	30	2
10	energy	768	8	3
11	lbp-riu-gris	1022	25	3
12	semeion	1593	256	10
13	statlog-landsat-test	2000	36	6
14	cardiotocography-3clases	2126	21	3
15	statlog-landsat-train	4435	36	6
16	twonorm	7400	20	2
17	mushroom	8124	21	2
18	statlog-shuttle	43500	9	7
19	PenDigits	10992	16	10
20	USPS	11000	256	10

Table 1: Description of the data sets

For a data set, the clustering result set is generated by running the k-means algorithm multiple times with different initial centers and different cluster numbers  $k$ . For a data set with  $n$  samples and  $k$  clusters, the numbers of clusters in the base clustering results are randomly selected in the range  $[k, (\max\{\min\{\sqrt{n}, 50\}, \lceil \frac{3}{2}k \rceil\})]$ . This setting could improve the diversity between the clustering results in the ensemble base.

### Working Mechanism

To show the working mechanism of CEHM, four two-dimensional artificial data sets are utilized. The information about these data is shown in Table 3 in Appendix A.5.

Figure 2 and Figure 5 to Figure 7 in Appendix A.5. show some of the process results of CEHM on these data sets. In Figure 2 and Figure 5 to Figure 7, the data points that are not in the hypergraph are shown by Gray dot,  $\mathcal{E}i$  indicates the result of the k-HyperEdge initialization step,  $\mathcal{E}d(1)$  and  $\mathcal{E}d(2)$  are two results of the k-HyperEdge diffusion step,  $\mathcal{E}a(1)$  and  $\mathcal{E}a(2)$  are two results of the k-HyperEdge adjustment step, and  $\mathcal{E}_m$  is the discovered k-HyperEdge Medoids. From Figure 2 and Figure 5 to Figure 7, it can be seen that the operation process conforms to the expectations of each stage.

### Convergence

The convergence of CEHM is analyzed on twenty widely used benchmark data sets. The information about these data sets is shown in Table 1.

For each data set, we generate 30 base clustering result sets and then show the loss of the hyperedge that is sequentially generated during the running of CEHM on each base clustering result set. The hyperedge loss is calculated by Equation (14). Figure 3 shows the results of the convergence analysis on the first 5 data sets. The rest convergence results are shown in Figure 8 in Appendix A.6. From Figure 3 and

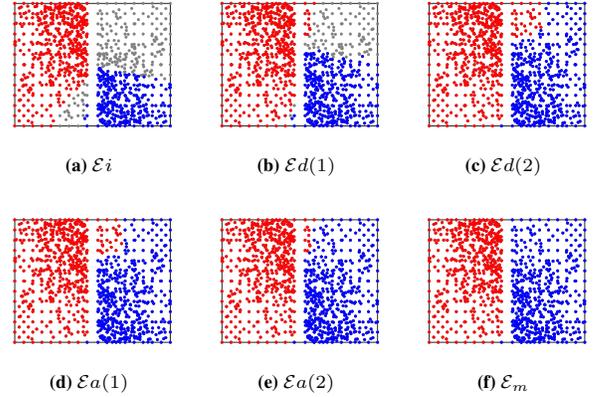


Figure 2: Working mechanism of CEHM on the WingNut data set

Figure 8, it can be seen that the loss of the hyperedge set gradually reduces for each test. The CEHM algorithm can converge quickly. On the test on these data sets, the maximum number of iterations is not greater than 30.

### Ensemble Performance

To test the clustering ensemble performance of CEHM, nine representative clustering ensemble methods are utilized as reference methods, which are U-SENC (Huang et al. 2020), DREC (Zhou, Zheng, and Pan 2019), EC-CMS (Jia et al. 2023), PTGP (Huang, Lai, and Wang 2016), ECPCS-MC (Huang et al. 2021), PTA (Huang, Lai, and Wang 2016), WCT (Iam-On et al. 2011), WTQ (Iam-On et al. 2011) and Voting (Li et al. 2017). The details about the methods and their settings are in the Appendix A.7.

To evaluate the clustering ensemble result of each method, two of the most widely used external indices are utilized, which are Normalized Mutual Information (NMI) (Strehl and Ghosh 2002) and Adjusted Rand Index (ARI) (Hubert and Arabie 1985).

