

# Evidential prototype-based clustering based on transfer learning

Kuang Zhou<sup>a,\*</sup>, Mei Guo<sup>a</sup>, Arnaud Martin<sup>b</sup>

<sup>a</sup>*School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, PR China*

<sup>b</sup>*Univ Rennes, CNRS, IRISA, DRUID, Rue E. Branly, 22300 Lannion, France*

---

## Abstract

In some real clustering tasks, the data may be sparse and uncertain. Although there is usually some useful knowledge in related scenes, the data among different domains is often of great inconsistency. A new unsupervised transfer learning method is proposed in the framework of belief functions to handle the insufficiency and uncertain problems in clustering simultaneously. Firstly, under the assumption that the source and target domains have the same number of clusters, the Transfer Evidential  $C$  Means (TECM) is developed by incorporating the idea of transfer learning and evidential clustering. A novel objective function is designed to employ the cluster prototypes of the source data as references to guide the clustering process on the target. Furthermore, ETECM, as an extended version of TECM, is also introduced for the situation that the two domains have different numbers of clusters. Some experiments conducted on synthetic and real-world data sets demonstrate the advantages of TECM and ETECM.

*Keywords:* Belief functions, Clustering, Transfer learning, Uncertainty, Source domain

---

## 1. Introduction

Clustering is an unsupervised technique aiming to assign patterns into groups, and it has been widely used in many fields such as image segmentation, market research, and data analysis [2, 3]. Traditional clustering methods, such as the  $c$ -means method, usually work well when the data are sufficient. However, in some real clustering tasks, we cannot get sufficient data to train a fine partition model. To address the problem caused by insufficient information, several advanced clustering approaches have been developed, such as semi-supervised learning [4], multi-view clustering [5], transfer learning [6, 7], and so on.

Transfer learning can learn an effective model for the target domain by effectively leveraging useful information from the source domain [8, 9]. Based on different situations between the source and target domains and tasks, transfer learning methods can be categorized into three sub-settings: inductive transfer learning, transductive transfer learning, and unsupervised transfer learning [9]. In the inductive transfer learning setting, the tasks in the target and source domains are different, no matter when the two domains are the same or not. Dai *et.al.* [10] extended the AdaBoost algorithm and proposed a transferred version named

---

\*Corresponding author

*Email addresses:* kzhoumath@nwpu.edu.cn (Kuang Zhou), guomei@mail.nwpu.edu.cn (Mei Guo), Arnaud.Martin@univ-rennes1.fr (Arnaud Martin)

This paper is an extension and revision of [1].

TrAdaBoost. It boosts a basic learner to transfer knowledge from one distribution to another distribution. On the contrary, the transductive transfer learning requires that the source and target tasks are the same, while the two domains are different [11]. Ling *et.al.* proposed an approach called Cross-Domain Spectral Classifier (CDSC) [12]. This method can classify the unlabeled data in out-of-domains by utilizing the labeled data from in-domains. Similar to the inductive transfer learning, in the unsupervised transfer learning setting, the target task is different from but related to the source task. However, it focuses on solving unsupervised learning tasks in the target domain, such as clustering, dimensionality reduction, and density estimation[13, 14].

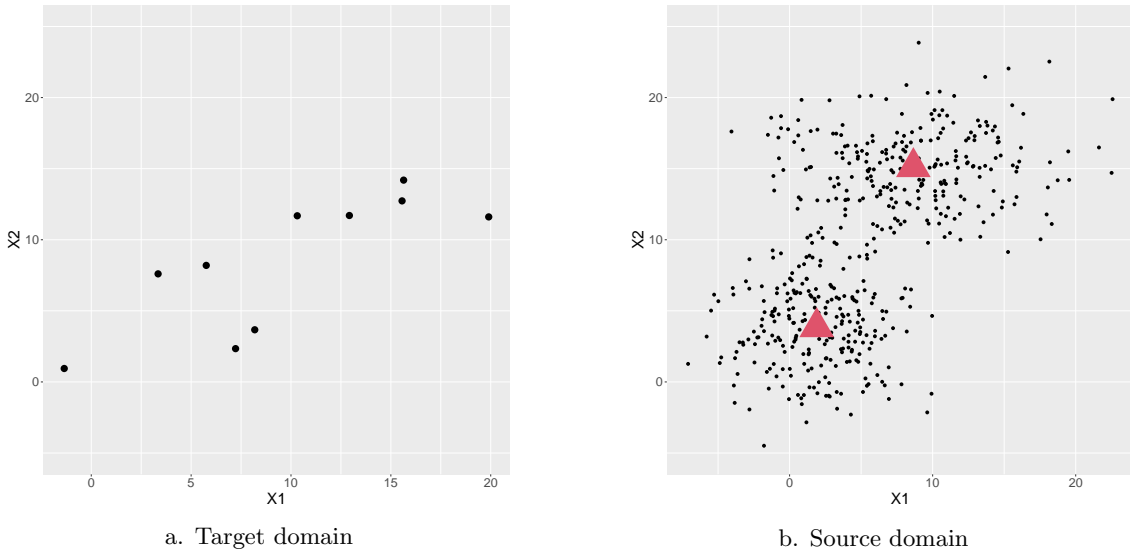


Figure 1: An example where transfer learning is required for the clustering task.

In this paper, we mainly focus on the transfer clustering problem. In the case of an unsupervised learning task with small number of samples, it is difficult to obtain an ideal partition (such as the data illustrated in the left figure in Fig. 1). These cases are not unusual in reality. For instance, in applications, new challenges frequently appear, *e.g.*, load balancing in distributed systems and attenuation correction in medical imaging. It is difficult to accumulate abundant and reliable data in the beginning phase for these new applications [15]. At this time, if some information from the source-related domain (see the right figure in Fig. 1) is available, more promising clustering results can be expected with the help of transfer learning technology. In general, two kinds of information can be transferred from the source to the target domain: raw data and knowledge [16]. Due to the necessity of privacy protection in some applications, such as users' personal information, the original raw data in the source domain are not always accessible. Thus, employing some advanced knowledge from the source domain instead of raw data is more practical. For example, in the clustering task, the cluster prototypes of the source data (red triangles in the right figure) can be regarded as good references for the target domain. There are already some transfer clustering methods. Deng *et.al.* [6] extended Fuzzy *C*-Means (FCM) to the transfer clustering scenario by exploiting the idea of leveraging knowledge from the source domain, and developed the Transfer FCM algorithms (TFCM). Gargees *et.al.* [7] proposed a Transfer-learning Possibilistic *C*-Means (TPCM) algorithm to handle the soft clustering problem in a domain that has insufficient

data. Jiang *et.al.* [17] presented a Negative-Transfer-Resistant Fuzzy Clustering model with a Shared Cross-domain Transfer latent space (named NTR-FC-SCT) by integrating negative-transfer-resistant and maximum mean discrepancy into the framework of FCM. The method has been successfully applied to brain CT image segmentation.

Another challenge in clustering is that the data are often of great uncertainty. Some objects distributed in the overlapped area among different specific clusters may be difficult to be assigned to a specific group. Although the available knowledge in the source domain can help us improve the cluster model, the inconsistency between information from the two domains may increase the uncertainty of the data. The theory of belief functions is an efficient mathematical tool for uncertain information representation and fusion [18], and it has been widely applied in image analysis [19, 20], social networks [21], reliability analysis [22], statistical inference [23, 24] *et.al.*

The concept of credal partitions defined in the framework of belief functions was first proposed by Dencœux *et.al.* [25] to deal with the uncertain cluster structure by introducing imprecise (meta) classes. Many evidential clustering methods have been designed for credal partitions [26–28]. Evidential *c*-means (ECM) [26] is a direct generalization of FCM in the framework of belief functions. Antoine *et.al.* [29] introduced a semi-supervised version of ECM algorithm called CECM, taking pairwise constraints into account. A new heuristic algorithm is presented in [30], which relaxes the constraints of semi-supervised evidential clustering in such a way that the optimization problem can be solved by using the Lagrangian method. Dencœux [31] presented a method to construct mass functions for representing the cluster-membership uncertainty, by bootstrapping finite mixture models. A neural network-based evidential clustering algorithm named NN-EVCLUS, which learns a mapping from attribute vectors to mass functions, is discussed in Ref. [32]. There are also some supervised transfer learning methods based on the theory of belief functions. For instance, a decision-level combination method was proposed by Liu *et.al.* [33], which can improve the classification accuracy through multi-source domain adaptation. Liu *et.al.* proposed a transfer classification method [34], where the source and target domains are linked by estimating the mapping from target pattern values to the source domain. Huang *et.al.* [35] proposed an evidential combination of augmented multi-source of information approach to deal with multi-source domain adaptation issues.

In this paper, we combine the idea of evidential clustering with transfer learning to develop a new clustering method, named Transfer Evidential *C*-means (TECM), for insufficient and uncertain data. This method can employ the cluster prototypes of the source data as references to guide the clustering process on the target domain with the theory of belief function. The experimental results on synthetic data and real data show the effectiveness of the proposed method. The preliminary results are published in our prior work in [1]. In this paper, we propose an extended version called ETECM to make the evidential transfer clustering method applicable to the case where the source and target domains have different numbers of clusters. Besides, more experiments are conducted to show the performance of the proposed methods.

The remainder of this paper is organized as follows. Some basic knowledge related to the proposed method is briefly introduced in Section 2. The proposed TECM algorithm is

presented in detail in Section 3. The extended version of TECM, called ETECM, is proposed in Section 4. Numerical experiments are conducted in Section 5. Conclusions are drawn in the final section.

## 2. Background

In this section, some related preliminary knowledge, including the transfer fuzzy  $c$ -means and the theory of belief functions will be presented.

### 2.1. Transfer fuzzy $c$ -means

Transfer fuzzy  $c$ -means [6] is a prototype-based transfer clustering method. It attempts to cluster the target domain instances by transferring knowledge from the source domains prototypes  $\mathbf{V}^{(s)} = \{\mathbf{v}_1^{(s)}, \mathbf{v}_2^{(s)}, \dots, \mathbf{v}_{c_s}^{(s)}\}$ . Define the data in the target domain by  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ . Notations  $\mathbf{U} = [\mu_{ij}]_{(c_t \times n)}$  and  $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_{c_t})_{(c_t \times p)}$  are used to describe the membership matrix and the prototype matrix of the target domain samples respectively. Assume that the source and target domains have the same number of clusters. The objective function of TFCM can be given as

$$\begin{aligned} J_{\text{TFCM}} &= \sum_{i=1}^{c_t} \sum_{j=1}^n u_{ij}^m \|\mathbf{x}_j - \mathbf{v}_i\|^2 + \lambda_1 \sum_{i=1}^{c_t} \sum_{j=1}^n u_{ij}^m \|\mathbf{x}_j - \mathbf{v}_i^{(s)}\|^2 \\ &\quad + \lambda_2 \sum_{i=1}^{c_t} \sum_{j=1}^n u_{ij}^m \|\mathbf{v}_i^{(s)} - \mathbf{v}_i\|^2, \\ \text{s.t. } &u_{ij} \in [0, 1], \sum_{i=1}^{c_t} u_{ij} = 1, 0 < \sum_{i=1}^n u_{ij} < n, \end{aligned} \quad (1)$$

where  $m \in [1, +\infty]$  is the weighting exponent reflecting the fuzziness degree. Parameters  $\lambda_1$  and  $\lambda_2$  are used to balance the influence of target domain learning and source domain learning. TFCM can find the best partition of the target data by iteratively updating the membership and prototype matrices. The updating rule is

$$\mathbf{v}_i = \frac{\sum_{j=1}^n u_{ij}^m \mathbf{x}_j + \lambda_2 \sum_{j=1}^n u_{ij}^m \mathbf{v}_i^{(s)}}{\sum_{j=1}^n u_{ij}^m + \lambda_2 \sum_{j=1}^n u_{ij}^m}, \quad (2)$$

$$u_{ij} = \frac{\left( \frac{1}{\|\mathbf{x}_j - \mathbf{v}_i\|^2 + \lambda_1 \|\mathbf{x}_j - \mathbf{v}_i^{(s)}\|^2 + \lambda_2 \|\mathbf{v}_i - \mathbf{v}_i^{(s)}\|^2} \right)^{\frac{1}{(m-1)}}}{\sum_{k=1}^{c_t} \left( \frac{1}{\|\mathbf{x}_j - \mathbf{v}_k\|^2 + \lambda_1 \|\mathbf{x}_j - \mathbf{v}_k^{(s)}\|^2 + \lambda_2 \|\mathbf{v}_k - \mathbf{v}_k^{(s)}\|^2} \right)^{\frac{1}{(m-1)}}}. \quad (3)$$

In order to apply TFCM to the case where the two domains have different numbers of cluster, the authors have introduced an extended version of TFCM (denoted by E-TFCM)

[6]. The objective function of E-TFCM is

$$\begin{aligned}
J_{\text{E-TFCM}} &= \sum_{i=1}^{c_t} \sum_{j=1}^n u_{ij}^{m_1} \|\mathbf{x}_j - \mathbf{v}_i\|^2 + \lambda_1 \sum_{i=1}^{c_t} \sum_{l=1}^{c_s} r_{il}^{m_2} \|\mathbf{v}_l^{(s)} - \mathbf{v}_i\|^2 \\
\text{s.t. } u_{ij} &\in [0, 1], \sum_{i=1}^{c_t} u_{ij} = 1, 0 < \sum_{j=1}^n u_{ij} < n, \\
r_{ij} &\in [0, 1], \sum_{i=1}^{c_t} r_{il} = 1, 0 < \sum_{l=1}^{c_s} r_{il} < c_s.
\end{aligned} \tag{4}$$

In this term,  $r_{ij}$  describes the similarity between prototype  $\mathbf{v}_i$  and  $\mathbf{v}_j^{(s)}$ , which is a nonnegative parameter that should be learned. For minimizing  $J_{\text{E-TFCM}}$ , the update date rule can be derived from

$$\mathbf{v}_i = \frac{\sum_{j=1}^n u_{ij}^{m_1} \mathbf{x}_j + \lambda_1 \sum_{l=1}^{c_s} r_{il}^{m_2} \mathbf{v}_l^{(s)}}{\sum_{j=1}^n u_{ij}^{m_1} + \lambda_1 \sum_{l=1}^{c_s} r_{il}^{m_2}}, \tag{5}$$

$$u_{ij} = \left( \frac{1}{\|\mathbf{x}_j - \mathbf{v}_i\|^2} \right)^{1/(m_1-1)} \bigg/ \sum_{k=1}^{c_t} \left[ \frac{1}{\|\mathbf{x}_j - \mathbf{v}_k\|^2} \right]^{1/(m_1-1)}, \tag{6}$$

$$r_{il} = \frac{1}{\sum_{q=1}^{c_s} \|\mathbf{v}_l^{(s)} - \mathbf{v}_i\|^2 / \|\mathbf{v}_q^{(s)} - \mathbf{v}_i\|^{2\frac{1}{m_2-1}}}. \tag{7}$$

As mentioned, the inconsistency among information from different domains may increase uncertainty. The theory of belief functions can be adopted to deal with the uncertain information. We will introduce some basic knowledge related to this theory in the following.

