

Credal Transfer Learning With Multi-Estimation for Missing Data

ZONGFANG MA¹, ZHE LIU¹, YIRU ZHANG^{1,2}, LIN SONG¹, AND JIHUAN HE^{1,3}

¹School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China

²Institute de Recherche en Informatique et Systèmes Aléatoires, University of Rennes 1, 35380 Rennes, France

³National Engineering Laboratory for Modern Silk, Soochow University, Suzhou 215006, China

Corresponding author: Yiru Zhang (yiru.zhang@univ-rennes1.fr)

ABSTRACT Transfer learning (TL) has grown popular in recent years. It is effective to improve the classification accuracy in the target domain by using the training knowledge in the related domain (called source domain). However, the classification of missing data (or incomplete data) is a challenging task for TL because different strategies of imputation may have strong impacts on learning models. To address this problem, we propose credal transfer learning (CTL) with multi-estimation for missing data based on belief function theory by introducing uncertainty and imprecision in data imputation procedure. CTL mainly consists of three steps: Firstly, the query patterns are reasonably mapped into multiple versions in source domain to characterize the uncertainty caused by missing values. Afterwards, the multiple mapping patterns are classified in the source domain to obtain the corresponding outputs with different discounting factors. Finally, the discounted outputs, represented by the basic belief assignments (BBAs), are submitted to a new belief-based fusion system to get the final classification result for the query patterns. Three comparative experiments are given to illustrate the interests and potentials of CTL method.


INDEX TERMS Transfer learning, missing data, belief function, Credal classification, uncertainty.

I. INTRODUCTION

Traditional machine learning algorithms have already achieved great success under the assumption that training and test set are drawn from the same feature space and data distributions [1]. In many real-world situations, however, this assumption is not satisfied, which usually makes the performance of traditional classifier unsatisfying. Recently, a new method, called Transfer learning (TL) [1]–[3], has been proposed, which can effectively solve the above problems and is widely used in many fields, such as indoor WiFi location [4], text classification [5], sentiment analysis [6], *etc.*

According to the availability of training patterns in the target domain, TL methods are categorized into three types: supervised TL [7]–[9], semi-supervised TL [10]–[12] and unsupervised TL [13]–[15]. Supervised TL methods utilize the labeled target domain patterns in addition to the source domain patterns for training. For example, Transfer Adaboost (TrAdaboost) method [7], which is quite typical, extends the Adaboost algorithm by adding a weighting

mechanism corresponding to the the similarity of patterns in source domain and target domain. In [8], a heterogeneous feature augmentation (HFA) method is proposed. HFA transforms the patterns of two domains into a common subspace to augment the mapping patterns. Semi-supervised TL methods further use unlabeled target domain patterns to help with classification. An extended version of HFA, called semi-supervised heterogeneous feature augmentation (SHFA) [10] addresses the heterogeneous situations with sufficient labeled source patterns and limited target patterns. Interestingly, a semi-supervised TL method based on manifold regularization is proposed in [11], which exploits similarity constraints in the target domain to improve performance. Unsupervised TL is an interesting but challenging task, since it is applicable to the target domain without labeled patterns. A representative work is transfer component analysis (TCA) [13], which minimizes the distances between two domain distributions by mapping the patterns of two domains to a reproducing kernel Hilbert space. In [14], a joint domain adaptation (JDA) strategy is presented to simultaneously adapt the margin and condition distribution differences between the labeled source domain and the unlabeled

The associate editor coordinating the review of this manuscript and approving it for publication was Inês Domingues .

target domain. However, all these TL methods are designed on complete patterns without considering the missing data situations. Unfortunately, missing data is a common issue in many real-world data sets. For example, UCI, one of the standard repositories commonly used in machine learning algorithms, contains 45% of the data sets with missing values [16]. Under such circumstances, these classical TL classification methods are no longer adaptable. Therefore, pre-process on the missing data before classification is necessary.

