# TECM: Transfer Learning-based Evidential C-Means Clustering

Lianmeng Jiao\*, Feng Wang, Zhun-ga Liu, Quan Pan

School of Automation, Northwestern Polytechnical University, Xi'an 710072, P.R. China

#### Abstract

As a representative evidential clustering algorithm, evidential c-means (ECM) provides a deeper insight into the data by allowing an object to belong not only to a single class, but also to any subset of a collection of classes, which generalizes the hard, fuzzy, possibilistic, and rough partitions. However, compared with other partition-based algorithms, ECM must estimate numerous additional parameters, and thus insufficient or contaminated data will have a greater influence on its clustering performance. To solve this problem, in this study, a transfer learning-based ECM (TECM) algorithm is proposed by introducing the strategy of transfer learning into the process of evidential clustering. The TECM objective function is constructed by integrating the knowledge learned from the source domain with the data in the target domain to cluster the target data. Subsequently, an alternate optimization scheme is developed to solve the constraint objective function of the TECM algorithm. The proposed TECM algorithm is applicable to cases where the source and target domains have the same or different numbers of clusters. A series of experiments were conducted on both synthetic and real datasets, and the experimental results demonstrated the effectiveness of the proposed TECM algorithm compared to ECM and other representative multitask or transfer-clustering algorithms.

<sup>\*</sup>Corresponding author at: School of Automation, Northwestern Polytechnical University, Xi'an 710072, P.R. China. Tel.: +86 29 88431306; Fax: +86 29 88431306.

 $<sup>\</sup>label{eq:email} Email \ addresses: \ \texttt{jiaolianmeng@nwpu.edu.cn} \ (Lianmeng \ Jiao^*),$ 

fengwang@mail.nwpu.edu.cn (Feng Wang), liuzhunga@nwpu.edu.cn (Zhun-ga Liu), quanpan@nwpu.edu.cn (Quan Pan)

*Keywords:* Evidential clustering, transfer learning, belief function theory, credal partition

## 1. Introduction

Clustering is a process of dividing a set of objects into groups of similar members. Classical clustering algorithms can be roughly divided into the following categories: partition-based methods [1, 2], hierarchy-based methods [3, 4], density-based methods [5, 6], grid-based methods [7, 8], and graph-based methods [9, 10]. Among these, partition-based methods are the most popular because they are universally applicable to numerous real-world clustering tasks. Although classical partition-based clustering algorithms such as c-means only provide hard partitions, a large number of variants have been proposed by incorporating fuzzy sets [11], possibility theory [12], and rough sets [13] to characterize the uncertainty of cluster membership. Recently, evidential clustering [14, 15, 16, 17, 18, 19, 20] has been developed as a promising framework that uses the Dempster-Shafer belief function theory [21, 22] as an uncertainty model. It generalizes the above-mentioned hard, fuzzy, possibilistic, and rough partitions, and thus, has been widely used in numerous application scenarios, including tumor segmentation [23, 24], community detection [25, 26], and items of interest recommendation [27, 28].

Evidential c-means (ECM) [14], as a representative evidential clustering algorithm, can assign objects not only to a single class, but also to any subset of a collection of classes by credal partition [29]. Therefore, ECM can provide a deeper insight into the data for situations where the data distribution is complex. However, this additional flexibility comes at the expense of algorithm complexity, that is, the ECM algorithm must estimate numerous additional parameters compared to the original c-means or fuzzy c-means (FCM) algorithms; therefore, insufficient or contaminated data will have a greater influence on its clustering performance. Noise is unavoidable in practical clustering. In addition, collecting sufficient data is difficult in certain practical applications. Such shortcomings severely restrict the applicability of ECM. One feasible method to address this problem is transfer learning [30, 31].

Transfer learning assumes that the data in the current scene (called the target domain) are inadequate for the learning task, whereas other helpful information is available from a relevant scene (called the source domain). Currently, transfer learning is mainly integrated with the following machine learning tasks [32]: classification [33, 34], regression [35, 36], feature extraction [37, 38], and clustering [39, 40]. The first three tasks have been studied extensively; however, research on integrating transfer learning with clustering remains inadequate, although it has wide application in reality. In the past decade, research on transfer clustering has gradually emerged. Studies on transfer clustering can be broadly classified into the following four categories based on the transfer strategy [41]: (1) the instance-based method [42], where a portion of the source data is selected and transferred to the target domain for clustering target data; (2) the feature-representation-based method [43, 44], whose intuitive idea is to generate a superior feature representation for the target data based on the knowledge learned from the source data; (3) the parameterbased method [39, 40, 45, 46, 47, 48], which assumes that two domains share similar parameters such as cluster centers; and (4) the relational knowledgebased method [49, 50], which builds a mapping of the relational knowledge between two domains. Among these, parameter-based transfer clustering is the focus of the current research because it can fully exploit the inter-domain similarity and is more suitable for integration with popular partition-based clustering algorithms.

In this paper<sup>1</sup>, transfer learning is applied to ECM to improve the clustering performance when the target data are insufficient or contaminated. A parameter-based transfer strategy is adopted to develop a transfer learning-

<sup>&</sup>lt;sup>1</sup>A preliminary version of this paper have been presented in BELIEF 2021 Conference [51]. This paper extends our short conference paper by adding a comprehensive review of related work, a deep analysis of the challenges, a more detailed description of the proposed algorithm with an analysis of its generality, convergence, and complexity, and additional datasets and comparison algorithms to illustrate the advantages of the proposed algorithm.

based ECM (TECM) algorithm, that is, the parameters induced from the source data, namely, barycenters, are appropriately used to improve the clustering performance of the target data. The main contributions of this study are summarized as follows.

(1) A novel TECM objective function is constructed by introducing the barycenters learned from the source data as knowledge to guide the clustering of the target data.

(2) An alternate optimization scheme is developed to solve the constrained TECM objective function, which derives closed-form solutions by alternately updating the parameters.

(3) The proposed TECM is applicable to any transfer-clustering scenario where the source and target domains have the same or different numbers of clusters.

A series of experiments on both synthetic and real datasets are designed to demonstrate the effectiveness of the TECM algorithm proposed in this study. In the experiments on synthetic datasets, considering three situations (the number of clusters in the source domain is equal to, greater than, or less than the number of clusters in the target domain) and two application scenarios (target data are insufficient or contaminated), six groups of synthetic datasets are designed to illustrate the TECM clustering performance. Furthermore, two synthetic ambiguous datasets are designed to demonstrate the superiority of TECM in characterizing the uncertainty of cluster membership. In the experiments on real datasets, the segmentation results of texture images are used to illustrate the TECM clustering performance compared to ECM and other representative multitask or transfer-clustering algorithms. The experimental results on the synthetic and real datasets demonstrate that the proposed TECM achieves superior clustering performance compared to the other methods on the three popular validity indices.

The remainder of this paper is organized as follows. Section 2 comprehensively reviews of the related work. The ECM algorithm is summarized and analyzed in Section 3. A detailed description of the proposed TECM is presented in Section 4. In Section 5, we evaluate the proposed TECM using both synthetic and real datasets. Section 6 concludes the paper and outlines the directions for future research.

# 2. Related work

In the past decade, transfer-learning techniques have been integrated with classical clustering algorithms. Based on their transfer strategy, the existing transfer-clustering algorithms can be categorized as follows.

Instance-based method. In [42], the unlabeled heterogeneous data from different domains are first projected into a common subspace, and then other informative source data are selected and transferred to the target domain to improve the clustering performance of the target data using an algorithm called DicTrans. However, because of the differences in data distribution between the two domains, it is difficult to determine how to select the source data and the amount of source data to be transferred.

Feature-representation-based method. In [43], based on co-occurrence data, a co-transfer-clustering algorithm was presented to simultaneously address multiple individual clustering tasks. A collective nonnegative matrix factorization algorithm is used to obtain a common subspace for all domains. Then, data in different domains are mapped into this subspace and the data from the different domains are clustered simultaneously using a symmetric nonnegative matrix factorization algorithm. In [44], a transfer-clustering algorithm called shared hidden space transfer fuzzy c-means was proposed. In this algorithm, source data and target data are mapped into a common subspace by attempting to identify a projected matrix with the strategy of maximum mean discrepancy; then, all data in the common subspace are clustered simultaneously.

Parameter-based method. In [40], a representative parameter-based transferclustering algorithm called transfer fuzzy c-means (TFCM) clustering was proposed using the transfer-learning strategy. The cluster centers obtained from the source domain via classical FCM clustering are used to construct the corresponding objective function to promote the clustering performance of the target domain. In [46, 47, 48], as in [40], the cluster centers obtained from the source domain are used as transfer knowledge to guide the clustering of the target domain. In [45], a novel transfer-clustering algorithm called transfer distributed EM clustering (TDEM) was proposed, where each node is considered as both the source and target domains; these nodes can learn from each other to complete the clustering task in distributed peer-to-peer (P2P) networks. In [39], a transfer maximum entropy clustering algorithm was proposed and the corresponding objective function was constructed using the cluster centers and membership matrix obtained from the source data. However, this algorithm is only applicable to situations where the source and target domains have the same number of clusters.