## 2.2. Theory of belief functions

Let  $\Omega = \{\omega_1, \omega_2, \dots, \omega_c\}$  be the finite domain, called the discernment frame. The belief functions are defined on the power set  $2^\Omega = \{A : A \subseteq \Omega\}$ .

The function  $m : 2^\Omega \rightarrow [0, 1]$  is said to be the Basic Belief Assignment (bba) on  $2^\Omega$ , if it satisfies

$$\sum_{A \subseteq \Omega} m(A) = 1. \tag{8}$$

Every  $A \in 2^\Omega$  such that  $m(A) > 0$  is called focal element.

The credibility and plausibility functions are defined, respectively, as

$$Bel(A) = \sum_{B \subseteq A, B \neq \emptyset} m(B) \quad \forall A \subseteq \Omega, \tag{9}$$

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \quad \forall A \subseteq \Omega. \tag{10}$$

Each quantity  $Bel(A)$  measures the total support given to  $A$ , while  $Pl(A)$  represents potential amount of support to  $A$ . Functions  $Bel$  and  $Pl$  are linked by the following relation:

$$Pl(A) = 1 - m(\emptyset) - Bel(\bar{A}), \quad (11)$$

where  $\bar{A}$  denotes the complement of  $A$  in  $\Omega$ .

A belief function can be transformed into a probability function by distributing equally each mass of belief  $m(A)$  among the elements of  $A$  [36]. This leads to the concept of Pignistic probability,  $BetP$ , defined by

$$BetP(\omega_i) = \sum_{\omega_i \in A \subseteq \Omega} \frac{m(A)}{|A|(1 - m(\emptyset))}, \quad (12)$$

where  $|A|$  is the cardinality of  $A \subseteq \Omega$ .

### 2.3. Evidential c-means

Assume that there are  $n$  objects  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$  to be clustered in ECM. The set of classes is denoted by  $\Omega = \{\omega_1, \dots, \omega_c\}$ . The uncertain membership of objects can be represented by bbas  $m_i = (m_i(A_k) : A_k \subseteq \Omega)$  ( $i = 1, 2, \dots, n$ ) over the given frame of discernment  $\Omega$ . The optimal credal partition is obtained by minimizing the following objective function:

$$J_{\text{ECM}} = \sum_{i=1}^n \sum_{A_k \subseteq \Omega, A_k \neq \emptyset} |A_k|^\alpha m_i(A_k)^\beta d_{ik}^2 + \sum_{i=1}^n \delta^2 m_i(\emptyset)^\beta \quad (13)$$

subject to

$$\sum_{A_k \subseteq \Omega, A_k \neq \emptyset} m_i(A_k) + m_i(\emptyset) = 1, \quad (14)$$

and

$$m_i(A_k) \geq 0, \quad m_i(\emptyset) \geq 0. \quad (15)$$

Parameters  $\alpha$  and  $\delta$  are adjustable parameters. In Eq. (13),  $d_{ik}$  denotes the distance (generally the Euclidean distance) between  $x_i$  and the barycenter (prototype, denoted by  $\bar{v}_k$ ) associated with  $A_k$ :

$$d_{ik}^2 = \|x_i - \bar{v}_k\|^2, \quad (16)$$

where  $\bar{v}_k$  is defined by

$$\bar{v}_k = \frac{1}{|A_k|} \sum_{h=1}^c s_{hk} v_h, \quad \text{with } s_{hk} = \begin{cases} 1 & \text{if } \omega_h \in A_k \\ 0 & \text{else} \end{cases}. \quad (17)$$

In Eq. (17),  $v_h$  is the geometrical center of points in cluster  $h$ . The updating process with the Euclidean distance alternates updating of the centers and the mass membership using the Lagrange multiplier method. Notation  $m_{ij}$ , which is equivalent to  $m_i(A_j)$ , means the bba of  $\mathbf{x}_i$  given to the nonempty set  $A_k$ , and  $m_{i\emptyset}$ , which is equivalent to  $m_i(\emptyset)$ , is the bba of  $\mathbf{x}_i$  assigned to the empty set. The update rule of mass assignment can be derived as

$$m_{ik} = \frac{|A_k|^{-\alpha/(\beta-1)} d_{ik}^{-2/(\beta-1)}}{\sum_{A_h \neq \emptyset} |A_h|^{-\alpha/(\beta-1)} d_{ih}^{-2/(\beta-1)} + \delta^{-2/(\beta-1)}}, \quad i = 1 \dots n \quad (18)$$

and

$$m_{i\emptyset} = 1 - \sum_{A_k \neq \emptyset} m_{ik}, \quad i = 1, 2, \dots, n. \quad (19)$$

The prototypes (centers) of the classes are given by the rows of matrix  $\mathbf{V}$  (the size is  $c \times p$ ), obtained as the solution of the following linear system:

$$\mathbf{H}\mathbf{V} = \mathbf{B}, \quad (20)$$

where  $\mathbf{H}$  is a matrix of size  $(c \times c)$  given by

$$\mathbf{H}_{lk} = \sum_{i=1}^n \sum_{A_k \supseteq \{\omega_k, \omega_l\}} |A_k|^{\alpha-2} m_{ik}^\beta, \quad (21)$$

and  $\mathbf{B}$  is a matrix of size  $(c \times p)$  defined as

$$\mathbf{B}_{lq} = \sum_{i=1}^n x_{iq} \sum_{A_k \ni \omega_l} |A_k|^{\alpha-1} m_{ik}^\beta. \quad (22)$$

As one of the prototype-based clustering methods, ECM is widely applied in uncertain data applications due to its simplicity and efficiency. As mentioned, if we have some information from a related domain, the available knowledge can help us improve the clustering performance. In the next section, we will introduce a new clustering method taking advantages of both evidential clustering and transfer learning.

### 3. Transfer evidential c-means

Inspired by the idea of evidential clustering and transfer learning, we will introduce the TECM clustering algorithm. As before, denote the  $n$  data samples in the target domain by  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$  and assume that there are  $c_t$  clusters. The set of classes is  $\Omega = \{\omega_1, \dots, \omega_{c_t}\}$ . The available supervised knowledge in a related domain is represented by prototypes  $\mathbf{V}^{(s)} = \{\mathbf{v}_1^{(s)}, \mathbf{v}_2^{(s)}, \dots, \mathbf{v}_{c_s}^{(s)}\}$ . The superscript (s) indicates that the prototypes are from the source domain. TECM assumes that the source and target domains have the same number of clusters ( $c_t = c_s$ ). The objective function of TECM and the optimization approach will be introduced in the following.

#### 3.1. Objective function of TECM

As an evidential clustering method in the framework of belief functions, TECM aims to obtain the optimal credal partition  $\mathbf{M} = (\mathbf{m}_1, \dots, \mathbf{m}_n) \in \mathbb{R}^{n \times 2^{c_t}}$  and the matrix  $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_{c_t})$  of size  $(c_t \times p)$  of cluster centers in the target domain by minimizing

the following objective function:

$$\begin{aligned}
J_{\text{TECM}}(\mathbf{M}, \mathbf{V}) = & \sum_{i=1}^n \sum_{\substack{A_j \subseteq \Omega \\ A_j \neq \emptyset}} c_j^\alpha m_{ij}^\beta d_{ij}^2 + \sum_{i=1}^n \delta^2 m_{i\emptyset}^\beta \\
& + \beta_1 \left[ \sum_{i=1}^n \sum_{\substack{A_j \subseteq \Omega \\ A_j \neq \emptyset}} c_j^\alpha m_{ij}^\beta d_{ij}^{(s)2} + \sum_{i=1}^n \delta^2 m_{i\emptyset}^\beta \right] + \beta_2 \sum_{k=1}^{c_t} \|\mathbf{v}_k^{(s)} - \mathbf{v}_k\|^2, \quad (23)
\end{aligned}$$

subject to

$$\sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij} + m_{i\emptyset} = 1. \quad (24)$$

The distance between samples  $\mathbf{x}_i$  and source domain barycenter is defined by  $d_{ij}^{(s)}$ . The symbol  $c_j$  denotes the cardinality of  $A_j$ . Parameters  $\alpha$ ,  $\beta$  and  $\delta$  control the degree of penalization for imprecise classes with high cardinality, the fuzziness of the partition, and the amount of data considered as outliers respectively. These parameters have the same meaning as those in ECM recalled in Section 2.3.

The objective function in Eq. (23) has four terms. The first two terms are directly inherited from ECM and used to learn from the target data. The third and fourth terms enable the model to learn with the source information, where the knowledge of cluster prototypes is available for the clustering task. Nonnegative parameters  $\beta_1$  and  $\beta_2$  balance the influence of data in the target domain and knowledge in the source domain.

Remark: A transfer evidential  $c$ -means clustering method is proposed by Jiao *et.al.* [37]. It can also leverage the knowledge from the source by qualifying the distance between the prototypes of classes in the source and target domains. The objective function of Jiao’s method is defined as follows:

$$J_{\text{Jiao}} = \sum_{i=1}^n \sum_{\substack{A_{t,j} \subseteq \Omega \\ A_{t,j} \neq \emptyset}} c_j^\alpha m_{ij}^\beta d_{ij}^2 + \sum_{i=1}^n \delta^2 m_{i\emptyset}^\beta + \lambda \sum_{\substack{A_{s,k} \subseteq \Omega \\ A_{s,k} \neq \emptyset}} \sum_{\substack{A_{t,j} \subseteq \Omega \\ A_{t,j} \neq \emptyset}} c_j^\alpha r_{kj}^\gamma \|\tilde{\mathbf{v}}_k - \bar{\mathbf{v}}_j\|^2, \quad (25)$$

where  $A_{t,j}$  and  $A_{s,k}$  denote the classes in the target and source respectively,  $r_{kj}$  denotes the similarity between the barycenter  $\bar{\mathbf{v}}_j$  in the target domain and the barycenter  $\tilde{\mathbf{v}}_k$  in the source domain,  $\lambda$  is a balance coefficient of transfer learning and  $\gamma$  is a weighting exponent. Please see Ref. [37] for more details. As we can see, different from Jiao’s method, in the objective function of TECM, not only the distances between prototypes in the two domains are taken into account, but also the distances between samples in the target domain and prototypes in the source domain are considered. As a result, more source knowledge can be learned by TECM. Besides, in calculating the distance between prototypes in the two domains, we only consider the specific classes (classes with only one focal element such as  $\{\omega_i\}$ ) in the two domains. But in Jiao’s method they consider all the classes in the power set of  $\Omega$ . As the prototypes of imprecise classes are completely determined by specific classes (see Eq. (17)), we consider that the distances between prototypes of specific classes are sufficient to qualify the difference between the two domains in terms of the prototypes. We will compare the behaviors of TECM with Jiao’s method in Section 5.