A number of methods [17], [18] have been developed to deal with traditional classification problems with missing data. Generally, these methods respect one missing randomness mechanism among three assumptions: missing completely at random (MCAR), missing at random (MAR) and not missing at random (NMAR). The simplest method is to discard incomplete patterns, which is acceptable when the missing values only count for a small proportion (less than 5%) of the whole data set. Imputation strategy, in many situations, is a popular method for incomplete pattern classification [19]–[23], [25], [26]. For instance, in *mean imputation* (MI) [19], the missing values are simply replaced by the mean values of the complete attributes in the same dimension. A commonly used *K*-nearest neighbor imputation (KNNI) method [20], [21] uses *K*-nearest neighbors (KNNs) of patterns to estimate missing values. In fuzzy *c*-means imputation (FCMI) method [22], [23], the missing values are filled with the clustering centers produced by fuzzy *c*-means (FCM) [24] and the distances between the pattern and the centers.

Particularly, linear local approximation (LLA) [25], uses the KNNs with optimal weights obtained by local linear reconstruction to estimate the missing values. Interestingly, some recent research works have been dedicated to multiple estimation or non-estimation of missing values [27]–[29]. These methods have achieved satisfying results in some way, whereas they cannot be directly employed in TL because of the inconsistency of feature space and distribution. In addition to missing values, knowledge transfer will also bring uncertainty in TL, since the calculation error caused by transfer rules will inevitably occur in the process of transfer. Thus, how to reasonably characterize the transfer and classification uncertainty caused by missing values is a very meaningful work.

In this paper, we propose a credal transfer learning (CTL) method for missing data, which introduces uncertainty and imprecision while imputing missing values based on the belief function theory. Belief function theory [30] has been widely used in modeling and reasoning uncertain information in application domains of pattern classification [31]–[34], pattern clustering [35], [36] and information fusion [37], [38], also conventionally called *credal methods*. In credal classification methods, one pattern may belong to multiple classes (*i.e.* particular dis-junction of several singleton classes), named *meta-classes*, with different belief degree. Such representation is able to characterize the imprecision of classification for uncertain patterns. There are some methods

developed for missing data based on belief function [31], [32]. For instance, a prototype-based credal classification (PCC) method is proposed in [31], where the missing values are estimated respectively with the class prototypes obtained by training patterns and they can be classified by traditional classifiers. More recently, a new transfer classification method, named evidence-based heterogeneous transfer classification (EHTC) is proposed in [34] to deal with the uncertainty in the process of feature mapping in heterogeneous domains. Nevertheless, there is no relevant literature(s) on transfer learning of incomplete pattern based on belief function theory.

CTL method applies belief function theory for the representation of uncertainty information in transfer learning on incomplete data. In CTL, we assume that patterns in source domain are attributed with ground truth labels while labels of patterns in target source are not observed. The feature distributions of the two domains are different while attributes of the patterns in the target domain are partially missing. Specifically, CTL first uses observed attributes to estimate multiple mapping patterns in the source domain for each pattern with missing values in the target domain based on KNNs techniques. Afterwards, a basic classifier (such as K-NN [39], EK-NN [40], NB [41]) that handles the complete pattern is selected. In this step, the labeled patterns in the source domain are used to classify the multi-mapped versions of each query pattern. Finally, different discounting factors of multi-classification results are obtained depending on the distances between query patterns and corresponding KNNs. Final classification results are obtained from multi-classification results. Non-conflicting classification results are directly fused by discounted averaging method while an adaptive fusion method is designed to aggregate the remaining conflicting results. By conducting such step, the patterns that are difficult to be classified are automatically submitted to the reasonable meta-class, which is able effectively reduce misclassification rate. The classification of the uncertain patterns in meta-class can be eventually identified (refined) using certain other (costly) techniques or with extra information sources.

The contributions of this work mainly concern three aspects.

1) A multi-estimation strategy in different distribution domains is proposed. In this strategy, the unobserved attributes of incomplete patterns are estimated based on observed ones, with an uncertainty degree reasoned by the belief function theory. The capability of uncertainty reasoning is one of the advantages over traditional imputation methods.

2) A new adaptive global fusion method for decision-making in classification is designed. Because of the uncertainty reasoned by the belief function theory, CTL is able to make the decision more cautiously by considering the imprecision and uncertainty of learning results. Such decision making method effectively reduces the error rate in practice, which is justified by experiments on real world data.