Relational knowledge-based method. In [49], transfer spectral clustering (TSC), a representative relational knowledge-based transfer-clustering algorithm, was proposed. Based on the assumption that embedded feature information can connect two clustering tasks, TSC can simultaneously improve the clustering performance in two domains. Furthermore, co-clustering is employed to control the knowledge transfer between the two tasks. In [50], a novel transfer-clustering algorithm called topic constraint transfer clustering was proposed. First, several clusters are obtained from the source data using an efficient clustering algorithm. Subsequently, several topics are extracted from each cluster. Then, the similarities between the target data. Finally, a semi-supervised clustering algorithm is applied to the target data.

Apart from the above four categories of transfer-clustering algorithms, our task is also related to multitask clustering, which addresses multiple clustering tasks simultaneously and uses mutual cooperation to improve the performance of all the clustering tasks. Representative multitask clustering algorithms include learning the shared subspace for multi-task clustering (LSSMTC) and combining k-means (CombKM) [53]. LSSMTC aims to learn a subspace shared by all tasks, through which the knowledge of the tasks can be transferred to each other; CombKM clusters all data from different tasks simultaneously using c-means.

Among these transfer or multitask clustering methods, parameter-based transfer clustering is the focus of the current research because it is more suitable for integration with popular partition-based clustering algorithms. In partitionbased clustering algorithms, the cluster membership of each object depends on its distance to the cluster centers; therefore, the parameter "cluster centers" is the most reliable and intuitive transfer knowledge. In addition, parameter-based transfer clustering typically obtains acceptable transfer performance because it can fully utilize the inter-domain similarity. Therefore, the parameter-based transfer strategy is adopted in this study to develop an ECM transfer-clustering version.

# 3. ECM: Basics and challenges

Belief function theory [21, 22] is a formal framework for approximate reasoning. In this theory, the set  $\Omega = \{w_1, ..., w_c\}$  containing all possible results of a discrimination problem is called the frame of discernment. A mass function m is used to express uncertain information, which is a mapping function from  $2^{\Omega}$  to [0, 1] that satisfies the following equations:

$$m(\emptyset) = 0, \ \sum_{A \in 2^{\Omega}} m(A) = 1.$$
(1)

Subsets A are called focal sets if  $A \in 2^{\Omega}$  satisfies m(A) > 0. Based on the type of information represented, the mass function m is classified into the following categories:

• *Bayesian*: The focal sets of this type of mass function are singletons. In this case, the mass function becomes the classical probability distribution function.

• Certain: This type of mass function has only one focal set, which can be denoted by  $m_A$ , where A is its unique focal set. This mass function certainly indicates that the real result is in set A.

• Vacuous: This type of mass function has only one focal set,  $\Omega$ , which can be denoted as  $m_{\Omega}$ . This mass function indicates that the actual result is unknown.

In [29], the notion of a credal partition was proposed to represent the class membership uncertainty of each object  $x_i$  by a mass function  $m_i$ , which can model different situations from completely unknown to total certainty regarding the class membership of the object.

**Example 1.** Consider an example of n = 5 objects belonging to c = 2 classes with evidential membership. The mass functions of these objects are listed in Table 1, which illustrates all possible situations from complete unknown to total certainty. The case of object 1 corresponds to the situation of total certainty ( $m_1$ is certain). Object 2 corresponds to the situation of completely unknown ( $m_2$  is vacuous). Object 3 corresponds to the situation of probabilistic uncertainty ( $m_3$ is Bayesian). Object 4 is a general example of evidential membership. Mass  $m_5(\emptyset) = 1$  indicates that the class of object 5 is not in  $\Omega$ .

A	$m_1(A)$	$m_2(A)$	$m_3(A)$	$m_4(A)$	$m_5(A)$
Ø	0	0	0	0.1	1
$w_1$	1	0	0.4	0.4	0
$w_2$	0	0	0.6	0.3	0
Ω	0	1	0	0.2	0

Table 1: Illustration of credal partition

The ECM algorithm [14] is a representative clustering algorithm with credal partition, which is an extended version of FCM in the framework of belief function theory. The mass  $m_{ij} = m_i(A_j)$  associated with object  $x_i$  with  $A_j$  $(A_j \neq \emptyset, A_j \subseteq \Omega)$  depends on the distance  $d_{ij} = ||x_i - \bar{v}_j||^2$  between object  $x_i$ and barycenter  $\bar{v}_j$  of  $A_j$ , which is greater when  $d_{ij}$  is smaller. Barycenter  $\bar{v}_j$  is defined as

$$\overline{v}_j = \frac{1}{c_j} \sum_{k=1}^c s_{kj} v_k, \quad \text{s. t.} \quad s_{kj} = \begin{cases} 1 & \text{if } w_k \in A_j \\ 0 & \text{else,} \end{cases}$$
(2)

where  $c_j$  is the cardinality of the focal set  $A_j$  and  $v_k$  is the center of the singleton  $w_k$ .

For *n* objects with *p* dimensions, the ECM objective function with respect to the credal partition  $M = \{m_1, ..., m_n\} \in \mathbb{R}^{n \times 2^c}$  and cluster centers  $V = \{v_1, ..., v_c\} \in \mathbb{R}^{c \times p}$  is defined as

$$\min J_{ECM}(M,V) = \sum_{i=1}^{n} \sum_{\{j/A_j \subseteq \Omega, A_j \neq \emptyset\}} c_j^{\alpha} m_{ij}^{\beta} ||x_i - \overline{v}_j||^2 + \sum_{i=1}^{n} \delta^2 m_{i\emptyset}^{\beta},$$
  
s. t. 
$$\sum_{\{j/A_j \subseteq \Omega, A_j \neq \emptyset\}} m_{ij} + m_{i\emptyset} = 1, \quad \forall i = 1, ..., n,$$
(3)

where  $m_{i\emptyset}$  represents  $m_i(\emptyset)$  and  $c_j$  represents the cardinality of the focal set  $A_j$ . Parameters  $\alpha$  and  $\beta$  control the penalization degree of high cardinality and credibility of the partition, respectively, and  $\delta$  controls the number of outliers.

Based on the objective function given in Eq. (3), it can be observed that the ECM must estimate a considerably greater number of parameters  $(n * 2^c)$  to obtain a credal partition compared to other classical partition-based algorithms (n \* c for FCM and n for c-means). Therefore, insufficient or contaminated data could have a greater influence on ECM compared to other partition-based algorithms. Here, we provide an example to illustrate this issue.

**Example 2.** Two synthetic datasets,  $Synt_1$  and  $Synt_2$ , are used to illustrate the ECM clustering performance for application scenarios where the data are insufficient and contaminated, respectively. Let  $\mu_i$  and  $\Sigma_i$  represent the mean vector and covariance matrix of the ith cluster in each dataset, respectively. Dataset  $Synt_1$  simulates a scenario where the data are insufficient. Dataset  $Synt_1$  is generated with  $\mu_1 = [0,0]$ ,  $\Sigma_1 = [1,0;0,1]$ ,  $\mu_2 = [0,3]$ ,  $\Sigma_2 = [1,0;0,1]$ ,  $\mu_3 = [3,0]$ ,  $\Sigma_3 = [1,0;0,1]$ ,  $\mu_4 = [3,3]$ , and  $\Sigma_4 = [1,0;0,1]$ , where each cluster consists of only five objects. The data distributions used to generate the dataset  $Synt_2$  are the same as those for  $Synt_1$ ; however, each cluster in  $Synt_2$ consists of 20 objects. In addition, zero-mean Gaussian noise with a standard deviation of 0.3 is added to  $Synt_2$  to simulate the scenario where the data are contaminated. The clustering results for datasets  $Synt_1$  and  $Synt_2$  using ECM are displayed in Fig. 1 and Fig. 2, respectively. It can be observed from the clustering results that the real cluster profiles were not recovered precisely, and the ambiguous objects at the boundaries were not properly recognized.



Figure 1: Clustering results for dataset  $Synt_1$  using ECM.



Figure 2: Clustering results for dataset  $Synt_2$  using ECM.

The above example confirms that the ECM clustering performance is not ideal for situations where the data are insufficient or contaminated. Therefore, it is necessary to improve the ECM clustering performance in these situations by exploiting sufficient high-quality data from a relevant scene.

## 4. TECM

In this section, a TECM clustering algorithm is developed to improve the clustering performance when the target data are insufficient or contaminated. In Section 4.1, the basic principle of TECM is introduced. In Section 4.2, a novel TECM objective function is presented. In Section 4.3, a procedure for solving the TECM objective function is developed. In Section 4.4, we provide the algorithm pseudocode and analyze the generality, convergence, and complexity of the proposed algorithm.

# 4.1. Basic principle of TECM

In transfer learning, as indicated in Fig. 3, sampled data and learned knowledge are typically the two types of information available from the source data. The sampled data are not appropriate for the direct clustering of the target data because of the differences in the data distribution between these two domains. In addition, source data are at times inaccessible because of privacy protection in certain practical scenarios. Compared with sampled data, learned knowledge such as feature representations, parameters, and relationships is typically considered more insightful and noise-resistant.