### 3.2. Optimization

To minimize the objective function  $J_{\text{TECM}}$ , we use the Lagrange multiplier method. First, assume that the prototypes in the target domain are fixed. To solve the constrained minimization problem with respect to the membership matrix  $\mathbf{M}$ ,  $n$  Lagrange multipliers  $\lambda_i (i = 1, \dots, n)$  are introduced and the Lagrangian can be written as

$$L(\mathbf{M}; \lambda_1, \dots, \lambda_n) = J_{\text{TECM}} - \sum_{i=1}^n \lambda_i \left( \sum_{\substack{A_j \subseteq \Omega \\ A_j \neq \emptyset}} m_{ij} + m_{i\emptyset} - 1 \right). \quad (26)$$

Differentiating the Lagrangian with respect to  $m_{ij}$ ,  $m_{i\emptyset}$ , and  $\lambda_i$  and setting the derivatives to zero, the following necessary conditions of optimality for  $\mathbf{M}$  are obtained:

$$\frac{\partial L}{\partial m_{ij}} = c_j^\alpha \beta m_{ij}^{\beta-1} \left( d_{ij}^2 + \beta_1 d_{ij}^{(s)2} \right) - \lambda_i = 0, \quad (27)$$

$$\frac{\partial L}{\partial m_{i\emptyset}} = \beta m_{i\emptyset}^{\beta-1} \left( \delta^2 + \beta_1 \delta^{(s)2} \right) - \lambda_i = 0, \quad (28)$$

$$\frac{\partial L}{\partial \lambda_i} = \sum_{\substack{A_j \subseteq \Omega \\ A_j \neq \emptyset}} m_{ij} + m_{i\emptyset} - 1 = 0. \quad (29)$$

From Eqs. (27) and (28), it is easy to obtain

$$m_{ij} = \left( \frac{\lambda_i}{c_j^\alpha \beta \left( d_{ij}^2 + \beta_1 d_{ij}^{(s)2} \right)} \right)^{1/(\beta-1)}, \quad (30)$$

$$m_{i\emptyset} = \left( \frac{\lambda_i}{\beta \left( \delta^2 + \beta_1 \delta^{(s)2} \right)} \right)^{1/(\beta-1)}. \quad (31)$$

Substituting Eqs. (30) and (31) into Eq. (29), we get

$$\left( \frac{\lambda_i}{\beta} \right)^{1/(\beta-1)} = \frac{1}{\sum_{\substack{A_j \subseteq \Omega \\ A_j \neq \emptyset}} \Delta_{ij} + \left( \frac{1}{\delta^2 + \beta_1 \delta^{(s)2}} \right)^{\frac{1}{\beta-1}}}, \quad (32)$$

where

$$\Delta_{ij} = \left( \frac{1}{c_j^\alpha \left( d_{ij}^2 + \beta_1 d_{ij}^{(s)2} \right)} \right)^{\frac{1}{\beta-1}}. \quad (33)$$

Returning to Eqs. (30) and (31), we get

$$m_{ij} = \frac{\left(1/\left(c_j^\alpha \left(d_{ij}^2 + \beta_1 d_{ij}^{(s)2}\right)\right)\right)^{\frac{1}{\beta-1}}}{\sum_{\substack{A_k \subseteq \Omega \\ A_k \neq \emptyset}} \left(1/\left(c_k^\alpha \left(d_{ik}^2 + \beta_1 d_{ik}^{(s)2}\right)\right)\right)^{\frac{1}{\beta-1}} + \left(\frac{1}{\delta^2 + \beta_1 \delta^2}\right)^{\frac{1}{\beta-1}}}, \quad (34)$$

and

$$m_{i\emptyset} = \frac{\left(\frac{1}{\delta^2 + \beta_1 \delta^2}\right)^{\frac{1}{\beta-1}}}{\sum_{\substack{A_k \subseteq \Omega \\ A_k \neq \emptyset}} \left(1/\left(c_k^\alpha \left(d_{ik}^2 + \beta_1 d_{ik}^{(s)2}\right)\right)\right)^{\frac{1}{\beta-1}} + \left(\frac{1}{\delta^2 + \beta_1 \delta^2}\right)^{\frac{1}{\beta-1}}}. \quad (35)$$

Next, we consider that the credal membership matrix  $\mathbf{M}$  is fixed. Then, the minimization of  $J_{\text{TECM}}$  with respect to  $\mathbf{V}$  is an unconstrained optimization problem. The partial derivatives of  $J_{\text{TECM}}$  with respect to the prototypes of the specific classes in the target domain are

$$\frac{\partial J_{\text{TECM}}}{\partial \mathbf{v}_l} = \sum_{i=1}^n \sum_{\substack{A_j \subseteq \Omega \\ A_j \neq \emptyset}} c_j^\alpha m_{ij}^\beta \frac{\partial d_{ij}^2}{\partial \mathbf{v}_l} - 2\beta_2 (\mathbf{v}_l^{(s)} - \mathbf{v}_l), \quad (36)$$

where

$$\frac{\partial d_{ij}^2}{\partial \mathbf{v}_l} = 2(\mathbf{x}_i - \bar{\mathbf{v}}_j) \left(-s_{lj} \frac{1}{c_j}\right), \quad (37)$$

and  $\bar{\mathbf{v}}_j$  is defined by Eq. (17). Thus, we have

$$\frac{\partial J_{\text{TECM}}}{\partial \mathbf{v}_l} = -2 \sum_{i=1}^n \sum_{\substack{A_j \subseteq \Omega \\ A_j \neq \emptyset}} c_j^{\alpha-1} m_{ij}^\beta s_{lj} \left(\mathbf{x}_i - \frac{\sum_{k=1}^{c_t} s_{kj} \mathbf{v}_k}{c_j}\right) - 2\beta_2 (\mathbf{v}_l^{(s)} - \mathbf{v}_l). \quad (38)$$

Setting these derivatives to zero, we can get  $l$  linear equations of  $\mathbf{v}_k$ :

$$\sum_i \mathbf{x}_i \sum_{\substack{A_j \subseteq \Omega \\ A_j \neq \emptyset}} c_j^{\alpha-1} m_{ij}^\beta s_{lj} = \sum_{k=1}^{c_t} \mathbf{v}_k \sum_{i=1}^n \sum_{\substack{A_j \subseteq \Omega \\ A_j \neq \emptyset}} c_j^{\alpha-2} m_{ij}^\beta s_{kj} s_{lj} - \beta_2 (\mathbf{v}_l^{(s)} - \mathbf{v}_l). \quad (39)$$

Let  $\mathbf{B}$  be the matrix of size  $(c_t \times p)$  defined by

$$\mathbf{B}_{lq} = \sum_{i=1}^n x_{iq} \sum_{\substack{A_j \subseteq \Omega \\ A_j \neq \emptyset}} c_j^{\alpha-1} m_{ij}^\beta s_{lj} = \sum_{i=1}^n x_{iq} \sum_{A_j \ni \omega_l} c_j^{\alpha-1} m_{ij}^\beta, \quad (40)$$

and let  $\mathbf{H}$  be the matrix of size  $(c_t \times c_t)$  given by

$$\mathbf{H}_{lk} = \sum_{i=1}^n \sum_{\substack{A_j \subseteq \Omega \\ A_j \neq \emptyset}} c_j^{\alpha-2} m_{ij}^\beta s_{lj} s_{kj} = \sum_i \sum_{A_j \ni \{\omega_k, \omega_l\}} c_j^{\alpha-2} m_{ij}^\beta. \quad (41)$$

Let  $\mathbf{I}$  be the  $(c_t \times c_t)$  identity matrix. The prototype matrix  $\mathbf{V}$  can be obtained by solving the following linear system

$$(\mathbf{H} + \beta_2 \mathbf{I}) \mathbf{V} = \mathbf{B} + \beta_2 \mathbf{V}^{(s)}. \quad (42)$$

#### 4. Extended transfer evidential $c$ -means

As will be shown in Section 5, the TECM clustering algorithm can effectively make use of the knowledge in the source domain and improve the performance of the clustering algorithm in the target domain, but it assumes that the source and target domains have the same number of clusters. However, in many practical applications, this assumption cannot always hold. In this section, an Extended version of Transfer Evidential  $C$ -Means (ETECM) is proposed for the clustering task with different numbers of clusters in the two domains.

##### 4.1. The objective function of ETECM

Let  $\mathbf{R}$  be a matrix of size  $(c_s \times c_t)$  with elements  $r_{ij}$ , which denotes the similarity between the  $i$ -th cluster in the source domain and the  $j$ -th cluster in the target domain. ETECM aims to get the optimal partition by updating the membership  $\mathbf{M}$ , the prototypes  $\mathbf{V}$  and the similarity matrix  $\mathbf{R}$  respectively. The objective function of ETECM is defined as

$$J_{\text{ETECM}}(\mathbf{M}, \mathbf{V}, \mathbf{R}) = \sum_{i=1}^n \sum_{\substack{A_j \subseteq \Omega \\ A_j \neq \emptyset}} c_j^\alpha m_{ij}^\beta d_{ij}^2 + \sum_{i=1}^n \delta^2 m_{i\emptyset}^\beta + \beta_3 \sum_{i=1}^{c_s} \sum_{j=1}^{c_t} r_{ij}^\eta \left\| \mathbf{v}_i^{(s)} - \mathbf{v}_j \right\|^2, \quad (43)$$

subject to

$$\sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij} + m_{i\emptyset} = 1, \quad (44)$$

and

$$\sum_{j=1}^{c_t} r_{ij} = 1. \quad (45)$$

As we can see, the first two terms are directly inherited from ECM, and the third term is designed to utilize the knowledge from the source domain. It should be noticed that, ETECM is not equivalent to TECM when the source and target domains have the same number of clusters. TECM uses two terms to transfer knowledge from the source domain, but ETECM only adopts one.

##### 4.2. Optimization

From the objective function of ETECM, we can see that  $\mathbf{M}$  is only related to the first two terms. Thus, the update rule for  $\mathbf{M}$  is the same as that in ECM:

$$m_{ij} = \frac{c_j^{-\alpha/(\beta-1)} d_{ij}^{-2/(\beta-1)}}{\sum_{A_k \neq \emptyset} c_k^{-\alpha/(\beta-1)} d_{ik}^{-2/(\beta-1)} + \delta^{-2/(\beta-1)}}, \quad i = 1, 2, \dots, n; j | A_j \subseteq \Omega, A_j \neq \emptyset, \quad (46)$$

$$m_{i\emptyset} = 1 - \sum_{A_j \neq \emptyset} m_{ij}, \quad i = 1, 2, \dots, n. \quad (47)$$

To update the cluster centers in the target domain  $\mathbf{v}_l$ , we let

$$\begin{aligned} \frac{\partial J_{\text{ETECM}}}{\partial \mathbf{v}_l} = & -2 \sum_{i=1}^n \sum_{\substack{A_j \subseteq \Omega \\ A_j \neq \emptyset}} \left\{ c_j^{\alpha-1} m_{ij}^\beta s_{lj} \times \left( \mathbf{x}_i - \frac{1}{c_j} \sum_{k=1}^{c_t} s_{kj} \mathbf{v}_k \right) \right\} \\ & - 2\beta_3 \sum_{i=1}^{c_s} r_{il}^\eta \left( \mathbf{v}_i^{(s)} - \mathbf{v}_l \right) = 0, \quad l = 1, \dots, c_t, \end{aligned} \quad (48)$$

from which we get

$$\sum_i \mathbf{x}_i \sum_{\substack{A_j \subseteq \Omega \\ A_j \neq \emptyset}} c_j^{\alpha-1} m_{ij}^\beta s_{lj} = \sum_{k=1}^{c_t} \mathbf{v}_k \sum_{i=1}^n \sum_{\substack{A_j \subseteq \Omega \\ A_j \neq \emptyset}} (c_j^{\alpha-2} \times m_{ij}^\beta s_{kj} s_{lj}) - \beta_3 \sum_{i=1}^{c_s} r_{il}^\eta \left( \mathbf{v}_i^{(s)} - \mathbf{v}_l \right). \quad (49)$$

Let  $\bar{\mathbf{R}}$  be a  $(c_t \times c_t)$  diagonal matrix with diagonal elements

$$r_l = \sum_{i=1}^{c_s} r_{il}, \quad l = 1, 2, \dots, c_t. \quad (50)$$

The prototype matrix in the target domain  $\mathbf{V}$  is obtained by solving the following linear system:

$$(\mathbf{H} + \beta_3 \bar{\mathbf{R}}) \mathbf{V} = \mathbf{B} + \beta_3 (\mathbf{R}^\eta)^\top \mathbf{V}^{(s)}, \quad (51)$$

where  $\mathbf{B}$  and  $\mathbf{H}$  are defined in Eqs. (40) and (41) respectively, and  $A^\top$  denotes the transpose of matrix  $A$ .

The update rule for  $r_{ij}$  can be obtained, which is similar to that of  $m_{ij}$  by introducing Lagrangian multipliers:

$$r_{ij} = \frac{(d_{ij}[st])^{-\frac{1}{\eta-1}}}{\sum_{j=1}^{c_t} (d_{ij}[st])^{-\frac{1}{\eta-1}}}, \quad i = 1, \dots, c_s; \quad j = 1, \dots, c_t, \quad (52)$$

where

$$d_{ij}[st] = \left\| \mathbf{v}_i^{(s)} - \mathbf{v}_j \right\|^2. \quad (53)$$

### 4.3. Computational complexity

The cluster membership of each object is distributed on the power set  $2^\Omega$  for credal partitions. Consequently, the number of parameters to be optimized in TECM and ETECM are linear in the number of objects and exponential in the number of clusters. TECM and ETECM both have the computational complexity of  $O(n2^{c_t})$ , where  $n$  is the size of data set in the target domain and  $c_t$  is the number of clusters in the target domain. In practice, it is possible to reduce the complexity of proposed methods by constraining the focal sets to be composed of singletons, the empty set and  $\Omega$ . In this way, the computational complexity can be simplified to  $O(2n + nc_t)$  in both the TECM and the ETECM.

## 5. Experiments

In this section, we will conduct some experiments on synthetic and real-world data sets to show the performance of the proposed methods. The Adjusted Rand Index (ARI) and the Normalized Mutual Information (NMI), which are commonly used in evaluating clustering methods, are adopted as the performance indexes. As ARI and NMI are used to measure the quality of hard partitions, when evaluating the partitions provided by ECM, TECM and ETECM, we use the corresponding hard partitions by assigning samples to the cluster with maximal Pignistic probability [36].