3) Evidential theory (belief function theory) is originally introduced in transfer learning, with effectiveness of classification application justified on incomplete data sets.

This paper is organized as follows. The preliminary information of transfer learning and belief function theory is shortly reviewed in section II, and the CTL method is introduced in the Section III. The performance of CTL is tested and compared with several other methods in Section IV. The conclusion of this paper is finally given in Section V.

II. PRELIMINARIES

In this section, brief introduction of transfer learning (TL) and belief function theory is given as well as corresponding notations.

A. TRANSFER LEARNING

In TL, the representation of patterns are transferred between different *domains*. A domain \mathcal{D} is constituted of two components: feature space \mathcal{X} and marginal probability distribution $P(X)$. Formally, $\mathcal{D} = \{\mathcal{X}, P(X)\}$, $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \in \mathcal{X}$, where \mathbf{x}_i is the i -th pattern of \mathcal{D} . For a given domain \mathcal{D} , a task \mathcal{T} is composed of two elements: label space \mathcal{Y} and decision function $f(\cdot)$. Formally, $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$, $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_n\} \in \mathcal{Y}$, where \mathbf{y}_i denotes the corresponding output label.

Given a source domain \mathcal{D}^s with a corresponding a source task \mathcal{T}^s and a target domain \mathcal{D}^t with a corresponding a target task \mathcal{T}^t , $\mathcal{D}^s \neq \mathcal{D}^t$ or $\mathcal{T}^s \neq \mathcal{T}^t$. TL is the process of improving the decision function $f_i(\cdot)$ in the target domain \mathcal{D}^t by using relevant knowledge in source domain \mathcal{D}^s and source task \mathcal{T}^s .

The existing TL methods can be divided into three categories: supervised TL [7]–[9], semi-supervised TL [10]–[12] and unsupervised TL [13]–[15]. supervised TL requires training classifiers with sufficient labeled patterns. Semi-supervised TL uses few labeled patterns but vast unlabeled patterns to train the classifier. In this paper, we mainly focuses on the unsupervised TL, where no labeled patterns are available for training. More detailed introduction and examples of TL are available in [1], [2] and [3].

B. BELIEF FUNCTION THEORY AND CREDAL PARTITION

Belief function theory, also known as Dempster-Shafer theory (DST) or evidence theory [30], is originally proposed by Dempster and formed by Shafer generalization. It is a theoretical framework for reasoning with partial and unreliable information, notably the uncertain problems. It has been successfully applied in many fields [31]–[38]. In this theory, a set of finite mutually exclusive and complete elements $\Omega = \{\omega_1, \omega_2, \dots, \omega_c\}$ is defined as the *framework of discernment* of the problem under study, usually a decision problem. The uncertainty is expressed on the power-set of Ω , denoted as 2^Ω , where the disjunctive elements imply information with imprecision.

The basic belief assignment (BBA) $m(\cdot)$ on the framework of discernment Ω is a function $m : 2^\Omega \rightarrow [0, 1]$, such that

$$\begin{cases} \sum_{A \in 2^\Omega} m(A) = 1 \\ m(\emptyset) = 0 \end{cases} \quad (1)$$

All the elements $A \in 2^\Omega$ such that $m(A) > 0$ are called *focal elements* of $m(\cdot)$.

A credal partition [35] is defined as the n -tuple $M = (m_1, \dots, m_n)$, where m_i is the BBA of the pattern $\mathbf{x}_i \in X$, $i = 1, 2, \dots, n$ associated with the different elements of the power-set 2^Ω .