In this study, the TECM clustering algorithm was developed using the learned knowledge from sufficient data of acceptable quality in a relevant scene. The basic principle of TECM is illustrated in Fig. 4. It can be observed that ECM is first applied to the source data to learn the knowledge, that is, barycenters of the clusters, which concisely represent the data distribution. These barycenters are then used to guide the clustering of the target data. Following this principle, we study how to construct the TECM objective function by appropriately using the knowledge learned from the source data and how to develop an efficient optimization scheme to solve the objective function.



Figure 3: Available information obtained from source data for clustering of target data.



Figure 4: Basic principle of TECM.

# 4.2. TECM Objective function

In this section, the TECM objective function is constructed by introducing the learned barycenters into the ECM objective function. The TECM objective function with regard to the credal partition M, association matrix R, and cluster centers  $\boldsymbol{V}$  is defined as

$$\min J_{TECM}(M, R, V) = \sum_{i=1}^{n} \sum_{\{j/A_j \subseteq \Omega_t, A_j \neq \emptyset\}} c_j^{\alpha} m_{ij}^{\beta} ||x_i - \bar{v}_j||^2 + \lambda \sum_{\{k/A_k \subseteq \Omega_s, A_k \neq \emptyset\}} \sum_{\{j/A_j \subseteq \Omega_t, A_j \neq \emptyset\}} c_j^{\alpha} r_{kj}^{\gamma} ||\tilde{v}_k - \bar{v}_j||^2 + \sum_{i=1}^{n} \delta^2 m_{i\emptyset}^{\beta},$$

$$s.t. \sum_{\{j/A_j \subseteq \Omega_t, A_j \neq \emptyset\}} m_{ij} + m_{i\emptyset} = 1, \quad \forall i = 1, ..., n,$$

$$\sum_{\{j/A_j \subseteq \Omega_t, A_j \neq \emptyset\}} r_{kj} = 1, \quad \forall k/A_k \subseteq \Omega_s, A_k \neq \emptyset,$$
(4)

where  $\tilde{v}_k$  is the k-th barycenter obtained from the source data and element  $r_{kj}$  of R denotes the association between the j-th barycenter learned from the target data and k-th barycenter learned from the source data. The sets  $\Omega_t$  and  $\Omega_s$  denote the frames of the discernment of the target and source domains, respectively. Parameter  $\gamma$  controls the credibility of the association and  $\lambda$  is the balance coefficient of transfer learning. The definitions of the other parameters  $\alpha$ ,  $\beta$ , and  $\delta$  are the same as those in ECM.

The characteristics of the objective function defined in Eq. (4) are as follows.

1) The first and third terms are inherited from the ECM objective function, which is used to generate the credal partition of the target data by minimizing the global inner-cluster distance by considering the membership degrees between the objects and barycenters.

2) The second term introduces knowledge learned from the source data. The barycenters generated from the source data provide concise yet rich class distribution information of the relevant source domain. This term requires that the barycenters of the target data be as close as possible to the barycenters learned from the source data as a whole by considering their potential associations.

3) The parameter  $\lambda$  is a balance coefficient used to control the transfer degree of the learned knowledge from the source data. If the value of  $\lambda$  is zero, TECM becomes the original ECM; if the value of  $\lambda$  is relatively large, a negative transfer could occur. Therefore, an appropriate value of  $\lambda$  must be selected. The choice of  $\lambda$  depends on the specific application; the grid search strategy [54] can be used to determine an appropriate value. A series of experiments [40] proved that as the value of  $\lambda$  increases, the clustering performance always improves first and then worsens. Based on this, we can first set a large search range with a large step size, and then select the part with superior clustering performance and reduce the step size. These steps are repeated to obtain an appropriate value of  $\lambda$  with acceptable clustering performance. Parameter  $\gamma$  can be set to 2, similar to that in [40]. The values of the other parameters  $\alpha$ ,  $\beta$ , and  $\delta$  can be set to the same values as in ECM [14].

4) The objective function is defined for the general case, where the frames of discernment  $\Omega_s$  and  $\Omega_t$  can be different. By minimizing the objective function, we can determine the optimal pairwise associations of the barycenters (represented by the association matrix R) between the source and target data. This property significantly expands the scope of application of TECM.

# 4.3. Optimization procedure

To minimize  $J_{TECM}$ , an alternate optimization scheme is used in this study, that is, optimizing one variable by setting the other variables to fixed values. From Eq. (4), it can be observed that the objective function  $J_{TECM}$  is unconstrained to V, yet is constrained to M and R. To solve the constrained object function for M and R, n Lagrange multipliers  $\lambda_i$  and k (k is the number of focal sets in  $\Omega_s$ ) Lagrange multipliers  $\eta_k$  are introduced to formulate the following Lagrangian objective function  $J_L$ :

$$J_{L} = J_{TECM} + \sum_{i=1}^{n} \lambda_{i} \left[ 1 - \left( \sum_{\{j/A_{j} \subseteq \Omega_{t}, A_{j \neq \emptyset}\}} m_{ij} + m_{i\emptyset} \right) \right] + \sum_{\{k/A_{k} \subseteq \Omega_{s}, A_{k} \neq \emptyset\}} \eta_{k} \left( 1 - \sum_{\{j/A_{j} \subseteq \Omega_{t}, A_{j \neq \emptyset}\}} r_{kj} \right).$$

$$(5)$$

4.3.1. Optimization with respect to credal partition M

First, let V and R be fixed. Taking the partial derivatives of  $J_L$  to  $m_{ij}$ ,  $m_{i\emptyset}$ , and  $\lambda_i$ , and setting them to 0, we obtain

$$\frac{\partial J_L}{\partial m_{ij}} = \beta c_j^{\alpha} m_{ij}^{\beta-1} d_{ij}^2 - \lambda_i = 0, \qquad (6)$$

$$\frac{\partial J_L}{\partial m_{i\emptyset}} = \beta \delta^2 m_{i\emptyset}^{\beta-1} - \lambda_i = 0, \tag{7}$$

$$\frac{\partial J_L}{\partial \lambda_i} = \sum_{\{j/A_j \subseteq \Omega_t, A_j \neq \emptyset\}} m_{ij} + m_{i\emptyset} - 1 = 0.$$
(8)

From Eq. (6), we obtain

$$m_{ij} = \left(\frac{\lambda_i}{\beta}\right)^{\frac{1}{\beta-1}} \left(\frac{1}{c_j^{\alpha} ||x_i - \bar{v}_j||^2}\right)^{\frac{1}{\beta-1}},\tag{9}$$

and from Eq. (7), we obtain

$$m_{i\emptyset} = \left(\frac{\lambda_i}{\beta}\right)^{\frac{1}{\beta-1}} \left(\frac{1}{\delta^2}\right)^{\frac{1}{\beta-1}}.$$
(10)

Using Eq. (8)–Eq. (10), we obtain

$$\left(\frac{\lambda_i}{\beta}\right)^{\frac{1}{\beta-1}} = \left(\sum_j \frac{1}{(c_j^{\alpha}||x_i - \bar{v}_j||^2)^{\frac{1}{\beta-1}}} + \frac{1}{\delta^{\frac{2}{\beta-1}}}\right)^{-1}.$$
 (11)

Substituting Eq. (11) into Eq. (9), we obtain the updated equation for M:

$$m_{ij} = \frac{\left(c_j^{\alpha}||x_i - \bar{v}_j||^2\right)^{\frac{-1}{\beta - 1}}}{\sum_{A_t \neq \emptyset} \left(c_t^{\alpha}||x_i - \bar{v}_t||^2\right)^{\frac{-1}{\beta - 1}} + \delta^{\frac{-2}{\beta - 1}}},$$

$$\forall i = 1, ..., n, \quad \forall j/A_j \subseteq \Omega_t, A_j \neq \emptyset,$$
(12)

and

$$m_{i\emptyset} = 1 - \sum_{j/A_j \subseteq \Omega_t, A_j \neq \emptyset} m_{ij}, \quad \forall i = 1, ..., n.$$
(13)

# 4.3.2. Optimization with respect to association matrix R

Next, we assume that V and M are fixed. Taking the partial derivatives of  $J_L$  to  $r_{kj}$  and  $\eta_k$  and setting them to 0, we obtain

$$\frac{\partial J_L}{\partial r_{kj}} = \lambda c_j^{\alpha} \gamma r_{kj}^{\gamma-1} (\tilde{v}_k - v_j)^2 - \eta_k = 0, \qquad (14)$$

$$\frac{\partial J_L}{\partial \eta_k} = \sum_{\{j/A_j \subseteq \Omega_t, A_j \neq \emptyset\}} r_{kj} - 1 = 0.$$
(15)

From Eq. (14), we obtain

$$r_{kj} = \left(\frac{\eta_k}{\lambda_{\gamma}}\right)^{\frac{1}{\gamma-1}} \left(\frac{1}{c_j^{\alpha} ||\tilde{v}_k - \bar{v}_j||^2}\right)^{\frac{1}{\gamma-1}}.$$
(16)