Besides, we use the evidential precision (EP) proposed in [38, 39] to show the advantages of imprecise clusters in credal partitions. EP is defined as

$$\text{EP} = \frac{n_{er}}{N_e}, \quad (54)$$

where  $N_e$  denotes the number of pairs partitioned into the same specific group by evidential clustering, and  $n_{er}$  is the number of relevant instance pairs out of these specifically clustered pairs. For hard partitions, EP boils down to the classical Precision (P) since there are no imprecise clusters. Precision is the fraction of relevant instances (pairs in identical groups in the clustering benchmark) out of those retrieved instances (pairs in identical groups of the discovered clusters), calculated by

$$\text{P} = \frac{a}{a + c}, \quad (55)$$

where  $a$  is the number of pairs of objects simultaneously assigned to identical classes by the reference partition and the obtained one, and  $c$  is the number of dissimilar pairs partitioned into the same cluster. The codes are available on GitHub<sup>2</sup>.

### 5.1. Gaussian data sets

Gaussian data sets were used to show how to determine the appropriate parameters of TECM. The source and target data both followed two-dimensional Gaussian distributions with parameters (mean values and covariance matrices) listed in Tables 1-2.

Table 1: Distributions of the source data.

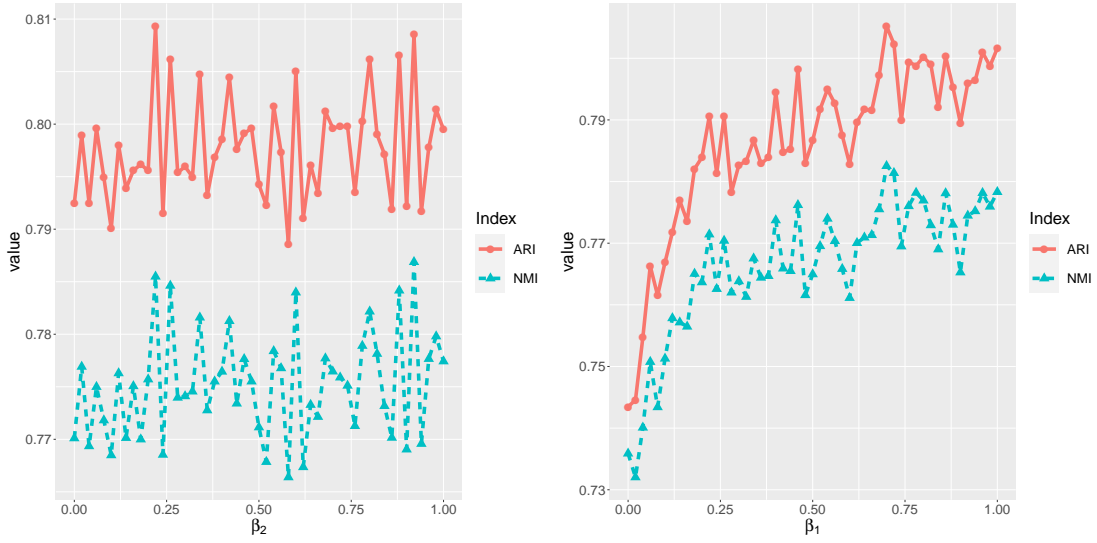
	ID	S1	S2	S3	S4
parameter					
Mean		[2, 4]	[9, 15]	[8, 30]	[13, 40]
Covariance		$\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$	$\begin{bmatrix} 25 & 0 \\ 0 & 7 \end{bmatrix}$	$\begin{bmatrix} 30 & 0 \\ 0 & 20 \end{bmatrix}$	$\begin{bmatrix} 15 & 0 \\ 0 & 10 \end{bmatrix}$

There were four alternative clusters in the source data and three clusters in the target. Notations  $n_t$  and  $n_s$  denote the numbers of data samples in the target and source domain respectively.

<sup>2</sup><https://github.com/kuangzhou/TECM>

Table 2: Distributions of the target data.

parameter \ ID	T1	T2	T3
	Mean	[3, 4]	[10.5, 12.5]
Covariance	$\begin{bmatrix} 10 & 0 \\ 0 & 11 \end{bmatrix}$	$\begin{bmatrix} 25 & 0 \\ 0 & 7 \end{bmatrix}$	$\begin{bmatrix} 30 & 0 \\ 0 & 10.9 \end{bmatrix}$



a.  $\beta_2$  is changed from 0 to 1 with  $\beta_1 = 1$

b.  $\beta_1$  is changed from 0 to 1 with  $\beta_2 = 1$

Figure 2: ARI and NMI of the clustering results on Gaussian data with changing  $\beta_1$  and  $\beta_2$ .

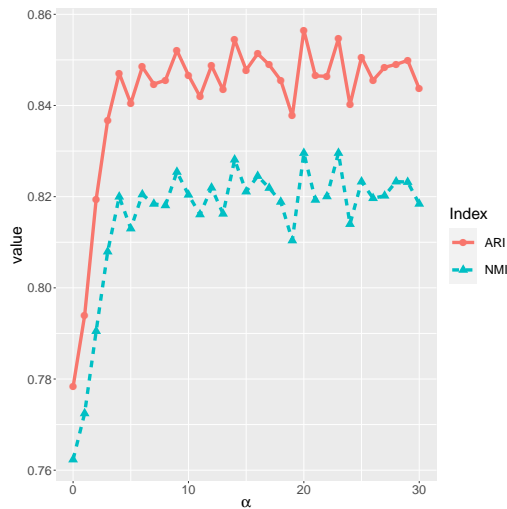


Figure 3: ARI and NMI of the clustering results on Gaussian data with changing  $\alpha$ .

Table 3: ARI by TECM on Gaussian data sets under different parameter settings.

ARI \ $\beta_2$											
$\beta_1$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0	0.729	0.726	0.734	0.730	0.732	0.727	0.735	0.73	0.736	0.742	0.745
0.1	0.731	0.739	0.749	0.769	0.764	0.758	0.768	0.764	0.767	0.778	0.767
0.2	0.767	0.776	0.770	0.782	0.772	0.770	0.771	0.777	0.776	0.775	0.784
0.3	0.773	0.774	0.773	0.775	0.782	0.773	0.78	0.786	0.774	0.772	0.785
0.4	0.778	0.783	0.787	0.794	0.790	0.785	0.787	0.787	0.784	0.785	0.786
0.5	0.782	0.779	0.775	0.790	0.783	0.784	0.783	0.79	0.788	0.793	0.793
0.6	0.785	0.776	0.796	0.780	0.798	0.791	0.793	0.792	0.796	0.795	0.787
0.7	0.801	0.798	0.782	0.797	0.787	0.798	0.789	0.793	0.787	0.783	0.789
0.8	0.792	0.804	0.792	0.794	0.802	0.801	0.801	0.799	0.796	0.788	0.797
0.9	0.792	0.798	0.799	0.783	0.801	0.796	0.803	0.793	0.798	0.789	0.799
1.0	0.799	0.789	0.794	0.790	0.802	0.801	0.799	0.793	0.794	0.795	<b>0.804</b>

Table 4: NMI by TECM on Gaussian data sets under different parameter settings.

NMI \ $\beta_2$											
$\beta_1$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0	0.722	0.721	0.729	0.723	0.727	0.721	0.731	0.725	0.731	0.735	0.734
0.1	0.721	0.730	0.734	0.753	0.750	0.743	0.750	0.748	0.754	0.760	0.749
0.2	0.748	0.761	0.752	0.763	0.757	0.752	0.756	0.756	0.758	0.757	0.766
0.3	0.757	0.758	0.755	0.756	0.764	0.754	0.758	0.768	0.757	0.753	0.765
0.4	0.759	0.763	0.766	0.774	0.770	0.767	0.768	0.766	0.763	0.765	0.767
0.5	0.765	0.759	0.756	0.768	0.764	0.766	0.761	0.770	0.768	0.773	0.772
0.6	0.766	0.757	0.776	0.757	0.776	0.770	0.772	0.770	0.773	0.773	0.766
0.7	0.778	0.775	0.759	0.777	0.764	0.775	0.768	0.770	0.768	0.763	0.768
0.8	0.771	0.781	0.770	0.774	0.780	0.778	0.778	0.775	0.773	0.770	0.774
0.9	0.772	0.778	0.776	0.761	0.780	0.775	0.781	0.771	0.773	0.766	0.776
1.0	0.776	0.765	0.772	0.767	0.779	0.780	0.776	0.770	0.774	0.772	<b>0.782</b>

The influence of  $\beta_1$  and  $\beta_2$  was tested during the experiment. The source data set was synthetized by Gaussian distribution with parameters S1, S2, and S3. The target data set was generated by Gaussian distribution with parameters T1, T2, and T3. Set  $n_t = 50$  and  $n_s = 500$ . The experiment was repeated 100 times by randomly generating Gaussian data with fixed parameters. The average values of ARI and NMI are reported in Table 3 and Table 4 respectively. When parameters  $\beta_1$  and  $\beta_2$  are both set to 1, TECM yields the best clustering results. Then we increased  $\beta_2$  gradually from 0 to 1 (with  $\beta_1 = 1$ ) and increased  $\beta_1$  gradually from 0 to 1 (with  $\beta_2 = 1$ ) respectively. The mean values of ARI and NMI are shown in Fig. 2. We can find that  $\beta_2$  has little impact on clustering results when  $\beta_1$  is set to 1. On the contrary,  $\beta_1$  have a great influence on the clustering results when  $\beta_2$  is fixed to 1. When  $\beta_1$  increases, clustering results tend to be better. Then, we set  $\beta_1 = 1$  and  $\beta_2 = 1$ , and increased  $\alpha$  gradually from 0 to 30. Fig. 3 shows that NMI and ARI are growing when  $\alpha$  is increasing from 0 to 4, but become stable when the  $\alpha$  is larger than 4. Thus, in the following experiments, we set  $\beta_1 = \beta_2 = 1$  and  $\alpha = 4$  as the default parameter values.

In additional, the sensitivity of initial prototypes was tested. TECM and ETECM were run with five different initial prototype sets, which were randomly selected from target data samples. Other parameters of TECM and ETECM used in this experiment are illustrated in the first two columns in Table 5. The final prototypes as well as the ARI, NMI, and EP of TECM and ETECM are shown in Table 6. It can be seen that different initial prototypes lead to similar results. In other word, TECM and ETECM are robust to the setting of initial prototypes. Thus, the initial prototypes will be randomly selected from the target data set in

Table 5: Parameter settings in TECM, ETECM, TFCM, E-TFCM, ECM, and Jiao’s method.

TECM	ETECM	TFCM	E-TFCM	Jiao	ECM
$\beta_1 = \beta_2 = 1$	$\alpha = 4, \beta = 2$	$\lambda_1 = 0.1$	$m_1 = 3$	$\alpha = 4, \delta = 10$	$\alpha = 4$
$\alpha = 4, \beta = 2$	$\beta_3 = 1, \eta = 2$	$\lambda_2 = 0.5$	$m_2 = 3$	$\beta = 2, \lambda = 1$	$\delta = 10$
$\delta = 10$	$\delta = 10$	$m=2$	$\lambda_1 = 1$	$\gamma = 2$	$\beta = 2$

the following experiments.

Table 6: The final prototypes, ARI, NMI, and EP obtained by TECM and ETECM with five different groups of initial prototypes.

	Initial prototypes	TECM				ETECM			
		Prototypes	ARI	NMI	EP	Prototypes	ARI	NMI	EP
1	(7.45, 19.15)	(2.54144, 4.19656)				(8.28819, 29.19526)			
	(11.82, 7.64)	(10.26608, 13.46417)	0.84	0.80	0.92	(11.27092, 12.66844)	0.84	0.80	0.91
	(7.37, 3.56)	(8.19675, 29.61525)				(2.66649, 4.24254)			
2	(1.37, 12.50)	(2.54137, 4.19646)				(11.27092, 12.66844)			
	(2.28, 4.43)	(10.26580, 13.46404)	0.84	0.80	0.92	(2.66649, 4.24254)	0.84	0.80	0.91
	(6.49, 32.24)	(8.19683, 29.61524)				(8.28819, 29.19526)			
3	(17.64, 14.99)	(2.54147, 4.19661)				(11.27180, 12.66883)			
	(11.93, 13.73)	(10.26611, 13.46428)	0.84	0.80	0.92	(2.66677, 4.24298)	0.84	0.80	0.91
	(7.60, 33.84)	(8.19675, 29.61531)				(8.28793, 29.19530)			
4	(2.43, 0.78)	(2.54145, 4.19659)				(2.66648, 4.24252)			
	(6.90, 26.97)	(10.26609, 13.46425)	0.84	0.80	0.92	(8.28820, 29.19526)	0.84	0.80	0.91
	(6.79, 9.02)	(8.19673, 29.61530)				(11.27087, 12.66842)			
5	(9.24, 15.06)	(2.54138, 4.19647)				(8.28821, 29.19526)			
	(0.08, -4.56)	(10.26587, 13.46404)	0.84	0.80	0.92	(2.66647, 4.24251)	0.84	0.80	0.91
	(-1.48, 4.35)	(8.19683, 29.61522)				(11.27086, 12.66842)			

As mentioned, TECM is only suitable for the case where the cluster numbers of the two domains are identical. In contrast, ETECM is able to handle transfer learning tasks in which the source and target domains have different cluster numbers. The application scope will be illustrated in detail in the following experiments. The parameters of these two clustering methods were set the same as before. Two scenarios for transfer learning were considered. The first scenario was that the data in the source and target domains had the same cluster numbers, and the second scenario was that cluster numbers in the two domains were different.