In classification problems, the output of each classifier can be regarded as an evidence on all possible classes represented by a BBA. The DS rule [30] is used in many applications to combine multiple evidence of different independent sources because of its commutative and associative properties. The DS combination of evidence $m_1(\cdot)$ and $m_2(\cdot)$ from two independent sources over a frame of discernment 2^Ω is defined by

$$m_{DS}(A) = \begin{cases} 0, & A = \emptyset \\ \frac{\sum_{B \cap C = A} m_1(B)m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B)m_2(C)}, & A \neq \emptyset, \quad \forall A \in 2^\Omega \end{cases} \quad (2)$$

In DS rule, all conflict belief mass $\sum_{B \cap C = \emptyset} m_1(B)m_2(C)$ is proportionally redistributed back to the focus element. However, DS rules can also produce very unreasonable results in high conflict situations and some special low conflict situations. Thus, a number of alternative combination rules have emerged to overcome the limitations of DS rule, such as the well-known Yager's rule [42], Dubois-Prade (DP) rule [43], and more recently the more complex Proportional Conflict Redistributions (PCR) rules [44] are found.

III. CREDAL TRANSFER LEARNING

CTL method is developed for incomplete data classification in transfer learning, where data is partially missing (or unobserved). It consists of two steps.

Firstly, CTL estimates multiple mapping patterns in the source domain for each pattern with missing values in the target domain by assuming that there are a few number of parallel (one-to-one) patterns pairs. In this step, we assume that some one-to-one pattern pairs are given to link two different domains, whereas the labels of these patterns are unknown. Afterwards, the mapping values for (incomplete) patterns will be classified by the corresponding trained classifiers to obtain multiple different classification results, which are submitted to a new belief-based fusion system with different weights (reliability) to get the final classification result for the each pattern with missing values. In this step, multiple labels may be assigned to one pattern, interpreted as imprecision. Under such circumstances, these patterns are submitted to a corresponding meta-class reasoning the imprecision. Finally,

the classification of the uncertain patterns in meta-class can be eventually identified using some cautious techniques or with extra information sources. Therefore, CTL method is able to prevent erroneous fatal decisions by cautiously partitioning the classification results when necessary.

A. MULTI-ESTIMATION AND CLASSIFICATION

Given a data set \tilde{X}^s with vast labeled complete patterns¹ in source domain and a data set \tilde{X}^t in target domain with unlabeled patterns partially observed, where the feature distribution of two domains are completely different but label spaces identical, i.e., patterns in \tilde{X}^s and \tilde{X}^t are in the class partition framework $\Omega = \{\omega_1, \dots, \omega_c\}$. Here, we assume that there are some one-to-one pattern pairs to build cross domain connections in two domains, respectively denoted as $X^s = \{\mathbf{x}_1^s, \dots, \mathbf{x}_m^s\}$ and $X^t = \{\mathbf{x}_1^t, \dots, \mathbf{x}_m^t\}$. The labels of these patterns are not available since there is no prior information in target domain. For a query (incomplete) pattern $\tilde{\mathbf{x}}_i$ in target domain, K -nearest neighbors (KNNs) strategy² is applied to estimate multiple mapping patterns in source domain.

In the process of multi-estimation, the KNNs of the query pattern $\tilde{\mathbf{x}}_i$ are firstly searched with the observed attributes. Hence the calculation of the distance between the incomplete pattern in \tilde{X}^t and the complete pattern in X^t is very critical. In CTL, the distance between the pattern $\tilde{\mathbf{x}}_i$ and complete pattern \mathbf{x}_k^t is given for $\tilde{\mathbf{x}}_i \in \tilde{X}^t$ and $\mathbf{x}_k^t \in X^t$ by:

$$\|\tilde{\mathbf{x}}_i, \mathbf{x}_k^t\| = \sqrt{\sum_{\exists s, s=1}^p (\tilde{x}_{is} - x_{ks}^t)^2} \quad (3)$$

where $\|\cdot\|$ denotes the Euclidean distance, \tilde{x}_{is} and x_{ks}^t the s -th attributes of the patterns, respectively. p is the number of dimensions of observed attributes in $\tilde{\mathbf{x}}_i$.