Using Eq. (15) and Eq. (16), we obtain

$$\left[\frac{\eta_k}{\lambda\gamma}\right]^{\frac{1}{\gamma-1}} = \left(\sum_j \left[\frac{1}{c_j^{\alpha}(\widetilde{v}_k - \overline{v}_j)^2}\right]^{\frac{1}{\gamma-1}}\right)^{-1}.$$
(17)

Substituting Eq. (17) in Eq. (16), we obtain the updated equation for R:

$$r_{kj} = \frac{(c_j^{\alpha} || \tilde{v}_k - \bar{v}_j ||^2)^{\frac{-1}{\gamma - 1}}}{\sum_{A_t \neq \emptyset} (c_t^{\alpha} || \tilde{v}_k - \bar{v}_t ||^2)^{\frac{-1}{\gamma - 1}}},$$

$$\forall k/A_k \subseteq \Omega_s, A_k \neq \emptyset, \ \forall j/A_j \subseteq \Omega_t, A_j \neq \emptyset.$$
(18)

# 4.3.3. Optimization with respect to cluster centers V

Taking the partial derivative of  $J_{TECM}$  to V and setting it to 0, we obtain

$$\frac{\partial J_{TECM}}{\partial v_l} = -2 \sum_{i=1}^n \sum_{A_{t,j} \neq \emptyset} c_j^{\alpha - 1} m_{ij}^\beta s_{lj} (x_i - \overline{v}_j)$$

$$- 2\lambda \sum_{A_{s,k} \neq \emptyset} \sum_{A_{t,j} \neq \emptyset} c_j^{\alpha - 1} r_{kj}^\gamma s_{lj} (\widetilde{v}_k - \overline{v}_j)$$

$$= -2 \sum_{i=1}^n \sum_{A_{t,j} \neq \emptyset} c_j^{\alpha - 1} m_{ij}^\beta s_{lj} (x_i - \frac{1}{c_j} \sum_z s_{zj} v_z)$$

$$- 2\lambda \sum_{A_{s,k} \neq \emptyset} \sum_{A_{t,j} \neq \emptyset} c_j^{\alpha - 1} r_{kj}^\gamma s_{lj} (\widetilde{v}_k - \frac{1}{c_j} \sum_z s_{zj} v_z) = 0.$$
(19)

This is equivalent to the following equation:

$$\sum_{i} x_{i} \sum_{A_{t,j} \neq \emptyset} c_{j}^{\alpha - 1} m_{ij}^{\beta} s_{lj} + \lambda \sum_{A_{s,k} \neq \emptyset} \widetilde{v}_{k} \sum_{A_{t,j} \neq \emptyset} c_{j}^{\alpha - 1} r_{kj}^{\gamma} s_{lj}$$

$$= \sum_{z} v_{z} \sum_{i} \sum_{A_{t,j} \neq \emptyset} c_{j}^{\alpha - 2} m_{ij}^{\beta} s_{lj} s_{zj} + \lambda \sum_{z} v_{z} \sum_{A_{s,k} \neq \emptyset} \sum_{A_{t,j} \neq \emptyset} c_{j}^{\alpha - 2} r_{kj}^{\gamma} s_{lj} s_{zj}.$$
(20)

Let  $B_1 \in R^{c \times p}$ ,  $B_2 \in R^{c \times p}$ ,  $H_1 \in R^{c \times c}$  and  $H_2 \in R^{c \times c}$  be matrices defined as

$$B_{1_{lq}} = \sum_{i=1}^{n} x_{iq} \sum_{A_{t,j} \neq \emptyset} c_j^{\alpha - 1} m_{ij}^{\beta} s_{lj} = \sum_{i=1}^{n} x_{iq} \sum_{w_l \in A_{t,j}} c_j^{\alpha - 1} m_{ij}^{\beta},$$

$$l = 1, ..., c, \quad q = 1, ..., p,$$
(21)

$$B_{2_{lq}} = \sum_{A_{s,k} \neq \emptyset} \widetilde{v}_{kq} \sum_{A_{t,j} \neq \emptyset} c_j^{\alpha - 1} r_{kj}^{\gamma} s_{lj} = \sum_{A_{s,k} \neq \emptyset} \widetilde{v}_{kq} \sum_{w_l \in A_{t,j}} c_j^{\alpha - 1} r_{kj}^{\gamma},$$

$$l = 1, \dots, c, \quad q = 1, \dots, p,$$
(22)

$$H_{1_{lz}} = \sum_{i} \sum_{A_{t,j} \neq \emptyset} c_j^{\alpha - 2} m_{ij}^{\beta} s_{lj} s_{zj} = \sum_{i} \sum_{A_{t,j} \supseteq \{w_z, w_l\}} c_j^{\alpha - 2} m_{ij}^{\beta},$$

$$l = 1, ..., c, \quad z = 1, ..., c,$$
(23)

$$H_{2_{lz}} = \sum_{A_{s,k} \neq \emptyset} \sum_{A_{t,j} \neq \emptyset} c_j^{\alpha - 2} r_{kj}^{\gamma} s_{lj} s_{zj} = \sum_{A_{s,k} \neq \emptyset} \sum_{A_{t,j} \supseteq \{w_z, w_l\}} c_j^{\alpha - 2} r_{kj}^{\gamma},$$

$$l = 1, ..., c, \quad z = 1, ..., c,$$
(24)

where c denotes the number of clusters in the target domain. Using the above definitions, V is obtained by solving the following linear equation:

$$B_1 + \lambda B_2 = (H_1 + \lambda H_2)V. \tag{25}$$

# 4.4. Summary and analysis

With the above optimization procedure, the clustering results of the objects can be finally obtained by the evidential membership  $m_{ij}$ ,  $i = 1, ..., n, \forall j/A_j \subseteq \Omega_t$ ,  $A_j \neq \emptyset$ , using Eqs. (12) and (13). The computed evidential membership Mcreates a credal partition of the objects. The credal partition provides a general clustering framework, which can be reduced to a traditional hard partition, fuzzy partition [55, 56], possibilistic partition [48, 57], and rough partition [58, 59]. A summary of TECM is presented in Algorithm 1.

Generality analysis: The proposed TECM can be reduced to the original ECM when  $\lambda = 0$  and to TFCM [40] when all the evidential components are constrained to singletons. The proposed TECM provides a fine framework for integrating the advantages of evidential clustering and transfer learning.

Algorithm 1 Transfer learning-based evidential c-means (TECM)

**Require:** objects in target domain  $\{x_1, ..., x_n\}$ , barycenters learned from source domain  $\tilde{v}_k, \forall A_k \subseteq \Omega_s, A_k \neq \emptyset$ , number of clusters c, weighting exponents  $\alpha \ge 0, \beta > 1, \gamma > 1$ , distance to the empty set  $\delta > 0$ , termination threshold  $\varepsilon$ , and balance coefficient of transfer learning  $\lambda$ .

**Ensure:** credal partition M.

- 1: initialize cluster centers V(0).
- 2:  $t \leftarrow 0; J_{TECM}(0) \leftarrow +\infty.$

3: repeat

- 4:  $t \leftarrow t + 1;$
- 5: compute credal partition M(t) using Eqs. (12) and (13);
- 6: compute cluster barycenter association matrix R(t) using Eq. (18);
- 7: compute  $B_1$ ,  $B_2$ ,  $H_1$ , and  $H_2$  using Eqs. (21)-(24);
- 8: compute cluster centers V(t) using Eq. (25);
- 9: compute objective function  $J_{TECM}(t)$  using Eq. (4);

10: **until**  $(|J_{TECM}(t) - J_{TECM}(t-1)| < \varepsilon).$ 

**Convergence analysis**: The convergence of TECM can be proved by Zangwill's convergence theorem [60] in a similar manner to [61]. It should be noted that although only a local optimal solution can be obtained for the TECM nonconvex optimization problem, similar to other FCM-like algorithms, this local optimal solution is sufficiently effective in the majority of cases.

**Complexity analysis:** Similar to ECM, the computational complexity of TECM is  $O(2^c * n)$ . However, in the majority of cases, the assignment of objects to focal sets with high cardinality is less interpretable. Therefore, in real-world scenarios, complexity can be reduced from  $O(2^c * n)$  to  $O(c^2 * n)$  by constraining the focal sets containing at most two clusters.

**Remark 1.** A similar study was recently presented in [52], which developed an evidential clustering algorithm based on transfer learning. Compared to this study, our proposed TECM algorithm has several advantages. First, the knowledge learned from the source data is represented by the cluster centers in [52], whereas in our proposed algorithm, it is represented by the barycenters, which can provide richer class distribution information of the source data. Furthermore, the clustering algorithm in [52] is only applicable to situations where the source and target domains have the same number of clusters, whereas our proposed TECM is also applicable to situations where the number of clusters is different, which has a wider application scope.

#### 5. Experiments

In this section, the TECM proposed in this study is evaluated on both synthetic and real datasets. First, the comparison algorithms and indices for the performance evaluation are introduced in Section 5.1. In Section 5.2 and Section 5.3, the TECM clustering performance is evaluated using synthetic datasets and texture image datasets and compared with other related algorithms. Finally, in Section 5.4, a parameter analysis of the proposed TECM is provided.