In the first scenario, Gaussian data sets were generated with parameters S1-S3 (T1-T3) as the source domain (target domain). We compared the proposed clustering methods (TECM and ETECM) with three existing algorithms: TFCM, ECM and Jiao’s method. We increased  $n_t$  gradually from 10 to 500. For each  $n_t$ , Gaussian data were generated 100 times under the fixed parameters of S1-S3 and T1-T3. The prototypes of clusters in the source domain were provided by the mean values shown in Table 1. The average values of ARI, NMI and EP are reported in Fig. 4.

From Fig. 4, TECM performs better than ETECM. The objective function of TECM takes the distances between samples and source domain prototypes into account, but the objective function of ETECM does not. Thus, TECM makes greater use of the source domain information. In addition, the clustering results obtained by TECM are better than those



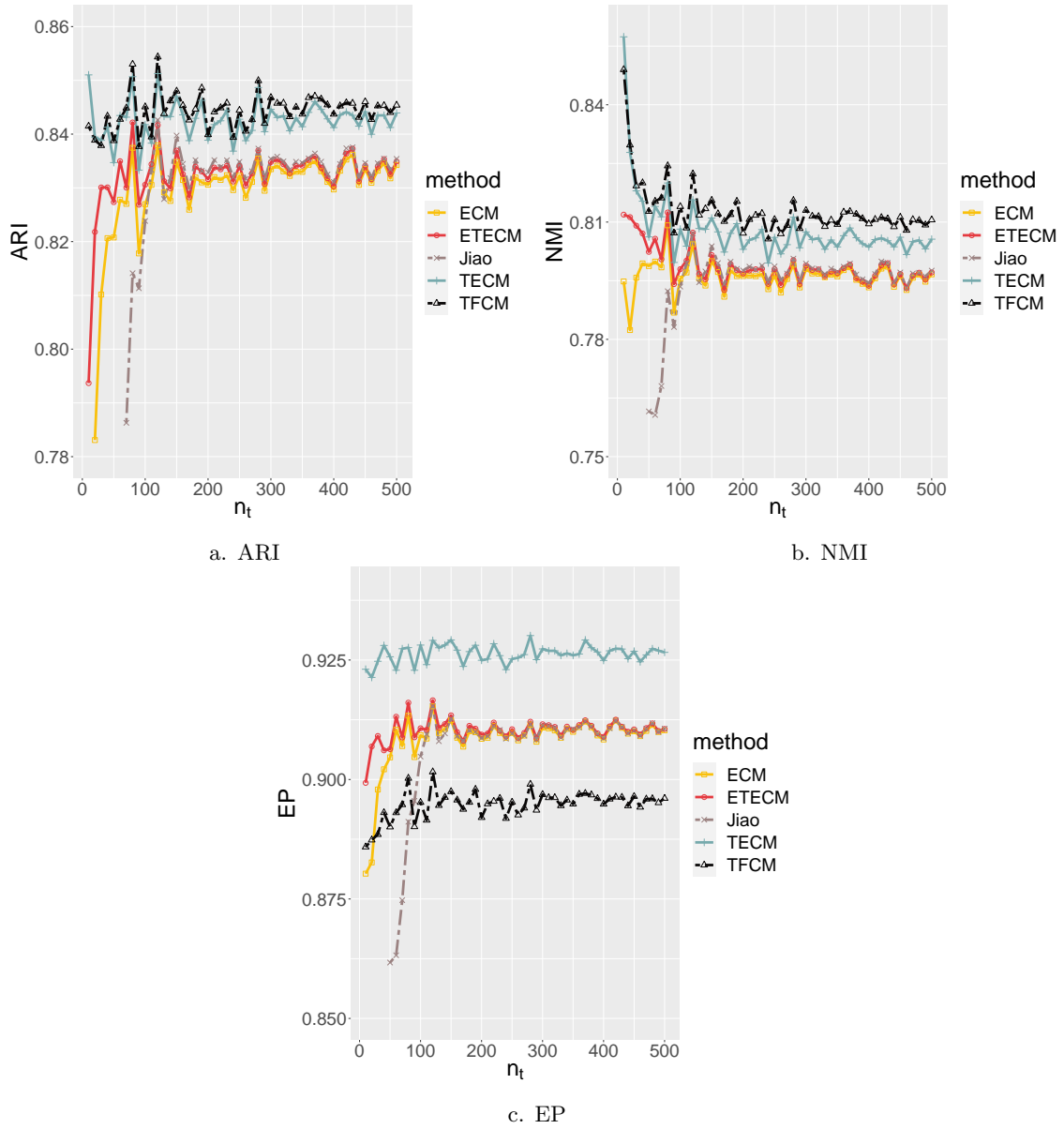


Figure 4: ARI, NMI, and EP of the clustering results on Gaussian data.

by ECM in terms of ARI, NMI, and EP. This demonstrates the effectiveness of transfer learning in this scenario. The results obtained by TECM are better than TFCM as indicated by the EP. Compared with TFCM, TECM introduces the theory of belief function to well express uncertain information in the cluster structure. We can also find that TECM performs significantly better than Jiao’s method when  $n_t$  is small. This is because TECM considers the distances between source domain prototypes and target domain samples. Consequently, it can learn more knowledge from the source domain.

In the second scenario, Gaussian data sets were generated with S1-S4 as the source domain and T1-T3 as the target domain. In this case, there were four clusters in the source domain while three in the target. We compared ETECM with E-TFCM conducted on above Gaussian sets. As Fig. 5 shows, the clustering results obtained by ETECM are close to those of E-TFCM in terms of both ARI and NMI when the target domain has a small number of samples, but better than E-TFCM in terms of EP. It demonstrates the effectiveness of ETECM in describing the uncertainty information among clusters.

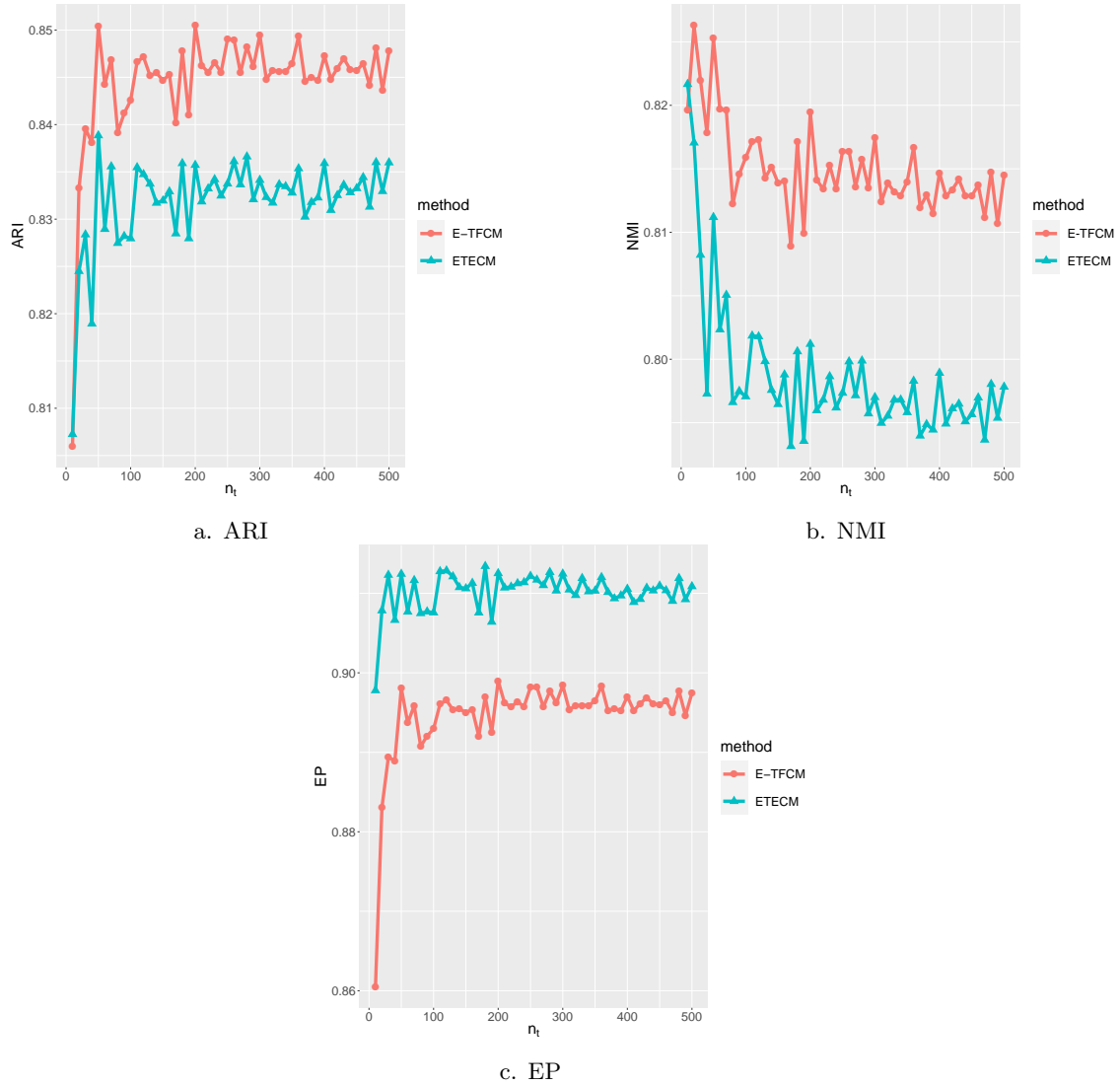


Figure 5: ARI, NMI, and EP for ETECM and E-TFCM clustering results on Gaussian data sets with 4 clusters in the source domain and 3 clusters in the target domain.

## 5.2. Iris data set

This subsection is designed to verify the effectiveness of the available prototypes in the source domain on the clustering performance. We consider the Iris data set consisting of 150 samples from three species. Four features were measured for each sample: Sepal.Length (SL), Sepal.Width (SW), Petal.Length (PL) and Petal.Width (PW). The four features were divided into two parts  $FT1$  and  $FT2$ . The six cases are listed in Table 7. The samples with features in  $FT1$  were regarded as the target data to be clustered.

Table 7: The feature division for Iris data.

Case	$FT1$	$FT2$	Case	$FT1$	$FT2$
Case 1	SL, SW	PL, PW	Case 4	SW, PL	SL, PW
Case 2	SL, PL	SW, PW	Case 5	SW, PW	SL, PL
Case 3	SL, PW	SW, PL	Case 6	PL, PW	SL, SW

In order to generate the prototypes in the source domain which were required before using TECM and ETECM, for each case, we first applied  $c$ -means clustering on the samples with features in  $FT2$  and got the best hard partition for the 150 samples. Then the following two schemes were designed to get the prototypes  $\mathbf{v}_k^{(s)}$ :

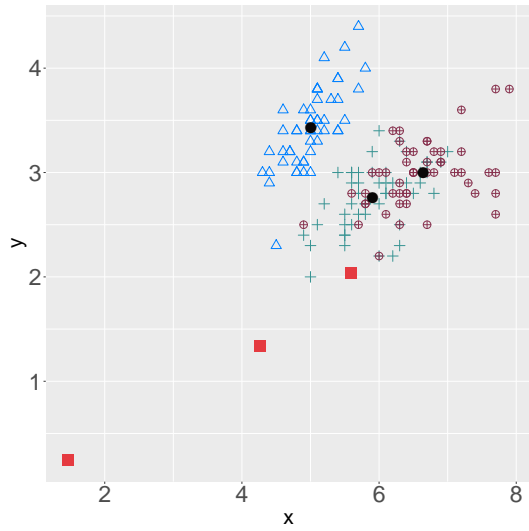
**Scheme A:** By the feature mean of samples in each group with feature set  $FT2$ ;

**Scheme B:** By the feature mean of samples in each group with feature set  $FT1$ ;

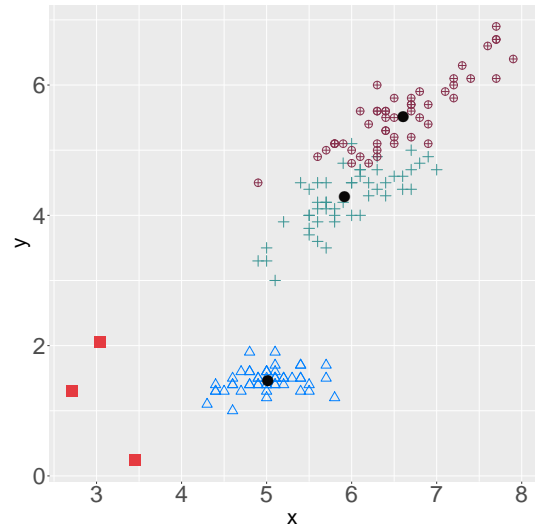
In Scheme B, the prototypes were from the target data (with  $FT1$ ) based on a clustering rule learned with the source domain (with  $FT2$ ), while in Scheme A the prototypes were from the source data (with  $FT2$ ) based on a clustering rule learned with the same domain (with  $FT2$ ). TECM with these two schemes were termed by TECM-A and TECM-B respectively. Analogously, ETECM with the two schemes were termed by ETECM-A and ETECM-B. We also designed two cases using ECM: ECM-Target and ECM-Emerge. The former denotes the ECM algorithm conducted on the target data, while the latter represents ECM conducted on the data set directly combining samples in the target and source domain. The ARI and NMI for the results by ECM-Target, ECM-Emerge, TECM-A, TECM-B, ETECM-A, and ETECM-B are displayed in Table 8. From the table we can see

- TECM-B performs better than ECM-Target in Cases 1 and 3, but the performance of the two methods is similar in Cases 2, 4, 5 and 6.
- TECM-A performs worse than TECM-B and ECM-Target in all cases.
- ECM-Emerge performs worse than ECM-Target in all cases.
- ETECM-B performs better than ECM-Emerge and TECM-B in all cases.