Afterwards, the \mathcal{K} minimum distances $\|\tilde{\mathbf{x}}_i, \mathbf{x}_k^t\|$ ($k = 1, \dots, \mathcal{K}$) and corresponding complete patterns $\mathbf{x}_k^t \in X$ ($k = 1, \dots, \mathcal{K}$) are obtained from these distances. Since \mathbf{x}_k^t and $\tilde{\mathbf{x}}_i$ are one-to-one pattern pairs that connecting target and source domain called as *bridge*, for the pattern $\tilde{\mathbf{x}}_i$ with missing values, \mathcal{K} version mapping values in source domain can be estimated for the pattern $\tilde{\mathbf{x}}_i$ according to KNNs as follows.

$$\tilde{\mathbf{x}}_i^k = \mathbf{x}_k^s \quad (4)$$

where \mathbf{x}_k^s is the mapping value of \mathbf{x}_k^t in source domain, and $k = 1, \dots, \mathcal{K}$.

A simple example is given to illustrate the process of estimating multiple mapping values in the source domain.

Example 1: Given a source domain with three attributes and a target domain with four. The 2nd attribute of a pattern $\tilde{\mathbf{x}}_i$ is unobserved. We assume that three nearest neighbors are found in X^t according to the observed attributes

¹in this paper, we focus on the classification of (incomplete) patterns in the target domain. Thus, the labeled patterns in the source domain are assumed to be complete.

² K -nearest neighbors (KNNs) strategy is a simple and rational method, because it can provide \mathcal{K} versions of possible estimations, which well reflects the uncertain and imprecision of estimation.

of $\tilde{\mathbf{x}}_i$, denoted as follows.

$$\begin{aligned} \mathbf{x}_1^t &= [x_{11}^t, x_{12}^t, x_{13}^t, x_{14}^t] \\ \mathbf{x}_2^t &= [x_{21}^t, x_{22}^t, x_{23}^t, x_{24}^t] \\ \mathbf{x}_3^t &= [x_{31}^t, x_{32}^t, x_{33}^t, x_{34}^t] \end{aligned}$$

Thus, according to Eq. (4), three mapping patterns $\tilde{\mathbf{x}}_i^k$, ($k = 1, 2, 3$) in source domain are estimated by \mathbf{x}_k^t , ($k = 1, 2, 3$) of pattern $\tilde{\mathbf{x}}_i$.

$$\begin{aligned} \tilde{\mathbf{x}}_i^1 &= [x_{11}^s, x_{12}^s, x_{13}^s] \\ \tilde{\mathbf{x}}_i^2 &= [x_{21}^s, x_{22}^s, x_{23}^s] \\ \tilde{\mathbf{x}}_i^3 &= [x_{31}^s, x_{32}^s, x_{33}^s] \end{aligned}$$

For each mapping pattern $\tilde{\mathbf{x}}_i^k$ in source domain, any classifiers adaptable for complete pattern is available. The \mathcal{K} pieces of sub-classification results for $\tilde{\mathbf{x}}_i^k$ are given by

$$\mathbf{P}_i^k = \Gamma(\tilde{\mathbf{x}}_i^k | \tilde{X}^s) \quad (5)$$

where $\Gamma(\cdot)$ represents the chosen classifier. \mathbf{P}_i^k is considered as a Bayesian BBA if the chosen classifier works under probability framework (e.g., K-NN [39], NB [41]), or a regular BBA with ignorance Ω returned by the classifier works under evidence theory (e.g., EK-NN [40]).

In the CTL method, we combine \mathcal{K} pieces of classification results to obtain the credal classification of incomplete patterns. Since the distances between patterns and KNNs are different, they are not equally weighted in the fusion process. Thus, discounting techniques are required for \mathcal{K} pieces of classification results. The details are given in the next section.

B. DISCOUNTING CLASSIFICATION RESULTS

For one pattern, the weighting factors of \mathcal{K} pieces of classification results correspond to the distances between the pattern $\tilde{\mathbf{x}}_i$ and its KNNs. In general, a larger distance from the pattern to the neighbor implies a less reliable estimated mapping value. i.e., a larger distance $\|\tilde{\mathbf{x}}_i, \mathbf{x}_k^t\|$ corresponds to a smaller discounting factor γ_i^k . An effective method is adopted to define the relative discounting factor γ_i^k , formally:

$$\gamma_i^k = \frac{w_i^k}{w_i^{max}} \quad (6)$$

with

$$w_i^k = e^{-\|\tilde{\mathbf{x}}_i, \mathbf{x}_k^t\|} \quad (7)$$

where $w_i^{max} = \max\{w_i^1, \dots, w_i^{\mathcal{K}}\}$. The \mathcal{K} pieces of classification results are discounted according to the discount factor γ_i^k . Afterwards, a well known discounted rule introduced by Shafer in [30] is applied here, more precisely, discounted masses of belief are obtained as follows:

$$\begin{cases} m_i^k(A) = \gamma_i^k P_i^k(A), A \subset \Omega \\ m_i^k(\Omega) = 1 - \gamma_i^k + \gamma_i^k P_i^k(\Omega). \end{cases} \quad (8)$$

where $m_i^k(\cdot)$ denotes the BBAs of different classes (focal elements) after discounting the classification results of mapping

pattern \tilde{x}_i^k in the source domain by the discounting factor γ_i^k . By doing this, one can obtain \mathcal{K} mass functions ($m_i^k(\cdot)$) for the pattern \tilde{x}_i , and they will be fused by the global fusion method we designed to obtain the final class information of \tilde{x}_i .

C. GLOBAL FUSION OF DISCOUNTED CLASSIFICATION RESULTS

After estimation and classification step, the *highest supported class* of a pattern \tilde{x}_i in one result is defined by:

$$\omega_{ij}^k = \omega_j | m_i^k(\omega_j) = \arg \max_{\omega_j \in \Omega} (m_i^k(\omega_j)) \quad (9)$$

where ω_{ij}^k denotes the highest support class ω_j of pattern \tilde{x}_i subjected to $m_i^k(\omega_j)$.

In the fusion process, we propose an adaptive fusion strategy to distinguish the discounted \mathcal{K} classification results into two following situations. For one pattern, its classification results with different $k \in \mathcal{K}$ may be either identical or different.

In order to further introduce this adaptive fusion method, we assume that the discounted \mathcal{K} classification results highest support ρ ($1 \leq \rho \leq \mathcal{K}$, $1 \leq \rho \leq c$) classes, and that ν ($1 \leq \nu \leq \rho$) classes of ρ are supported by φ ($2 \leq \varphi \leq \mathcal{K}$) discounted classification results, that is, $\rho - \nu$ classes are supported by only one discounted classification result.

The adaptive fusion strategy consists of two steps: fusion of non-conflict discounted results and global fusion of conflict discounted results.

Step 1: Fusion of non-conflict discounted results

Let's consider that the ζ ($2 \leq \zeta \leq \varphi$) classification results of the pattern \tilde{x}_i strongly support ω_{ij}^k ($j = 1, \dots, \rho, k = 1, \dots, \zeta$), indicating that the ζ classification results are not conflicting. Therefore, these results are directly fused with the simple rule as follows. The fusion results of the BBAs are given to a focal element A by

$$\tilde{m}_i^{1, \dots, \zeta}(A) = \prod_{k=1}^{\zeta} m_i^k(A), \quad A \in \Omega \quad (10)$$

The fusion results obtained from Eq. (10) need to be normalized for the convenience of credal classification. In order to the convenience of computation, we use the classical normalization given by

$$m_i^{1, \dots, \zeta}(A) = \frac{\tilde{m}_i^{1, \dots, \zeta}(A)}{\sum_{A \in \Omega} \tilde{m}_i^{1, \dots, \zeta}(A)} \quad (11)$$

The final fusion result can be obtained directly with this rule if \mathcal{K} discounted classification results of a pattern \tilde{x}_i strongly support a specific class (i.e., $\zeta = \mathcal{K}$), which indicates that the \mathcal{K} discounted classification results are consistent and non-conflicting. However, a more cautious method is essential to model conflicts between different discounted classification results for a pattern due to the invalidity of this rule in dealing with conflict information if $\zeta \neq \mathcal{K}$ (i.e., $2 \leq \zeta < \mathcal{K}$), which is introduced in the next step.