The ensemble performance results are shown in Table 2 and Table 4 (in Appendix A.8). The tables respectively show the average NMI and AIR of the 30 times for each method on each data set. The maximum index value of each comparison is marked in bold type with underlining. The OM means the corresponding method out of memory during handling the data. From the tables, it is easy to see that the CEHM is marked on most of the data sets, which show the effectiveness of CEHM in handling the clustering ensemble problem on these data sets.

We statistically analyze the experimental results that are shown in the Table 2 and Table 4 (in Appendix A.8). The analysis methods and results are included in Appendix A.9.

### Time Cost

The time complexity of CEHM consists of three parts, which correspond to the three steps. The time complexity of the k-HyperEdge initialization step is  $O(n_c^2)$  (Tiwari et al. 2020). The time complexity of the k-HyperEdge diffusion step and

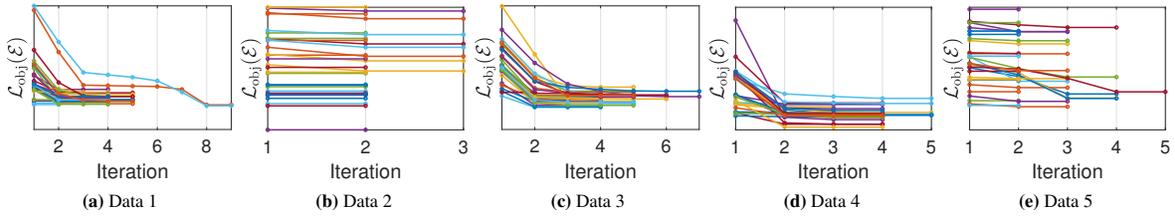


Figure 3: The convergence demonstration of CEHM on the first five data sets

Data	U-SENC	DREC	EC-CMS	PTGP	ECPCS-MC	PTA	WCT	WTQ	Voting	CEHM
1	0.6782	0.7123	0.7123	0.6958	0.6907	0.5706	0.7128	0.7080	0.6309	<b>0.7456</b>
2	0.8961	0.8871	0.8442	0.8833	0.8783	0.8685	0.8947	0.8562	0.8276	<b>0.9039</b>
3	0.7316	0.6290	0.6770	0.6426	0.6960	0.6022	0.6915	0.6788	0.7297	<b>0.7320</b>
4	0.3054	0.2644	0.3033	0.1422	0.2835	0.1179	0.2790	0.2106	0.1797	<b>0.3072</b>
5	0.7373	0.7277	0.7262	0.6916	0.7366	0.6957	0.7135	0.7160	0.6952	<b>0.7420</b>
6	0.6178	0.6048	0.6482	0.5757	0.6493	0.6063	0.6000	0.5924	0.5371	<b>0.7007</b>
7	0.8910	0.8757	0.8572	0.8838	0.8952	0.8873	0.8532	0.8336	0.8109	<b>0.9109</b>
8	0.4278	0.4228	0.4633	0.4269	0.4619	0.4346	0.4045	0.4069	0.3967	<b>0.4674</b>
9	0.6074	0.6315	0.2299	0.5885	0.5570	0.5398	0.5710	0.5841	0.4522	<b>0.6511</b>
10	<b>0.5853</b>	<b>0.5853</b>	<b>0.5853</b>	<b>0.5853</b>	0.5853	<b>0.5853</b>	0.5580	0.5612	0.3762	<b>0.5853</b>
11	0.3850	0.4118	0.3156	0.3210	0.3234	0.3768	0.3605	0.3539	0.3400	<b>0.4718</b>
12	0.6424	0.6573	0.6369	0.6463	0.6457	0.6608	0.6265	0.5987	0.6008	<b>0.6665</b>
13	0.6175	0.6142	0.5348	0.5998	0.6190	0.5864	0.6119	0.5748	0.5621	<b>0.6316</b>
14	0.2035	0.1365	0.1352	0.1044	0.1865	0.1137	0.1166	0.1694	0.1088	<b>0.2391</b>
15	0.6299	0.6479	0.6071	0.6376	0.6510	0.6630	0.6134	0.5642	0.5322	<b>0.6687</b>
16	0.8400	0.8384	OM	0.8378	0.8426	0.8204	0.8411	0.8400	0.8432	<b>0.8444</b>
17	0.1907	0.2196	OM	0.2214	0.1902	0.0925	0.1909	0.2486	0.2272	<b>0.3359</b>
18	0.4931	0.3352	OM	0.4019	0.1002	0.4253	0.4024	0.3353	0.3043	<b>0.5994</b>
19	0.7526	OM	OM	OM	0.7550	OM	0.7273	0.6911	0.6752	<b>0.7567</b>
20	0.5434	OM	OM	OM	0.5603	OM	0.5613	0.4983	0.5025	<b>0.5818</b>