# 5.1. Experimental setup

In addition to ECM, six other related algorithms: LSSMTC [53], CombKM [53], TSC [49], TDEM [45], TFCM [40], and TLEC [52], are employed for comparison. Among these, LSSMTC and CombKM are representative multitask clustering algorithms, and TSC, TDEM, TFCM, and TLEC are representative transfer-clustering algorithms. The main ideas of these algorithms are described as follows.

• LSSMTC [53]: The intuitive idea of LSSMTC is to learn a subspace shared by all clustering tasks such that knowledge can be transferred through the learned subspace. The objective of LSSMTC consists of two parts. The first part involves clustering the data of each task individually; the second part involves learning the shared subspace and clustering the data of all the tasks in the shared subspace simultaneously.

• CombKM [53]: The intuitive idea of CombKM is to perform the c-means algorithm on a combined dataset created by merging the data of all the tasks together.

• TSC [49]: TSC assumes that embedded feature information can connect two clustering tasks, which can simultaneously improve the clustering performance in two domains. Therefore, the data manifold information of the clustering tasks and feature manifold information shared between the two clustering tasks are both introduced in TSC.

• TDEM [45]: TDEM was proposed to improve the clustering performance of distributed gaussian mixture model clustering and accelerate clustering convergence, where each node is regarded as both the source domain and target domain, which can learn from each other to complete the clustering task in distributed P2P networks.

• TFCM [40]: TFCM is a representative parameter-based transfer-clustering algorithm that uses the cluster centers obtained from the source domain by FCM to guide clustering in the target domain.

• TLEC [52]: Evidential clustering based on transfer learning (TLEC) is another transfer-clustering version of ECM that uses the cluster centers obtained from the source domain by ECM to guide the clustering in the target domain. However, this algorithm requires the same number of clusters in both the source and target domains.

The parameter settings of the involved algorithms are displayed in Table 2, which are the recommendations of the corresponding authors of each proposal with generally acceptable performance. For the balance coefficient of transfer learning,  $\lambda$ , the grid search strategy [54] was used to determine the optimal value. All experiments were implemented using MATLAB 2016b on a computer with an Intel i9-10900K 3.70 GHZ CPU and 64 GB RAM.

Algorithm	Parameter Settings
ECM [14]	$\alpha = 1,  \beta = 2,  \delta = 10$
	task number $T = 2$
LSSMTC [53]	$l^* \in \{2, 4, 8, 16\}$
	$\lambda^* \in \{0.2, 0.4, 0.6, 0.8\}$
CombKM [53]	k is equal to the number of clusters
TSC [49]	$\lambda = 3, step = 1$
TDEM $[45]$	$\lambda \in \{0.5:0.1:1.5\}, \eta_1 = \eta_2 = 10^{-10}$
	The fuzzifers $m_1, m_2 = 2, \varepsilon = 10^{-3}$
TFCM $[40]$	maximum iteration $= 100$
	$\lambda \in \{0, 0.1, 0.5, 1, 5, 10, 50, 100, 300, 500, 1000\}$
TLEC [52]	$\alpha = 1,  \beta = 2,  \delta = 10,  \beta_1 = \beta_2 = 1$
	$\alpha=1,\beta=2,\delta=10,\gamma=2,arepsilon=10^{-3}$
TECM	maximum iteration $= 100$
	$\lambda \in \{0, 0.1, 0.5, 1, 5, 10, 50, 100, 300, 500, 1000\}$

Table 2: Parameter settings for involved algorithms

The clustering performance was evaluated using three popular validity indices [62]: accuracy (ac), rand index (RI), and normalized mutual information (NMI). Their definitions are as follows.

(1) ac

$$ac = \frac{n_{cor}}{n},$$

where n is the total number of objects and  $n_{cor}$  is the number of objects that are clustered correctly. The greater the value of ac, the better the clustering performance.

(2) RI

$$RI = \frac{f_{00} + f_{11}}{n(n-1)/2},$$

where  $f_{00}$  and  $f_{11}$  are the number of any two objects from the same class belonging to different clusters and the same cluster, respectively. The greater the value of RI, the better the clustering performance.

(3) NMI

$$NMI(X;Y) = 2\frac{I(X;Y)}{H(X) + H(Y)},$$

where  $H(\cdot)$  is the information entropy and I(X;Y) is the mutual information, with X and Y being the real label and clustering result, respectively. The greater the value of NMI, the better the clustering performance.

#### 5.2. Synthetic dataset test

The TECM clustering performance in cases of insufficient target data and sufficient yet contaminated target data is evaluated in this section. In addition, two ambiguous datasets are used to illustrate the advantages of TECM for detecting ambiguous objects.

#### 5.2.1. Clustering performance evaluation

In this section, the TECM clustering performance is evaluated on synthetic datasets in cases where the target data are insufficient or sufficient yet contaminated, although an abundance of source data with similar data distributions are available.

• Insufficient target data: Four synthetic datasets, S1-1, T1-1, S1-2, and T1-2, were generated to verify three cases:  $c_s = c_t$ ,  $c_s > c_t$ , and  $c_s < c_t$ , where  $c_s$  and  $c_t$  are the number of clusters in the source and target domains, respectively. The parameters for generating these four synthetic datasets are listed in Table 3, and the distributions of these datasets are displayed in Fig. 5.

S1-1	mean	covariance	size
Cluster 1	$[0 \ 0 \ 0]$	$[3\ 0\ 0;\ 0\ 3\ 0;\ 0\ 0\ 3]$	200
Cluster 2	$[0 \ 0 \ 5]$	$[3\ 0\ 0;\ 0\ 3\ 0;\ 0\ 0\ 3]$	200
Cluster 3	$[0 \ 5 \ 0]$	$[3\ 0\ 0; 0\ 3\ 0; 0\ 0\ 3]$	200
T1-1	mean	covariance	size
Cluster 1	$[0 \ 0 \ 0]$	$[4\ 0\ 0;\ 0\ 4\ 0;\ 0\ 0\ 4]$	20
Cluster 2	$[0 \ 0 \ 5]$	$[4\ 0\ 0;\ 0\ 4\ 0;\ 0\ 0\ 4]$	20
Cluster 3	$[0 \ 5 \ 0]$	$[4\ 0\ 0;\ 0\ 4\ 0;\ 0\ 0\ 4]$	20
S1-2	mean	covariance	size
Cluster 1	$[0 \ 0 \ 0]$	$[3\ 0\ 0;\ 0\ 3\ 0;\ 0\ 0\ 3]$	200
Cluster 2	$[0 \ 0 \ 5]$	$[3\ 0\ 0;\ 0\ 3\ 0;\ 0\ 0\ 3]$	200
Cluster 3	$[0 \ 5 \ 0]$	$[3\ 0\ 0;\ 0\ 3\ 0;\ 0\ 0\ 3]$	200
Cluster 4	$[5 \ 0 \ 0]$	$[3\ 0\ 0; 0\ 3\ 0; 0\ 0\ 3]$	200
T1-2	mean	covariance	size
Cluster 1	$[0 \ 0 \ 0]$	$[4\ 0\ 0;\ 0\ 4\ 0;\ 0\ 0\ 4]$	20
Cluster 2	$[0 \ 0 \ 5]$	$[4\ 0\ 0;\ 0\ 4\ 0;\ 0\ 0\ 4]$	20
Cluster 3	$[0 \ 5 \ 0]$	$[4\ 0\ 0;\ 0\ 4\ 0;\ 0\ 0\ 4]$	20
Cluster 4	$[5 \ 0 \ 0]$	$[4\ 0\ 0;\ 0\ 4\ 0;\ 0\ 0\ 4]$	20

Table 3: Parameters for generating synthetic datasets S1-1, T1-1, S1-2, and T1-2



Figure 5: Distributions of synthetic datasets S1-1, T1-1, S1-2, and T1-2.

• Sufficient yet contaminated target data: Two synthetic datasets, T1-3 and T1-4, were generated to evaluate the TECM clustering performance in the case of sufficient yet contaminated target data. Dataset T1-3 was generated by adding zero-mean Gaussian noise with a standard deviation of five to S1-1. Dataset T1-4 was generated by adding zero-mean Gaussian noise with a standard deviation of three to S1-2. The distributions of T1-3 and T1-4 are displayed in Fig. 6. Four datasets, S1-1, S1-2, T1-3, and T1-4, were used to verify three cases:  $c_s = c_t$ ,  $c_s > c_t$ , and  $c_s < c_t$ .



Figure 6: Distributions of synthetic datasets T1-3 and T1-4.

To minimize the influence of random initialization on the clustering performance evaluation, each algorithm was executed ten times on each target dataset, and then the mean value of each evaluation index was recorded. The clustering performance evaluation results for insufficient target data and sufficient yet contaminated target data are listed in Table 4 and Table 5, respectively, based on the means of *ac*, *RI*, and *NMI*. In this comparison, ECM used only target data for clustering. LSSMTC and CombKM clustered target data and source data simultaneously, and TSC, TDEM, TFCM, TLEC, and TECM used knowledge learned from the source data to guide the clustering of the target data. Because LSSMTC, CombKM, TSC, TDEM, and TLEC require that the source and target domains have the same number of clusters, parts of the experimental results are absent.