Step 2: Global fusion of conflict (discounted) results

After non-conflict discounted fusion, ρ fused (discounted) completely conflicting results for a pattern \tilde{x}_i is obtained, of which ν results are obtained from **Step 1**, and $\rho - \nu$ results are the discounted classification results. It should be noted that although the classes supported by ρ results are different, their supports for the most likely classes are different, and it is difficult to accurately classify into a small number of classes (e.g. 2 or 3 classes) for a pattern in general. Therefore, we should attribute priorities to obtain the most likely meta-class composed of the highest supported and difficult-divided-singleton classes the pattern belongs to, and generate a new framework consisting of the meta-class and singleton classes for the pattern. The most likely meta-class for a pattern can be obtained by a threshold parameter ϵ , defined as follows:

$$\psi_{\tilde{x}_i} = \{\omega_j \cup \dots \cup \omega_t | Z - Y \leq \epsilon\}, \quad 1 \leq j, t \leq c, j \neq t \quad (12)$$

in which

$$\begin{cases} Z = \max\{m(\omega_{ij}^k), \dots, m(\omega_{it}^k)\} \\ Y = m(\omega_{ij}^k) \end{cases} \quad (13)$$

where ω_{ij}^k ($1 \leq k \leq \rho$) denotes one of the highest supported class ω_j for the pattern \tilde{x}_i in ρ results, and $\psi_{\tilde{x}_i}$ is the most likely meta-class of the pattern \tilde{x}_i , which is composed of the highest supported and difficult divided singleton classes (such as ω_j, ω_t , etc). For a specific pattern \tilde{x}_i , the new global fusion rule are defined as:

$$\begin{cases} m_i(A) = \frac{1}{K} \cdot \sum_{\bigcap_{i=1}^{\rho} B_i \neq \emptyset} m_i^1(B_1) \dots m_i^{\rho}(B_{\rho}), \\ \text{for } |A| = 1, \text{ with } A \in \Omega \\ m_i(A) = \frac{1}{K} \cdot \prod [m(\omega_{ij}^k) \dots m(\omega_{it}^k)], \\ \text{for } |A| \geq 2, \text{ with } A = \psi_{\tilde{x}_i} \end{cases} \quad (14)$$

subject to

$$K = \sum_{B_1 \cap \dots \cap B_{\rho} \neq \emptyset} [m_i^1(B_1) \dots m_i^{\rho}(B_{\rho})] + \prod [m(\omega_{ij}^k) \dots m(\omega_{it}^k)] \quad (15)$$

where K is the normalization factor, $|A|$ is the number of singleton elements included in A . It is not difficult to find Eq.(14) that the precision of classification of one pattern (i.e. whether the pattern is classified into meta-class or not) depends mainly on the parameter ϵ . Parameter ϵ is a conflict measure factor, which essentially characterizes the degree of conflict between different evidences (classification results). ϵ effects the number of meta-classes in the fusion process, in order to reduce the risk of misclassification, all subsets of set $\psi_{\tilde{x}_i}$ are retained and the corresponding conflict information is assigned to meta-classes. Here is the guideline given for adjusting the parameter ϵ as follow.

1) Guideline for Choosing the Parameter ϵ : In practice, the threshold ϵ is used for meta-classes selection in classification. A bigger ϵ value corresponds to a smaller number of

TABLE 1. Credal Transfer Learning (CTL) Method.

Input:
Complete source data set \tilde{X}^s , incomplete target data set \tilde{X}^t
Parameters:
\mathcal{K} : the number of neighbors (default $\mathcal{K} = 5$)
ϵ : threshold of meta-class ($\epsilon \in [0, 1]$)
for
Searching for KNNs of pattern \tilde{x}_i in X ;
Estimate \mathcal{K} mapping patterns \tilde{x}_i^k ;
Classify \mathcal{K} mapping patterns \tilde{x}_i^k by Eq.(5);
Discount \mathcal{K} classification result by Eqs.(6)-(8);
Select the most possible meta-class by Eqs.(9)-(13);
Fuse \mathcal{K} results adaptively by Eqs.(10)-(15).
end

are mis-classified patterns, as well as a more ambiguous classification result, i.e., more patterns belong to meta-classes. A small ϵ value results in fewer patterns in the meta-classes, but may cause more misclassifications for imprecise patterns. Therefore, ϵ should be tuned according to the adapted imprecision degree. In this paper, a proper interval $\epsilon \in [0, 0.3]$ is recommended, and $\epsilon = 0.1$ is regarded as a default value in most situations.