Table 2: The NMI from the compared methods on the twenty data sets

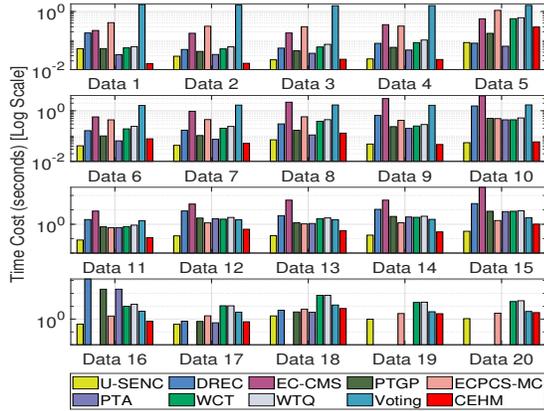


Figure 4: The time cost of the compared methods on the twenty data sets

the k-HyperEdge adjustment are both  $O(tnn_c)$ . Then the total time the k-HyperEdge adjustment is  $O(n_c^2 + 2tnn_c)$ . The time costs of these methods on the twenty data sets are also compared. Figure 4 shows the average running time of each method on the twenty data sets. From Figure 4, it can be seen

that CEHM consumes the least running time except for U-SENC, a method specifically designed for large-scale data with the downsampling techniques. The results illustrate the efficiency of CEHM.

## Conclusion

In this paper, we proposed k-HyperEdge Medoids for clustering ensemble. The k-HyperEdge Medoids were defined as nonoverlapping hyperedges, and each hyperedge in the hypergraph could find a similar hyperedge medoid. We solved the clustering ensemble problem by discovering a set of k-HyperEdge Medoids. The proposed discovery method contains k-HyperEdge initialization, k-HyperEdge diffusion, and k-HyperEdge adjustment. Concretely, a set of initial sub-hyperedges is selected in the k-HyperEdge initialization step, and the quality of the chosen hyperedges is improved by diffusion and adjustment guided by a hyperedge loss function that was reduced by assigning samples to the hyperedge with the highest degree of belonging. The rationality of the solution, the assignment method, and the belonging degree estimation method are illustrated by theoretical analysis. The working mechanism, convergence, effectiveness, and efficiency were illustrated by experimental analysis. In the future, it is interesting to develop novel learning methods to construct the k-HyperEdge Medoids.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 62476160, 62136005, 62306170, 62472268, 62306171), the Science and Technology Major Project of Shanxi (No. 202201020101006), the Special Fund for Science and Technology Innovation Teams of Shanxi Province (No. 202304051001001), and the Funds for Central-Government-Guided Local Science and Technology Development (YDZJSX20231C001).