			Algorithms								
Datasets	Metrics	ECM	LSSMTC	CombKM	TSC	TDEM	TFCM	TLEC	TECM		
	mean_ac	0.850	0.670	0.595	0.795	0.767	0.867	0.867	0.867		
T1-1 (S1-1⇒T1-1)	mean_RI	0.825	0.694	0.682	0.779	0.763	0.842	0.842	0.842		
	mean_NMI	0.595	0.337	0.334	0.471	0.482	0.626	0.626	0.626		
	mean_ac	0.750	/	/	/	/	0.850	/	0.867		
T1-1 (S1-2⇒T1-1)	mean_RI	0.757	/	/	/	/	0.825	/	0.842		
	mean_NMI	0.504	/	/	/	/	0.595	/	0.626		
T1-2 (S1-1⇒T1-2)	mean_ac	0.738	/	/	/	/	0.800	/	0.813		
	mean_RI	0.788	\	\	\	\	0.831	\	0.839		
	mean_NMI	0.452	\	\	\	\	0.549	\	0.559		

Table 4: Clustering performance of algorithms for insufficient target data

 $^*$  For a  $\Rightarrow$  b, a and b denote the source and target datasets, respectively.

Table 5: Clustering performance of algorithms for sufficient yet contaminated target data

	Metrics	Algorithms								
Datasets		ECM	LSSMTC	CombKM	TSC	TDEM	TFCM	TLEC	TECM	
	mean_ac	0.682	0.533	0.500	0.603	0.481	0.702	0.692	0.697	
T1-3 (S1-1⇒T1-3)	mean_RI	0.683	0.613	0.603	0.636	0.525	0.696	0.688	0.693	
	mean_NMI	0.253	0.140	0.123	0.168	0.109	0.274	0.258	0.267	
	mean_ac	0.498	/	/	/	/	0.522	/	0.548	
T1-3 (S1-2⇒T1-3)	mean_RI	0.599	/	/	/	/	0.604	/	0.607	
	mean_NMI	0.097	/	/	/	/	0.102	\	0.105	
T1-4 (S1-1⇒T1-4)	mean_ac	0.619	/	/	/	/	0.635	/	0.641	
	mean_RI	0.721	\	/	\	\	0.728	\	0.732	
	mean_NMI	0.243	/	/	/	/	0.263	\	0.264	

 $^*$  For a  $\Rightarrow$  b, a and b denote the source and target datasets, respectively.

Based on the above six groups of experiments, the following conclusions can be drawn.

(1) For all six groups of experiments, the TECM clustering performance was superior to that of ECM, which indicates that the proposed TECM improved the clustering performance with the assistance of the knowledge learned from the source domain.

(2) In the majority of cases, the TECM clustering performance was superior to that of the other representative multitask or transfer-clustering algorithms, which demonstrates its competitive transfer capability.

(3) Compared with TFCM, the performance improvement of TECM was not evident based on the indices of the hard partition. Nevertheless, the advantage of TECM over TFCM was not only reflected in its hard-partition performance. For certain cases with complex data distributions, TECM with credal partition can provide a deeper insight into the data compared to TFCM. This point will be discussed further in future studies.

#### 5.2.2. Comparison between TECM and TFCM

In this section, two synthetic ambiguous datasets are used to illustrate the superiority of TECM over TFCM in detecting ambiguous objects. The parameters for generating the synthetic datasets T2-1 and T2-2 are listed in Table 6, and the distributions of the datasets are displayed in Fig. 7. The number of clusters in TECM and TFCM were set to two for T2-1 and four for T2-2; TECM had three barycenters for T2-1 ( $w_1$ ,  $w_2$  and  $w_1 \cup w_2$ ) and 15 barycenters for T2-2 ( $w_1$ ,  $w_2$ ,  $w_1 \cup w_2$ ,...,  $w_1 \cup w_2 \cup w_3 \cup w_4$ ).

S2-1	mean	covariance	size
Cluster 1	$[0 \ 0]$	$[1\ 0;\ 0\ 1]$	100
Cluster 2	$[1 \ 0]$	$[1 \ 0; \ 0 \ 1]$	100
T2-1	mean	covariance	size
Cluster 1	$[0 \ 0.2]$	$[1\ 0;\ 0\ 1]$	10
Cluster 2	$[1 \ 0.2]$	$[1 \ 0; \ 0 \ 1]$	10
S2-2	mean	covariance	size
Cluster 1	$[0 \ 0]$	$[1\ 0;\ 0\ 1]$	100
Cluster 2	$[1 \ 0]$	$[1\ 0;\ 0\ 1]$	100
Cluster 3	$[0 \ 1]$	$[1\ 0;\ 0\ 1]$	100
Cluster 4	$[1 \ 1]$	$[1 \ 0; \ 0 \ 1]$	100
T2-2	mean	covariance	size
Cluster 1	$[0.2 \ 0.2]$	$[1\ 0;\ 0\ 1]$	30
Cluster 2	$[1.2 \ 0.2]$	$[1\ 0;\ 0\ 1]$	30
Cluster 3	$[0.2 \ 1.2]$	$[1\ 0;\ 0\ 1]$	30
Cluster 4	$[1.2 \ 1.2]$	$[1\ 0;\ 0\ 1]$	30

Table 6: Parameters used to generate synthetic datasets S2-1, T2-1, S2-2, and T2-2



Figure 7: Distributions of synthetic datasets S2-1, T2-1, S2-2, and T2-2.

Fig. 8 displays the clustering results of target data T2-1 and T2-2 obtained by transferring the corresponding source data S2-1 and S2-2, respectively. It can be observed that TFCM cannot cluster ambiguous objects well, whereas the proposed TECM can discover those objects located at the boundaries of different clusters and assign them to ambiguous sets of clusters. These examples confirm that the TECM proposed in this study is more effective in detecting ambiguous objects, and thus provides a deeper insight into the distribution of the data. In addition, Table 7 indicates that TECM's clustering performance is superior or comparable to that of TFCM in terms of ac, RI, and NMI.



Figure 8: Clustering results of T2-1 and T2-2 using TFCM and TECM.

Table 7:	Comparison	of clustering	performance	of TECM	and	TFCM	on T	2-1	and	T2-2.

Algorithms	Т2-	1 (S2-1⇒	·T2-1)	T2-2 (S2-2 $\Rightarrow$ T2-2)				
Ū	mean_ac	mean_RI	mean_NMI	mean_ac	mean_RI	mean_NMI		
TFCM	0.850	0.732	0.399	0.633	0.735	0.371		
TECM	0.850	0.732	0.399	0.700	0.766	0.428		

\* For  $a \Rightarrow b$ , a and b denote the source and target datasets, respectively.

#### 5.3. Texture image segmentation

In this section, five basic textures D6, D8, D11, D46, and D96 in the Brodatz texture image dataset [63] are used to create the texture images employed in this experiment; these texture images are then resized to 90 by 90 pixels. In addition, Gaussian noise is added to the target images. Fig. 9 (a) displays the source image and Figs. 9 (b)–9 (g) display the three types of target images

in our experiments. Specifically, Figs. 9 (b) and 9 (c) have the same texture number as that in Fig. 9 (a) but have an inconsistent data distribution. Figs. 9 (d) and 9 (e) have smaller texture numbers than those in Fig. 9 (a)). Figs. 9 (f) and 9 (g) have larger texture numbers than those in Fig. 9 (a)). In addition, the target images had different noise levels. The number of textures and noise levels in each image are given in brackets. The texture features of the images were obtained using the Gabor filter proposed in [64]. The ideal segmentation results for the target images are displayed in Fig. 10. In each subgraph, small squares of the same color represent the same texture.



(a) S3-1 (K=4,  $\sigma=0$ )



(b) T3-1 (K=4,  $\sigma$ =0.05)



(e) T3-4 (K=3,  $\sigma{=}0.05)$ 







(d) T3-3 (K=3,  $\sigma$ =0.05)



(g) T3-6 (K=5,  $\sigma$ =0.05)

Figure 9: Texture images used in the experiment.



Figure 10: Ideal segmentation of target texture images.

Fig. 11–Fig. 16 display the segmentation results for each target image using the different algorithms. In addition, Table 8 provides the quantitative scores of *ac*, *RI*, and *NMI* obtained using the different algorithms. For the same reasons as those indicated in Section 5.2, the results of LSSMTC, CombKM, TSC, TEDM, and TLEC for certain datasets are absent. Based on the above results, the following conclusions can be drawn.

1) Owing to the existence of noise, ECM could not achieve acceptable texture segmentation performance on noisy target images. Conversely, benefiting from the reference information across domains, TECM could obtain a relatively superior performance compared to ECM. This demonstrates the effectiveness of transfer clustering.