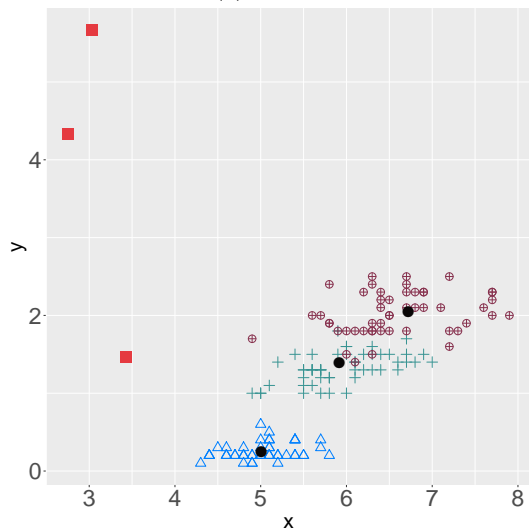
In TECM-B, the transferred knowledge of prototypes has the same feature set as the target samples (this corresponds to the illustrative example in the introduction). The results show that the clustering performance is indeed improved by the use of information from the



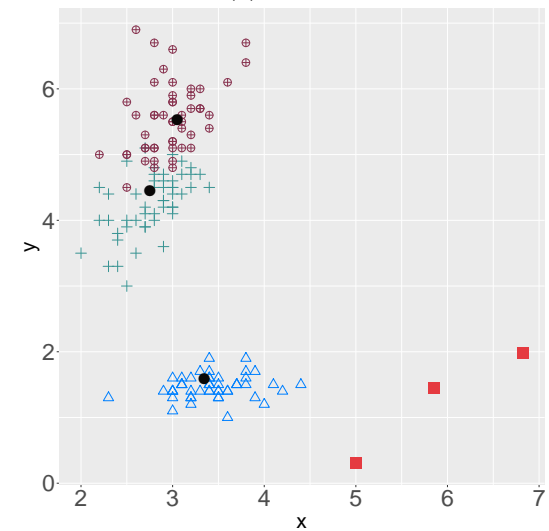
(a) Case 1



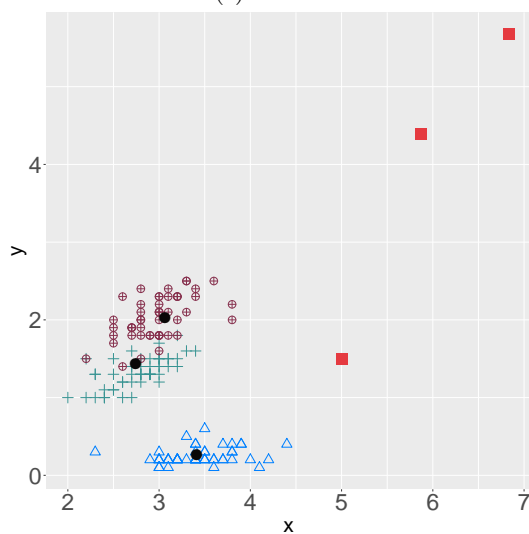
(b) Case 2



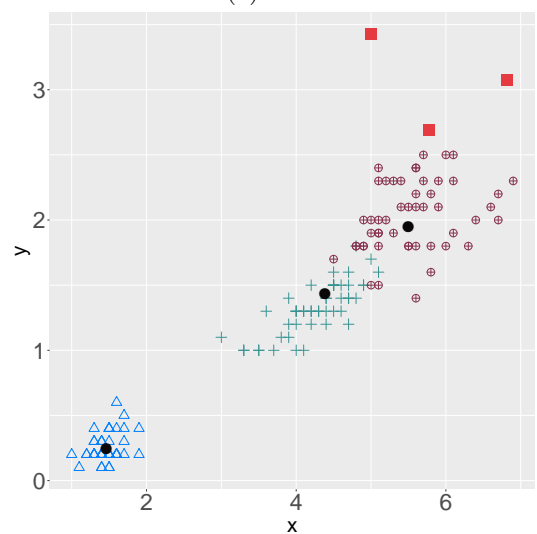
(c) Case 3



(d) Case 4



(e) Case 5



(f) Case 6

Figure 6: The FT1 and prototype  $v_k^{(s)}$  in different cases. In the above figures, the blue  $\Delta$  represents setosa; the green  $+$  represents versicolor; the brown  $\oplus$  represents virginica; the red filled squares represent the source prototypes of Scheme A and the black filled circles represent the source prototypes of Scheme B.

source. In contrast,  $\mathbf{v}_k^{(s)}$  generated from TECM-A is far away from target data sets in all cases (Fig. 6). The knowledge from the source domain has a negative influence on the performance of transfer clustering in this situation. We can deduce that the performance degradation of TECM-A may be caused by the imperfect information matching between the two domains. Although ETECM-A adopts the same source prototypes as TECM-A, it can reduce the risk of negative transfer clustering and get the best clustering results in most cases.

Table 8: ARI and NMI for the clustering results on Iris set by TECM, ETECM and ECM.

Method \ Case	ARI						NMI					
	1	2	3	4	5	6	1	2	3	4	5	6
ECM-Target	0.56	0.69	0.58	0.80	0.79	0.78	0.58	0.70	0.60	<b>0.83</b>	0.78	0.78
ECM-Emerge	0.16	0.56	0.30	0.57	0.00	0.43	0.15	0.55	0.28	0.58	0.02	0.48
TECM-A	0.01	0.57	0.00	0.55	0.00	0.51	0.02	0.58	0.00	0.57	0.00	0.49
ETECM-A	0.42	<b>0.74</b>	0.46	<b>0.83</b>	<b>0.80</b>	<b>0.81</b>	0.38	<b>0.73</b>	0.42	<b>0.83</b>	0.77	<b>0.79</b>
TECM-B	<b>0.57</b>	0.68	<b>0.61</b>	0.80	0.79	0.78	<b>0.59</b>	0.68	<b>0.62</b>	0.80	0.78	0.75
ETECM-B	0.44	0.70	0.59	0.82	<b>0.80</b>	0.78	0.46	0.70	0.61	0.82	<b>0.79</b>	0.78

Table 9: ARI and NMI for the clustering results on TCA-processing Iris set by TECM, ETECM and ECM.

Method \ Case	ARI						NMI					
	1	2	3	4	5	6	1	2	3	4	5	6
ECM-Target	<b>0.35</b>	<b>0.79</b>	0.68	0.71	0.80	0.40	<b>0.41</b>	<b>0.80</b>	0.70	0.71	0.79	0.49
ECM-Emerge	<b>0.35</b>	<b>0.79</b>	0.68	0.71	0.80	0.40	<b>0.41</b>	<b>0.80</b>	0.70	0.71	0.79	0.49
TECM-A	<b>0.35</b>	0.74	<b>0.70</b>	<b>0.74</b>	<b>0.82</b>	<b>0.47</b>	<b>0.41</b>	0.77	<b>0.71</b>	<b>0.72</b>	<b>0.80</b>	<b>0.53</b>
ETECM-A	<b>0.35</b>	<b>0.79</b>	0.68	0.57	0.79	0.44	<b>0.41</b>	<b>0.80</b>	0.70	0.61	0.78	0.50
TECM-B	<b>0.35</b>	0.74	<b>0.70</b>	<b>0.74</b>	<b>0.82</b>	<b>0.47</b>	<b>0.41</b>	0.77	<b>0.71</b>	<b>0.72</b>	<b>0.80</b>	<b>0.53</b>
ETECM-B	<b>0.35</b>	<b>0.79</b>	0.47	0.71	0.80	0.41	<b>0.41</b>	<b>0.80</b>	0.46	0.71	0.79	0.49

From the results of TECM-A and TECM-B in Table 8, the performance of TECM degrades when source domain and target domain distributions are greatly different. To improve the performance of TECM in Scheme A, the TCA algorithm [40] was used to reduce the distribution discrepancy first. Table 9 shows ARI and NMI values of the results by ECM-Target, ECM-Emerge, TECM-A, TECM-B, ETECM-A, ETECM-B with the data processed by TCA. From the table, we can find that, in this way, TECM-A makes great progress in Cases 2-5. TECM-A gains the best clustering results than any other methods in most of cases. In summary, TECM is recommended when the distribution of source and target domain are close. When the distribution of source and target domain are greatly different, we suggest using ETECM, or using TCA first before TECM to make the data distributions in the two domains close.

### 5.3. Texture segmentation

In this subsection, clustering methods were conducted on texture segmentation. The Brodatz texture dataset<sup>3</sup> was utilized to verify the effectiveness of the proposed methods. Five basic textures in this data set including D1, D16, D35, D43, and D49 were used to synthesize the source or target domain data.

Firstly, the artificial scenario was constructed for transfer learning, as shown in Fig. 7. All these synthetic textures were resized to 45×45 pixels. It is noted that Fig. 7a is regarded

<sup>3</sup>T. Randen, Brodatz Texture, <https://www.ux.uio.no/tranden/brodatz.html>.

as the source domain for all tasks. In this experiment, target pictures with different cluster numbers were designed. The target data sets T4-T6 had 3, 4, and 5 clusters respectively. Some Gaussian noise with different standard deviations were added to these images. In Fig. 7, T4-1, T4-2, and T4-3 denote the data sets generated by T4 adding Gaussian noise with  $\delta = 0.05, 0.1, 0.2$  respectively. Similarly, T5-1 to T5-3 and T6-1 to T6-3 were designed. The Gabor filter [41] was used to extract texture features of all target domain pictures in terms of the filtering banks with 6 orientations (at every  $30^\circ$ ) and 5 frequencies (0.46, 0.47, 0.48, 0.49 and 0.50). The ideal segmentation results of all target images are shown in Fig. 8, where the small squares from the same textures should be grouped together. Table 10 shows the ARI, NMI, and EP of the obtained segmentation. From the table, we can see that TECM performs better in most cases when the target and source domains are of the same number of clusters.

Table 10: ARI, NMI and EP calculated by clustering methods on the different texture images.

Target image	Index	Clustering methods					
		ECM	TFCM	E-TFCM	Jiao	TECM	ETECM
T4-1	ARI	<b>0.516</b>	-	0.419	0.492	-	0.472
	NMI	<b>0.493</b>	-	0.446	0.484	-	0.478
	EP	<b>0.749</b>	-	0.699	0.731	-	0.722
T5-1	ARI	<b>0.386</b>	-	0.381	0.378	-	0.374
	NMI	<b>0.485</b>	-	0.442	0.447	-	0.448
	EP	0.596	-	<b>0.599</b>	0.591	-	0.587
T6-1	ARI	0.196	0.376	0.251	0.201	<b>0.390</b>	0.216
	NMI	0.307	0.375	0.255	0.309	<b>0.381</b>	0.304
	EP	0.531	<b>0.684</b>	0.606	0.530	0.681	0.546
T4-2	ARI	<b>0.506</b>	-	0.402	0.456	-	0.440
	NMI	<b>0.492</b>	-	0.441	0.473	-	0.467
	EP	<b>0.742</b>	-	0.690	0.706	-	0.701
T5-2	ARI	<b>0.383</b>	-	0.345	0.350	-	0.341
	NMI	<b>0.475</b>	-	0.410	0.418	-	0.414
	EP	<b>0.590</b>	-	0.579	0.577	-	0.568
T6-2	ARI	0.204	0.361	0.244	0.208	<b>0.375</b>	0.219
	NMI	0.313	0.377	0.245	0.327	<b>0.385</b>	0.307
	EP	0.536	0.676	0.599	0.538	<b>0.678</b>	0.548
T4-3	ARI	<b>0.450</b>	-	0.325	0.354	-	0.346
	NMI	<b>0.455</b>	-	0.368	0.402	-	0.400
	EP	<b>0.713</b>	-	0.629	0.633	-	0.629
T5-3	ARI	<b>0.378</b>	-	0.308	0.279	-	0.291
	NMI	<b>0.463</b>	-	0.356	0.349	-	0.365
	EP	<b>0.591</b>	-	0.556	0.524	-	0.538
T6-3	ARI	0.205	0.327	0.257	0.222	<b>0.330</b>	0.264
	NMI	0.315	<b>0.363</b>	0.271	0.303	0.362	0.322
	EP	0.533	<b>0.653</b>	0.609	0.575	0.646	0.616

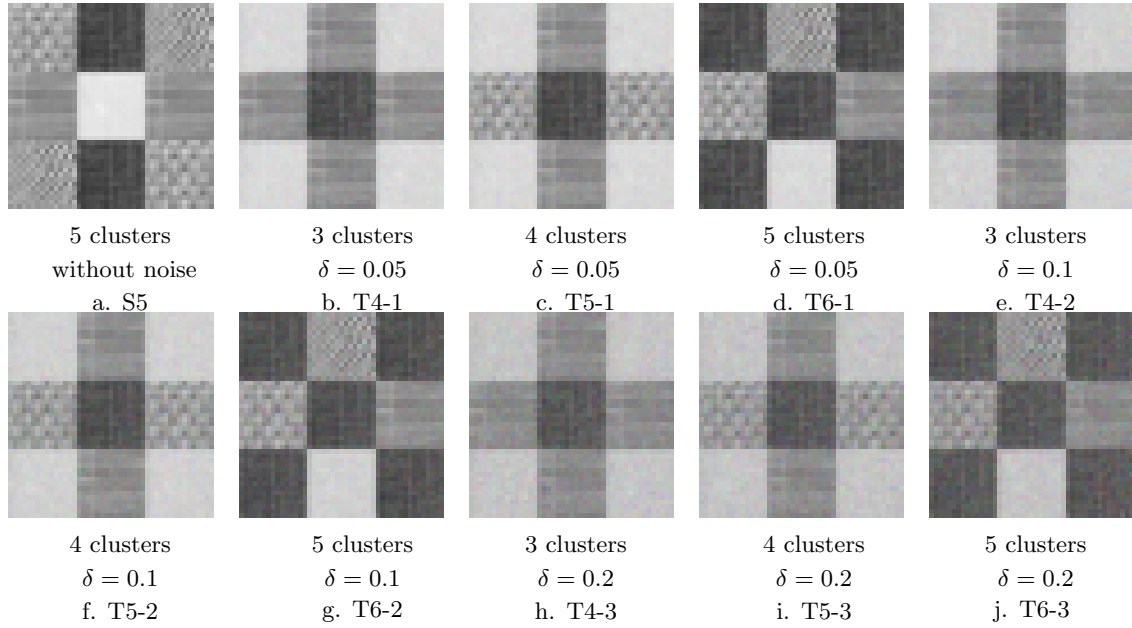


Figure 7: Texture images for transfer clustering.