## References

- Ayad, H.; and Kamel, M. S. 2008. Cumulative Voting Consensus Method for Partitions with Variable Number of Clusters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1): 160–173.
- Ayad, H.; and Kamel, M. S. 2010. On voting-based consensus of cluster ensembles. *Pattern Recognition*, 43(5): 1943–1953.
- Carpineto, C.; and Romano, G. 2012. Consensus clustering based on a new probabilistic rand index with application to subtopic retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12): 2315–2326.
- Fern, X. Z.; and Brodley, C. E. 2004. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the twenty-first international conference on Machine learning*, 36.
- Fred, A. L.; and Jain, A. K. 2005. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6): 835–850.
- Fu, H.; Yue, X.; Liu, W.; and Denooux, T. 2022. Stable Clustering Ensemble Based on Evidence Theory. In *2022 IEEE International Conference on Image Processing (ICIP)*, 2046–2050.
- Guilbert, M.; Vrain, C.; Dao, T.-B.-H.; and de Souto, M. C. P. 2022. Anchored Constrained Clustering Ensemble. In *2022 International Joint Conference on Neural Networks (IJCNN)*, 1–8.
- Hao, Z.; Lu, Z.; Li, G.; Nie, F.; Wang, R.; and Li, X. 2024. Ensemble Clustering With Attentional Representation. *IEEE Transactions on Knowledge and Data Engineering*, 36(2): 581–593.
- Huang, D.; Lai, J.-H.; and Wang, C.-D. 2016. Robust ensemble clustering using probability trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 28(5): 1312–1326.
- Huang, D.; Wang, C.-D.; Peng, H.; Lai, J.; and Kwok, C.-K. 2021. Enhanced ensemble clustering via fast propagation of cluster-wise similarities. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(1): 508–520.
- Huang, D.; Wang, C.-D.; Wu, J.-S.; Lai, J.-H.; and Kwok, C.-K. 2020. Ultra-Scalable Spectral Clustering and Ensemble Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6): 1212–1226.
- Huang, Q.; Gao, R.; and Akhavan, H. 2023. An ensemble hierarchical clustering algorithm based on merits at cluster and partition levels. *Pattern Recognition*, 136: 109255.
- Hubert, L.; and Arabie, P. 1985. Comparing partitions. *Journal of Classification*, 2(1): 193–218.
- Iam-On, N.; Boongoen, T.; Garrett, S.; and Price, C. 2011. A link-based approach to the cluster ensemble problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12): 2396–2409.
- Jia, Y.; Tao, S.; Wang, R.; and Wang, Y. 2023. Ensemble Clustering via Co-Association Matrix Self-Enhancement. *IEEE Transactions on Neural Networks and Learning Systems*, 1–12.
- Li, F.; Qian, Y.; and Wang, J. 2021. GoT: a Growing Tree Model for Clustering Ensemble. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 8349–8356.
- Li, F.; Qian, Y.; Wang, J.; Dang, C.; and Jing, L. 2019. Clustering ensemble based on sample’s stability. *Artificial Intelligence*, 273: 37–55.
- Li, F.; Qian, Y.; Wang, J.; and Liang, J. 2017. Multigranulation information fusion: A Dempster-Shafer evidence theory-based clustering ensemble method. *Information Sciences*, 378: 389–409.
- Li, F.; Wang, J.; Zhang, L.; Qian, Y.; Jin, S.; Yan, T.; and Du, L. 2024. k-HyperEdge Medoids for Clustering Ensemble. arXiv:2412.08289.
- Liu, H.; Wu, J.; Liu, T.; Tao, D.; and Fu, Y. 2017. Spectral ensemble clustering via weighted k-means: theoretical and practical evidence. *IEEE Transactions on Knowledge and Data Engineering*, 29(5): 1129–1143.
- Strehl, A.; and Ghosh, J. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3(Dec): 583–617.
- Tiwari, M.; Zhang, M. J.; Mayclin, J.; Thrun, S.; Piech, C.; and Shomorony, I. 2020. BanditPAM: almost linear time k-medoids clustering via multi-armed bandits. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Curran Associates Inc.
- Vega-Pons, S.; and Ruiz-Shulcloper, J. 2011. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03): 337–372.
- Zhou, J.; Zheng, H.; and Pan, L. 2019. Ensemble clustering based on dense representation. *Neurocomputing*, 357(SEP.10): 66–76.
- Zhou, P.; Du, L.; and Li, X. 2020. Self-paced Consensus Clustering with Bipartite Graph. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 2133–2139.
- Zhou, P.; Wang, X.; Du, L.; and Li, X. 2022. Clustering ensemble via structured hypergraph learning. *Information Fusion*, 78: 171–179.
- Zhou, Z.-H.; and Tang, W. 2006. Clusterer ensemble. *Knowledge-Based Systems*, 19(1): 77–83.