2) TECM and TFCM could obtain relatively superior performance compared to the other multitask and transfer-clustering algorithms. Owing to the existence of noise and differences in data distribution between the two domains, raw source data were unsuitable for direct use in the clustering of the target data. Therefore, LSSMTC and CombKM could easily encounter negative effects unexpectedly between tasks, rather than simultaneously achieving performance improvements. 3) TECM and TFCM were similar in terms of the clustering indices of the hard partition. However, it can be observed from the experimental results that in the cases where TFCM was superior, the clustering performance gap between TFCM and TECM was small. However, in the cases where TECM was superior, the clustering performance gap between TECM and TFCM was large, which indicates that the proposed TECM was more powerful than TFCM overall. In addition, TECM had an advantage over TFCM in that it provided a deeper insight into the data when the data distribution was complex, as demonstrated in Section 5.2.2.

4) Regardless of whether the number of clusters in the source domain was equal to, greater than, or less than the number of clusters in the target domain, the TECM clustering performance always improved in comparison with ECM. Therefore, if the structure of the source data is partially similar to that of the target data, the learned knowledge from the source data will be beneficial for clustering the target data.



Figure 11: Segmentation results of T3-1 obtained by eight clustering algorithms.



Figure 12: Segmentation results of T3-2 obtained by eight clustering algorithms.



Figure 13: Segmentation results of T3-3 obtained by three clustering algorithms.



Figure 14: Segmentation results of T3-4 obtained by three clustering algorithms.



Figure 15: Segmentation results of T3-5 obtained by three clustering algorithms.



Figure 16: Segmentation results of T3-6 obtained by three clustering algorithms.

	Matelaa	Algorithms								
Datasets	Metrics	ECM	LSSMTC	CombKM	TSC	TDEM	TFCM	TLEC	TECM	
	mean_ac	0.705	0.664	0.646	0.690	0.805	0.793	0.729	0.836	
T3-1 (S3-1⇒T3-1)	mean_RI	0.815	0.751	0.749	0.721	0.822	0.818	0.764	0.865	
	mean_NMI	0.536	0.472	0.443	0.380	0.586	0.579	0.480	0.629	
	mean_ac	0.722	0.604	0.561	0.677	0.718	0.735	0.685	0.742	
T3-2 (S3-1⇒T3-2)	mean_RI	0.786	0.726	0.715	0.723	0.770	0.794	0.733	0.828	
	mean_NMI	0.463	0.401	0.368	0.406	0.417	0.480	0.424	0.503	
	mean_ac	0.774	/	/	/	/	0.918	/	0.916	
T3-3 (S3-1⇒T3-3)	mean_RI	0.785	/	/	/	/	0.895	/	0.901	
	mean_NMI	0.550	/	/	/	/	0.734	/	0.700	
	mean_ac	0.814	/	/	/	\	0.814	\	0.840	
T3-4 (S3-1⇒T3-4)	mean_RI	0.806	/	/	/	/	0.806	/	0.830	
T3-4 (S3-1⇒T3-4)	mean_NMI	0.566	/	/	/	/	0.566	/	0.564	
	mean_ac	0.800	/	/	/	/	0.879	/	0.877	
T3-5 (S3-1⇒T3-5)	mean_RI	0.875	/	/	/	/	0.916	/	0.916	
	mean_NMI	0.671	/	/	/	/	0.755	/	0.738	
	mean_ac	0.874	/	/	\	/	0.902	/	0.890	
T3-6 (S3-1⇒T3-6)	mean_RI	0.910	/	/	\	\	0.925	\	0.920	
	mean_NMI	0.707	/	/	\	\	0.751	\	0.737	

Table 8: Clustering performance comparisons of six algorithms on T3-1 to T3-6

 $^*$  For a  $\Rightarrow$  b, a and b denote the source and target datasets, respectively.

#### 5.4. Parameter analysis

The balance coefficient of transfer learning  $\lambda$  is an important parameter in the TECM algorithm, which significantly influences the algorithm performance. In this section, the influence of  $\lambda$  on the TECM clustering performance is discussed. Here, the synthesis datasets described in Table 3 were used to study the variation in the algorithm performance with  $\lambda$ . The performance of the algorithm was evaluated using *ac*.

Fig. 17 indicates that the TECM clustering performance first improves and then decreases with an increase in  $\lambda$ . When the value of  $\lambda$  is relatively small, the clustering performance is not significantly improved compared to ECM because the transferred knowledge has a small role. When the value of  $\lambda$  is relatively large, the clustering performance is even worse than that of ECM because the clustering is completely dependent on the knowledge learned from the source data, which results in a negative transfer. In practice, the grid search strategy can be used to determine an effective value of  $\lambda$  by optimizing an internal clustering index.



Figure 17: TECM clustering performance with different values of  $\lambda$ .

#### 6. Conclusions

In this study, a TECM clustering algorithm was proposed to improve the clustering performance of the target data with the help of barycenters learned from the source data for situations where target data are insufficient or contaminated. The proposed algorithm is applicable to situations where the source and target domains have the same or different numbers of clusters. In synthetic and real dataset experiments, the proposed TECM achieved approximately 5% and 7% performance improvement over ECM for insufficient and contaminated target data on average, respectively. In addition, compared with other related multitask or transfer-clustering algorithms, the proposed TECM performed the best overall.

In the future, this research topic can be further explored by integrating TECM with deep learning such that the clustering algorithm can be better applied to raw data such as images and texts.

# Acknowledgments

This work was funded by the National Natural Science Foundation of China (Grant Nos. 62171386, 61801386, and 61790552), Key Research and Development Program in Shaanxi Province of China (Grant No. 2022GY-081), and China Postdoctoral Science Foundation (Grant Nos. 2019M653743 and 2020T130538).

#### References

# References

- X. Chang, Q. Wang, Y. Liu, and Y. Wang, Sparse regularization in fuzzy c-means for high-dimensional data clustering, *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2616–2627, 2017.
- [2] I. D. Borlea, R. E. Precup, A. B. Borlea, and D. Iercan, A unified form of fuzzy c-means and k-means algorithms and its partitional implementation, *Knowl-Based. Syst.*, vol. 214, pp. 106731, 2021.
- [3] Q. Cai and J. Liu, Hierarchical clustering of bipartite networks based on multiobjective optimization, *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 421–434, 2020.
- [4] C. Wu, Q. Peng, J. Lee, K. Leibnitz, and Y. Xia, Effective hierarchical clustering based on structural similarities in nearest neighbor graphs, *Knowl-Based. Syst.*, vol. 228, pp. 107295, 2021.
- [5] Z. Su and T. Denoeux, BPEC: belief-peaks evidential clustering, *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 1, pp. 111–123, 2019.
- [6] Y. Ren, N. Wang, M. Li, and Z. Xu, Deep density-based image clustering, *Knowl-Based. Syst.*, vol. 197, pp. 105841, 2020.
- [7] M. Tareq, E. A. Sundararajan, M. Mohd, and N. S. Sani, Online clustering of evolving data streams using a density grid-based method, *IEEE Access*, vol. 8, pp. 166472–166490, 2020.

- [8] C. Hireche, H. Drias, and H. Moulai, Grid based clustering for satisfiability solving, *Appl. Soft Comput.*, vol. 88, pp. 106069, 2020.
- [9] H. Wang, Y. Yang, and B. Liu, GMC: graph-based multi-view clustering, *IEEE Trans. Knowl.*, Data, Eng., vol. 32, no. 6, pp. 1116–1129, 2020.
- [10] Z. Zhang, Q. Zhu, F. Zhu, J. Li, D. Cheng, Y. Liu, and J. Luo, Density decay graph-based density peak clustering, *Knowl-Based. Syst.*, vol. 224, pp. 107075, 2021.
- [11] Y. Gao, Z. Wang, J. Xie, and J. Pan, A new robust fuzzy c-means clustering method based on adaptive elastic distance, *Knowl-Based. Syst.*, vol. 237, pp. 107769, 2022.
- [12] J. Zhou, Z. Lai, C. Gao, D. Miao, and X. Yue, Rough possibilistic c-means clustering based on multigranulation approximation regions and shadowed sets, *Knowl-Based. Syst.*, vol. 160, pp. 144-166, 2018.
- [13] T. Zhang and F. Ma, Improved rough k-means clustering algorithm based on weighted distance measure with Gaussian function, *Int. J. Comput. Math.*, vol. 94, no. 4, pp. 663–675, 2017.
- [14] M. H. Masson and T. Denoeux, ECM: an evidential version of the fuzzy c-means algorithm, *Pattern. Recog.*, vol. 41, no. 4, pp. 1384–1397, 2008.
- [15] Z. G. Liu, Q. Pan, J. Dezert, and G. Mercier, Credal c-means clustering method based on belief functions, *Knowl-Based. Syst.*, vol. 74, pp. 119–132, 2015.
- [16] T. Denoeux and O. Kanjanatarakul, Evidential clustering: A review, in Proc. 5th Int. Symp. Integr. Uncertainty Knowl. Model. Decis. Making, pp. 24–35, 2016.
- [17] T. Denoeux, S. Sriboonchitta, O. Kanjanatarakul, Evidential clustering of large dissimilarity data, *Knowl-Based. Syst.*, vol. 106, pp. 179–195, 2016.