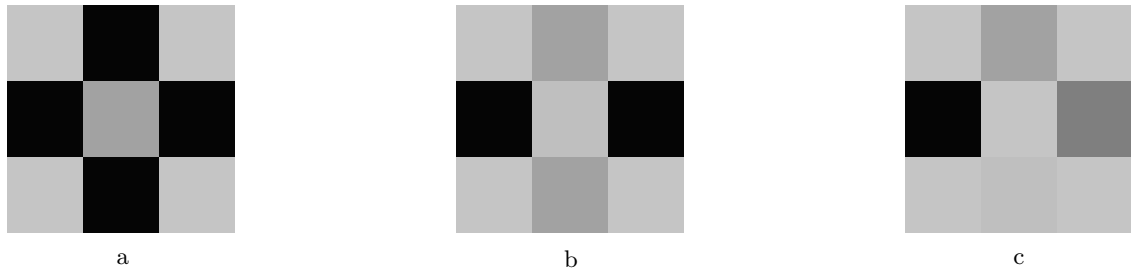


Figure 8: The ideal segmentation of target texture images. a for T4-1, T4-2 and T4-3; b for T5-1, T5-2 and T5-3; c for T6-1, T6-2 and T6-3.

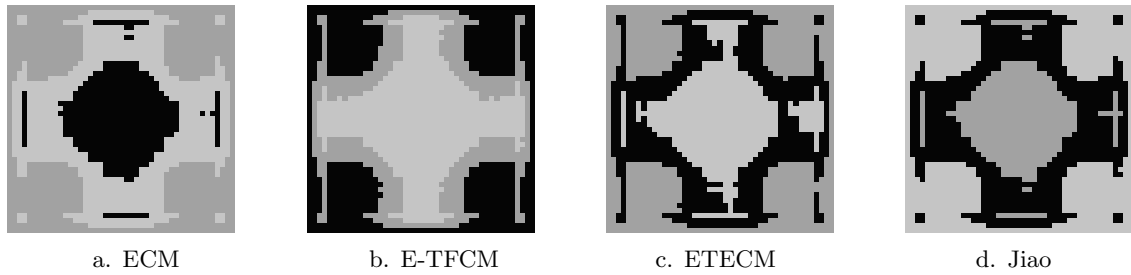


Figure 9: Segmentation of T4-1.

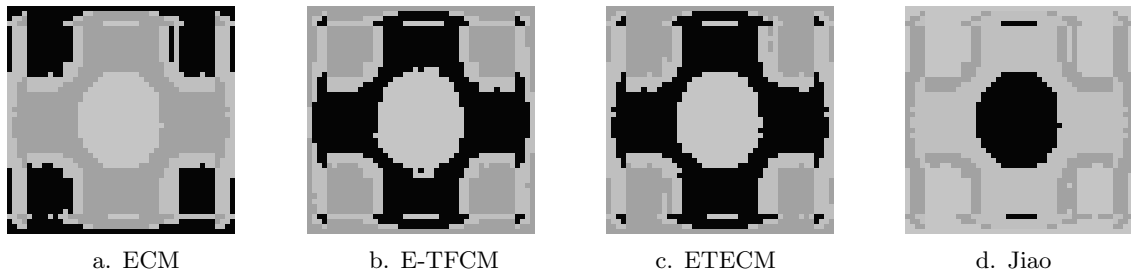


Figure 10: Segmentation of T5-1.

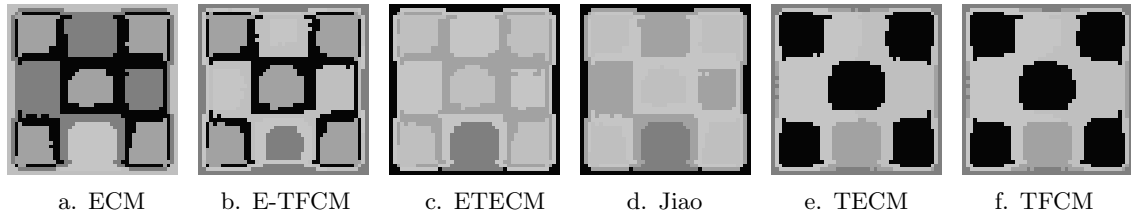


Figure 11: Segmentation of T6-1.

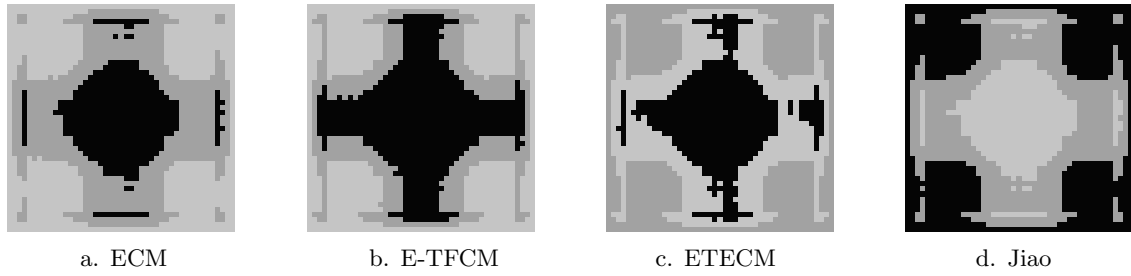


Figure 12: Segmentation of T4-2.

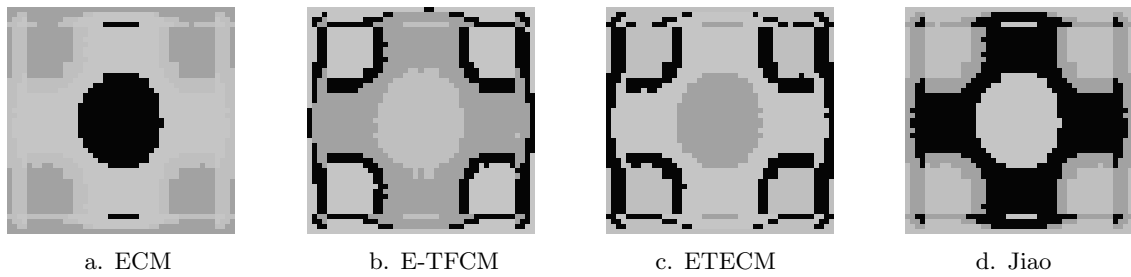


Figure 13: Segmentation of T5-2.

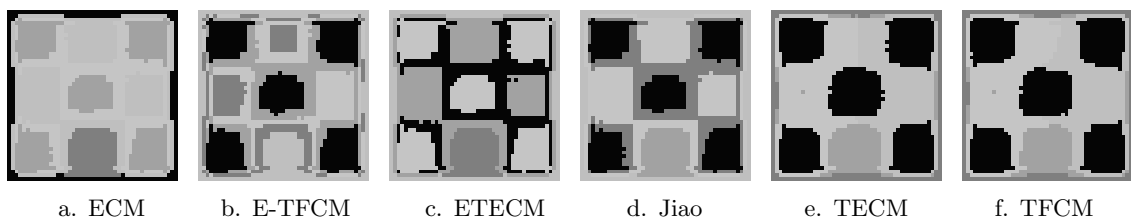


Figure 14: Segmentation of T6-2.



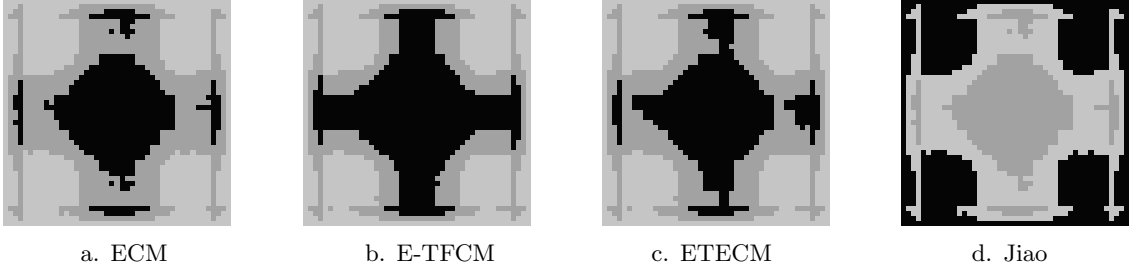


Figure 15: Segmentation of T4-3.

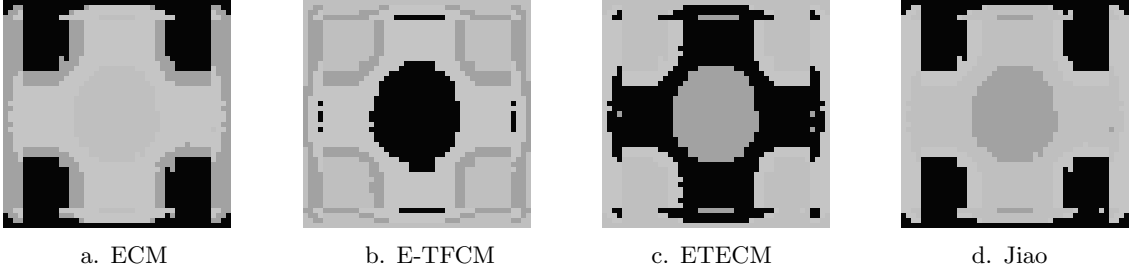


Figure 16: Segmentation of T5-3.

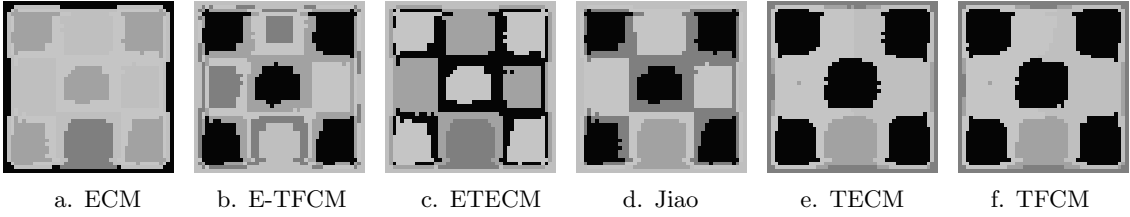


Figure 17: Segmentation of T6-3.

#### 5.4. UCI data sets

Ten UCI data sets were used in this subsection: Seeds, Wine, Glass, Breast Tissue (BT), Wisconsin Diagnostic Breast Cancer (WDBC), Robot Navigation (RN), Breast Cancer (BC), Diabetes, Vehicle data set, and Karate Club network (Karate). We show data size, feature dimension, and the number of clusters (NC) in Table 11. It is remarked here as Karate Club network is a graph data set, it has no dimension. In the experiment,  $nf$  features were selected from the original data to form the target data sets. The selected feature indexes are reported in Table 11. For Karate graph, the vector embedding was first calculated by spectral decomposition of its adjacency matrices [42]. The embedding dimension was set to  $nf$ .

Then the ECM, TFCM, TECM, and ETECM algorithms were used. We note here as the benchmarks for these data sets are known, in TECM, TFCM and ETECM, the average values of the samples in the target data were directly used to simulate the prototypes in the source domain.

The ARI, NMI, and EP values of the clustering results provided by ECM, TFCM, and TECM are listed in Tables 12-13. TECM and ETECM perform better in most data sets than other methods. This confirms the advantages of the evidential transfer clustering approach

Table 11: The information of the UCI data sets.

Data set	size	Dimension	NC	feature index(nf = 2)	feature index(nf = 3)
Seeds	210	7	3	(4, 6)	(2, 3, 5)
Wine	178	13	3	(3, 5)	(4, 9, 6)
Glass	214	9	6	(1, 8)	(6, 8, 9)
BT	106	9	6	(1, 8)	(2, 4, 7)
WDBC	569	30	2	(4, 26)	(5, 11, 22)
RN	5456	24	4	(16,24)	(3, 11, 23)
BC	699	9	2	(4, 9)	(3, 4, 7)
Diabetes	768	8	2	(3, 6)	(1, 2, 3)
Vehicle	846	18	4	(7, 11)	(8, 11, 18)
Karate	34	-	2	-	-

Table 12: ARI, NMI, and EP of clustering results on the UCI data sets with  $nf = 2$ .

Dataset	ARI				NMI				EP			
	ECM	TFCM	TECM	ETECM	ECM	TFCM	TECM	EETECM	ECM	TFCM	TECM	EETECM
Seeds	0.178	0.216	<b>0.254</b>	0.143	0.178	<b>0.203</b>	0.222	0.161	0.444	0.474	<b>0.495</b>	<b>0.495</b>
Wine	0.119	0.122	0.125	<b>0.151</b>	0.122	0.111	0.109	<b>0.148</b>	<b>0.466</b>	0.418	0.407	0.407
Glass	0.140	0.147	<b>0.150</b>	0.129	<b>0.216</b>	0.196	0.189	0.185	<b>0.348</b>	0.321	0.321	0.321
BT	0.319	0.356	<b>0.472</b>	0.045	0.502	0.485	<b>0.570</b>	0.087	<b>1.000</b>	0.422	<b>1.000</b>	<b>1.000</b>
WDBC	0.436	0.514	<b>0.562</b>	0.070	0.368	0.415	<b>0.449</b>	0.188	0.870	0.736	<b>1.000</b>	<b>1.000</b>
RN	0.035	0.047	<b>0.071</b>	0.048	0.079	0.094	<b>0.122</b>	0.080	<b>0.386</b>	0.364	0.380	0.380
BC	0.373	0.461	<b>0.473</b>	0.427	0.279	0.347	<b>0.358</b>	0.319	<b>0.765</b>	0.717	0.723	0.723
Diabetes	0.012	0.003	<b>0.026</b>	0.008	0.002	0.021	<b>0.024</b>	0.014	0.515	0.547	<b>0.559</b>	<b>0.559</b>
Vehicle	0.101	0.102	0.114	<b>0.129</b>	0.138	0.106	0.122	<b>0.154</b>	<b>0.471</b>	0.310	0.304	0.304
Karate	0.264	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.317	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.903	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>

when there is some available positively transferred knowledge in the source domain.

## 6. Conclusion

In this study, the concept of knowledge transfer has been used to develop an evidential transfer clustering method named TECM for the application of clustering task when the target data are uncertain or insufficient. The proposed TECM algorithm can learn from not only the data of the target domain but also the knowledge of the source domain in the form of prototypes.

In TECM, the cluster numbers of the source and target domains are assumed to be identical. This may be difficult to satisfy in practice. ETECM is designed for the case where source and target domains have different numbers of clusters. The experimental results on generated data and UCI data have demonstrated the effectiveness of the TECM and ETECM algorithms. The problem of TECM, which can be seen from the experimental results in Section 5, is that the source data should be relevant to the target data. This is a general assumption in transfer learning. In the following research, we will also study how to address this problem when the information from the two domains is not very related. Another problem of the proposed methods is that the number of attributes of samples in the source and target should be the same. However, sometime this condition cannot be satisfied in reality. In this case we can use some feature selection methods first to make the data in the two domains sharing the same dimension. As mentioned, how to reduce the distribution discrepancy should also be considered in the feature selection. These problems may be addressed by introducing new items in the objective function. We will study these points in future work.

Table 13: ARI, NMI, and EP of clustering results on UCI data sets with  $nf = 3$ .

Dataset	ARI				NMI				EP			
	ECM	TFCM	TECM	ETECM	ECM	TFCM	TECM	ETECM	ECM	TFCM	TECM	ETECM
Seeds	0.556	<b>0.663</b>	0.621	0.620	0.573	<b>0.642</b>	0.591	0.597	<b>0.900</b>	0.771	0.794	0.794
Wine	<b>0.154</b>	0.142	0.143	0.107	<b>0.198</b>	0.177	0.178	0.171	<b>0.456</b>	0.432	0.436	0.436
Glass	0.129	0.139	<b>0.173</b>	0.097	0.244	0.219	<b>0.247</b>	0.224	0.342	0.346	<b>0.361</b>	<b>0.361</b>
BT	0.272	0.243	<b>0.274</b>	0.087	<b>0.467</b>	0.421	0.422	0.151	0.143	0.339	<b>0.373</b>	<b>0.373</b>
WDBC	0.190	<b>0.195</b>	<b>0.195</b>	0.190	<b>0.144</b>	0.136	0.137	0.144	<b>0.716</b>	0.627	0.651	0.651
RN	0.015	0.052	<b>0.059</b>	0.015	0.057	0.093	<b>0.096</b>	0.057	0.376	0.379	<b>0.386</b>	<b>0.386</b>
BC	0.675	<b>0.730</b>	<b>0.730</b>	0.690	0.552	<b>0.606</b>	0.603	0.567	<b>0.914</b>	0.856	0.858	0.858
Diabetes	0.171	0.191	<b>0.194</b>	0.066	0.116	0.121	<b>0.125</b>	0.114	<b>0.804</b>	0.635	0.603	0.603
Vehicle	0.105	0.105	<b>0.120</b>	0.116	0.133	0.115	<b>0.130</b>	0.125	<b>0.393</b>	0.314	0.369	0.369
Karate	0.264	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.317	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.760	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.61701409), the Aeronautical Science Foundation of China (No.20182053023), and the Science Research Plan of China (Xi'an) Institute for Silk Road Research (2019ZD02).

## References

- [1] K. Zhou, M. Guo, A. Martin, Evidential clustering based on transfer learning, in: International Conference on Belief Functions, Springer, 2021, pp. 269–277.
- [2] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. P. Patel, A. Tiwari, M. J. Er, W. Ding, C.-T. Lin, A review of clustering techniques and developments, *Neurocomputing* 267 (2017) 664–681.
- [3] A. K. Jain, Data clustering: 50 years beyond k-means, *Pattern recognition letters* 31 (2010) 651–666.
- [4] L. Bai, J. Liang, F. Cao, Semi-supervised clustering with constraints of different types from multiple information sources, *IEEE transactions on pattern analysis and machine intelligence* 43 (2021) 3247–3258.
- [5] X. Liu, M. Li, C. Tang, J. Xia, J. Xiong, L. Liu, M. Kloft, E. Zhu, Efficient and effective regularized incomplete multi-view clustering, *IEEE transactions on pattern analysis and machine intelligence* (2020).
- [6] Z. Deng, Y. Jiang, F.-L. Chung, H. Ishibuchi, K.-S. Choi, S. Wang, Transfer prototype-based fuzzy clustering, *IEEE Transactions on Fuzzy Systems* 24 (2015) 1210–1232.
- [7] R. Gargees, J. M. Keller, M. Popescu, TLPCM: Transfer learning possibilistic  $c$ -means, *IEEE Transactions on Fuzzy Systems* (2020).
- [8] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, *Proceedings of the IEEE* 109 (2021) 43–76.
- [9] S. J. Pan, Q. Yang, A survey on transfer learning, *IEEE Transactions on Knowledge and Data Engineering* 22 (2010) 1345–1359.

- [10] W. Dai, Q. Yang, G.-R. Xue, Y. Yu, Boosting for transfer learning, in: ICML '07, 2007, pp. 193–200.
- [11] A. Arnold, R. Nallapati, W. W. Cohen, A comparative study of methods for transductive transfer learning, in: Seventh IEEE international conference on data mining workshops (ICDMW 2007), IEEE, 2007, pp. 77–82.
- [12] X. Ling, W. Dai, G.-R. Xue, Q. Yang, Y. Yu, Spectral domain-transfer learning, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, pp. 488–496.
- [13] W. Dai, Q. Yang, G.-R. Xue, Y. Yu, Self-taught clustering, in: Proceedings of the 25th international conference on Machine learning, 2008, pp. 200–207.
- [14] Z. Wang, Y. Song, C. Zhang, Transferred dimensionality reduction, in: Joint European conference on machine learning and knowledge discovery in databases, Springer, 2008, pp. 550–565.
- [15] P. Qian, S. Sun, Y. Jiang, K.-H. Su, T. Ni, S. Wang, R. F. Muzic Jr, Cross-domain, soft-partition clustering with diversity measure and knowledge reference, *Pattern Recognition* 50 (2016) 155–177.
- [16] P. Qian, Y. Jiang, Z. Deng, L. Hu, S. Sun, S. Wang, R. F. Muzic, Cluster prototypes and fuzzy memberships jointly leveraged cross-domain maximum entropy clustering, *IEEE transactions on cybernetics* 46 (2015) 181–193.
- [17] Y. Jiang, X. Gu, D. Wu, W. Hang, J. Xue, S. Qiu, C.-T. Lin, A novel negative-transfer-resistant fuzzy clustering model with a shared cross-domain transfer latent space and its application to brain ct image segmentation, *IEEE/ACM transactions on computational biology and bioinformatics* 18 (2020) 40–52.
- [18] G. Shafer, *A mathematical theory of evidence*, Princeton University Press, 1976.
- [19] Z. Tong, P. Xu, T. Denœux, Fusion of evidential cnn classifiers for image classification, in: *International Conference on Belief Functions*, Springer, 2021, pp. 168–176.
- [20] X. Yue, Y. Chen, B. Yuan, Y. Lv, Three-way image classification with evidential deep convolutional neural networks, *Cognitive Computation* (2021) 1–13.
- [21] V.-D. Nguyen, V.-N. Huynh, Two-probabilities focused combination in recommender systems, *International Journal of Approximate Reasoning* 80 (2017) 225–238.
- [22] T. Xiahou, Z. Zeng, Y. Liu, H.-Z. Huang, Measuring conflicts of multisource imprecise information in multistate system reliability assessment, *IEEE Transactions on Reliability* (2021).
- [23] T. Denœux, Maximum likelihood estimation from uncertain data in the belief function framework, *Knowledge and Data Engineering, IEEE Transactions on* 25 (2013) 119–130.
- [24] J. Cahoon, R. Martin, Generalized inferential models for censored data, *International Journal of Approximate Reasoning* 137 (2021) 51–66.

- [25] T. Denœux, M.-H. Masson, EVCLUS: evidential clustering of proximity data, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 34 (2004) 95–109.
- [26] M.-H. Masson, T. Denœux, ECM: An evidential version of the fuzzy  $c$ -means algorithm, *Pattern Recognition* 41 (2008) 1384–1397.
- [27] C. Gong, Z.-g. Su, P.-h. Wang, Q. Wang, An evidential clustering algorithm by finding belief-peaks and disjoint neighborhoods, *Pattern Recognition* 113 (2021) 107751.
- [28] K. Zhou, A. Martin, Q. Pan, Z. Liu, SELP: Semi-supervised evidential label propagation algorithm for graph data clustering, *International Journal of Approximate Reasoning* 92 (2018) 139–154.
- [29] V. Antoine, B. Quost, M.-H. Masson, T. Denœux, CECM: Constrained evidential  $c$ -means algorithm, *Computational Statistics & Data Analysis* 56 (2012) 894–914.
- [30] V. Antoine, J. A. Guerrero, J. Xie, Fast semi-supervised evidential clustering, *International Journal of Approximate Reasoning* 133 (2021) 116–132.
- [31] T. Denœux, Calibrated model-based evidential clustering using bootstrapping, *Information Sciences* 528 (2020) 17–45.
- [32] T. Denœux, Nn-evclus: Neural network-based evidential clustering, *Information Sciences* 572 (2021) 297–330.
- [33] Z. Liu, L.-Q. Huang, K. Zhou, T. Denœux, Combination of transferable classification with multisource domain adaptation based on evidential reasoning, *IEEE Transactions on Neural Networks and Learning Systems* 32 (2020) 2015–2029.
- [34] Z. Liu, G. Qiu, G. Mercier, Q. Pan, A transfer classification method for heterogeneous data based on evidence theory, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51 (2019) 5129–5141.
- [35] L. Huang, Z. Liu, Q. Pan, J. Dezert, Evidential combination of augmented multi-source of information based on domain adaptation, *Science China Information Sciences* 63 (2020) 1–14.
- [36] P. Smets, Decision making in the TBM: the necessity of the pignistic transformation, *International Journal of Approximate Reasoning* 38 (2005) 133–147.
- [37] L. Jiao, F. Wang, Q. Pan, Transfer evidential  $c$ -means clustering, in: *International Conference on Belief Functions*, Springer, 2021, pp. 47–55.
- [38] K. Zhou, A. Martin, Q. Pan, Z. Liu, Median evidential  $c$ -means algorithm and its application to community detection, *Knowledge-Based Systems* 74 (2015) 69–88.
- [39] K. Zhou, A. Martin, Q. Pan, Z. Liu, ECMdd: Evidential  $c$ -medoids clustering with multiple prototypes, *Pattern Recognition* 60 (2016) 239–257.
- [40] S. J. Pan, I. W. Tsang, J. T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, *IEEE transactions on neural networks* 22 (2010) 199–210.

- [41] V. Kyrki, J.-K. Kamarainen, H. Kälviäinen, Simple gabor feature space for invariant object recognition, *Pattern recognition letters* 25 (2004) 311–318.
- [42] D. L. Sussman, M. Tang, D. E. Fishkind, C. E. Priebe, A consistent adjacency spectral embedding for stochastic blockmodel graphs, *Journal of the American Statistical Association* 107 (2012) 1119–1128.