- [18] L. Jiao, T. Denoeux, Z. G. Liu, and Q. Pan, EGMM: an evidential version of the gaussian mixture model for clustering, *Appl. Soft Comput.*, vol. 129, pp. 109619, 2022.
- [19] T. Denoeux, NN-EVCLUS: Neural network-based evidential clustering, Inf. Sci., vol. 572, pp. 297–330, 2021.
- [20] Z. Zhang, Z. Liu, A. Martin, Z. Liu, and K. Zhou, Dynamic evidential clustering algorithm, *Knowl-Based. Syst.*, vol. 213, pp. 106643, 2021.
- [21] A. P. Dempster, Upper and lower probabilities induced by a multi-valued mapping, Ann. Math. Stat., vol. 38, pp. 325–339, 1967.
- [22] G. Shafer, A Mathematical Theory of Evidence, Princeton, NJ: Princeton University Press, 1976.
- [23] C. Lian, S. Ruan, T. Denoeux, H. Li, and P. Vera, Spatial evidential clustering with adaptive distance metric for tumor segmentation in FDG-PET images, *IEEE Trans. Biomed. Eng.*, vol. 65, no. 1, pp. 21–30, 2017.
- [24] C. Lian, S. Ruan, T. Denoeux, H. Li, and P. Vera, Joint tumor segmentation in PET-CT images using co-clustering and fusion based on belief functions, *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 755–766, 2018.
- [25] K. Zhou, A. Martin, Q. Pan, and Z. G. Liu, Median evidential c-means algorithm and its application to community detection, *Knowl-Based. Syst.*, vol. 74, pp. 69–88, 2015.
- [26] K. Zhou, A. Martin, Q. Pan, and Z. G. Liu, SELP: semi-supervised evidential label propagation algorithm for graph data clustering, *Int. J. Approx. Reason.*, vol. 92, pp. 139–154, 2018.
- [27] R. Abdelkhalek, I. Boukhris, and Z. Elouedi, An evidential collaborative filtering approach based on items contents clustering, in *Proc. 5th Int. Conf. Belief Functions*, 2018, pp. 1–9.

- [28] R. Abdelkhalek, I. Boukhris, and Z. Elouedi, An evidential clustering for collaborative filtering based on users preferences, in *Proc. 16th Int. Conf. Model. Decis. Artif. Intell.*, 2019, pp. 224–235.
- [29] T. Denoeux and M. H. Masson, EVCLUS: evidential clustering of proximity data, *IEEE Trans. Syst. Man Cybern. B*, vol. 34, no. 1, pp. 95–109, 2004.
- [30] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, Transfer learning using computational intelligence: A survey, *Knowl-Based. Syst.*, vol. 80, pp. 14–23, 2015.
- [31] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, A comprehensive survey on transfer learning, *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, 2021.
- [32] S. Pan and Q. Yang, A survey on transfer learning, *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 10, pp.1345–1359, 2010.
- [33] Z. Liu, L. Huang, K. Zhou, and T. Denoeux, Combination of transferable classification with multi-source domain adaptation based on evidential reasoning, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 2015– 2029, 2021.
- [34] G. Xu, Z. Zhang, T. Zhang, S. Yu, Y. Meng, and S. Chen, Aspectlevel sentiment classification based on attention-BiLSTM model and transfer learning, *Knowl-Based. Syst.*, vol. 245, pp. 108586, 2022.
- [35] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, Fuzzy regression transfer learning in Takagi-Sugeno fuzzy models, *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1795–1807, 2017.
- [36] X. Liu, Y. Li, and G. Chen, Transfer learning for regression via latent variable represented conditional distribution alignment, *Knowl-Based. Syst.*, vol. 240, pp. 108110, 2022.

- [37] L. Zhang, W. Zuo, and D. Zhang, LSDT: Latent sparse domain transfer learning for visual adaptation, *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1177–1191, 2016.
- [38] Y. Chen, S. Song, S. Li, L. Yang, and C. Wu, Domain space transfer extreme learning machine for domain adaptation, *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1909–1922, 2019.
- [39] P. Qian, Y. Jiang, Z. Deng, L. Hu, S. Sun, S. Wang, and R. F. Muzic, Cluster prototypes and fuzzy memberships jointly leveraged cross-domain maximum entropy clustering, *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 181– 193, 2015.
- [40] Z. Deng, Y. Jiang, F. Chung, H. Ishibuchi, K. S. Choi, and S. Wang, Transfer prototype-based fuzzy clustering, *IEEE Trans. Fuzzy Syst.* vol. 24, no. 5, pp. 1210–1232, 2015.
- [41] F. Wang, L. Jiao, and Q. Pan, A Survey on unsupervised transfer clustering, in *Proc. 40th Chin. Control Conf.*, 2021, pp. 7361–7365.
- [42] S. Kong and D. Wang, Transfer heterogeneous unlabeled data for unsupervised clustering, in *Proc. 21th Int. Conf. Pattern Recog.*, 2013, pp. 1193–1196.
- [43] L. Yang, L. Jing, B. Liu, and J. Yu, Common latent space identification for heterogeneous co-transfer clustering, *Neurocomputing*, vol. 269, pp. 29–39, 2017.
- [44] K. Xia, H. Yin, Y. Jin, S. Qiu, and H. Zhao, Cross-domain brain CT image smart segmentation via shared hidden space transfer FCM clustering. ACM Trans. Multi. Comput., Commun., Appl., vol. 16, no. 2s, pp. 1–21, 2020.
- [45] R. Wang, S. Han, J. Zhou, Y. Chen, L. Wang, T. Du, K. Ji, Y. Zhao, K. Zhang, Transfer-learning-based gaussian mixture model for distributed clustering, *IEEE Trans. Cybern.*, early access, 10 June 2022, DOI:10.1109/TCYB.2022.3177242.

- [46] P. Qian, K. Zhao, Y. Jiang, K. Su, Z. Deng, S. Wang, and R. F. Muzic, Knowledge-leveraged transfer fuzzy c-means for texture image segmentation with self-adaptive cluster prototype matching, *Knowl-Based Syst.*, vol. 130, pp. 33–50, 2017.
- [47] Y. Cheng, Y. Jiang, P. Qian, and S. Wang, A maximum entropy clustering algorithm based on knowledge transfer and its application to texture image segmentation, *CAAI Trans. Intell. Syst.*, vol. 12, no. 2, pp. 179–187, 2017.
- [48] R. Gargees, J. M. Keller, and M. Popescu, TLPCM: Transfer learning possibilistic c-means, *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 4, pp. 940–952, 2020.
- [49] W. Jiang and F. Chung, Transfer spectral clustering, in Proc. Eur. Conf. Mach. Learn. Knowl. Disc. Databases, 2012, pp. 789–803.
- [50] L. Yu, Y. Dang, and G. Yang, Transfer clustering via constraints generated from topics, in *Proc. IEEE Int. Conf. Syst.*, Man and Cybern., 2012, pp. 3203–3208.
- [51] L. Jiao, F. Wang, and Q. Pan, Transfer evidential c-means clustering, in Proc. 6th Int. Conf. Belief Functions, 2021, pp. 47-55.
- [52] K. Zhou, M. Guo, and A. Martin, Evidential clustering based on transfer learning, in *Proc. 6th Int. Conf. Belief Functions*, 2021, pp. 56-65.
- [53] Q. Gu and J. Zhou, Learning the shared subspace for multi-task clustering and transductive transfer classification, in *Proc. 9th IEEE Int. Conf. Data Mining*, 2009, pp. 159–168.
- [54] Z. Deng, K. S. Choi, F. L. Chung, S. Wang, Enhanced soft subspace clustering integrating within-cluster and between-cluster information, *Pattern Recog.*, vol. 43, no. 3, pp.767–781, 2010.
- [55] J. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithm, New York, NY: Plenum Press, 1981.

- [56] S. Zhou, D. Li, Z. Zhang, and R. Ping, A new membership scaling fuzzy c-means clustering algorithm, *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 9, pp. 2810–2818, 2020.
- [57] R. Krishnapuram and J. Keller, A possibilistic approach to clustering, *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 98–111, 1993.
- [58] G. Peters, Is there any need for rough clustering? Pattern Recogn. Lett., vol. 53, pp. 31–37, 2015.
- [59] R. Saltos, R. Weber, and S. Maldonado, Dynamic rough-fuzzy support vector clustering, *IEEE Trans. Fuzzy Syst.*, vol. 25, pp. 1508–1521, 2017.
- [60] R. Hathaway, J. Bezdek, and W. Tucker, An improved convergence theorem for the fuzzy c-means clustering algorithms, *Anal. Fuzzy Inf.*, vol. 3, pp. 123– 129, 1987.
- [61] G. Gan and J. Wu, A convergence theorem for the fuzzy subspace clustering (FSC) algorithm, *Pattern Recog.*, vol. 41, no. 6, pp. 1939–1947, 2008.
- [62] H. Schütze, C.D. Manning, P. Raghavan, Introduction to Information Retrieval, London, UK: Cambridge University Press, 2008.
- [63] T. Randen, Brodatz Texture, http://www.ux.uis.no/tranden/brodatz.html, 2021-10-01.
- [64] V. Kyrki, J. K. Kamarainen, and H. Kaelviaeinen, Simple Gabor feature space for invariant object recognition, *Pattern Recog. Letts.*, vol. 25, no. 3, pp. 311–318, 2004.