For the convenience of implementation, the CTL method is outlined in Algorithm 1.

IV. EXPERIMENT APPLICATIONS

In this section, we test and evaluate CTL method through extensive experiments on twelve real data sets from UCI repository [16] and five public high dimensional data sets (i.e., MNIST + USPS, COIL20 and Office + Caltech). In order to fully justify the CTL method, we consider the different ways of combining the classical missing value estimation methods and with the traditional transfer learning methods. The imputation methods and transfer learning methods used in the comparison methods are listed as follows.

• Imputation Methods:

1) Mean Imputation (MI) [19]: In MI, the missing values are replaced using the mean value of the same attribute of the data set in the target domain.

2) K -Nearest Neighbor Imputation (KNNI) [20], [21]: In KNNI, the missing values are estimated by KNNs of the patterns in the target domain.

3) Locally Linear Approximation (LLA) [25]: In LLA, the missing values are estimated using KNNs with optimal weights obtained by the locally linear reconstruction.

• **Transfer Learning Methods:** 1) Single Value Mapping-Based Transfer Learning (SVMTL) [34]: In SVMTL, only one mapping value is found for each incomplete pattern with estimation in the target domain, which means that when we find the nearest neighbor of a pattern, its corresponding pattern in the source domain is directly taken as the mapping value.

TABLE 2. Basic information of the used data sets.

Data	#Class	#Attr.	N_s	N_t	#Inst.
Ionosphere (Io)	2	34	16	18	351
Magic (Ma)	2	10	5	5	19020
Pen-Based Recognition (Pen)	10	16	7	9	10992
Spambase (Sp)	2	57	28	29	4597
Red wine quality (Rwq)	6	11	5	6	1599
White wine quality (Wwq)	7	11	5	6	4898
Vehicle (Ve)	4	18	8	10	846
Connections (Con)	11	10	5	5	990
Wall-following robot (Wall)	4	24	11	13	5456
Contraceptive (Co)	3	9	4	5	1473
Segment (Seg)	7	19	9	10	2310
Movement-libras (MI)	15	90	44	46	360

2) Weighted Mapping-Based Transfer Learning (WMTL) [34]: In WMTL, KNNs are found for each incomplete pattern with estimation in the target domain, then the KNNs corresponding patterns in the source domain are weighted to synthesize a new pattern as the mapping value according to the distance between pattern and KNNs.

3) Transfer Component Analysis (TCA) [13]: In TCA, patterns in the source domain are mapped to patterns in the target domain in the reproducing kernel Hilbert space, where common latent features of similar marginal distribution are defined on the two domains.

4) Joint distribution adaptation (JDA) [14]: In JDA, *Maximum Mean Discrepancy* (MMD) is applied to measure the difference between marginal and conditional distributions, and a new feature representation is constructed to train classifiers.

In experiments, the performance of CTL method is compared with twelve different combination methods, i.e., MI + SVMTL (MSTL), KNNI + SVMTL (KSTL), LLA + SVMTL (LSTL), MI + WMTL (MWTL), KNNI + WMTL (KWTL), LLA + WMTL (LWTL), MI + TCA (MTCA), KNNI + TCA (KTCA), LLA + TCA (LTCA), MI + JDA (MJDA), KNNI + JDA (KJDA) and LLA + JDA (LJDA).

In this paper, the K -Nearest Neighbor (K-NN) [39], Evidence K -Nearest Neighbor (EK-NN) [40], Naive Bayesian (NB) and Adaboost [41], [45] classifier are employed as basic classifiers respectively. $K = 5$ is default in KNNs, K-NN and EK-NN, and the parameters of EK-NN are automatically optimized by the method introduced in [40].

In our simulations, the misclassification is declared (counted) for one pattern truly originated from ω_i if it is classified into A with $\omega_i \cap A = \emptyset$. If $\omega_i \cap A \neq \emptyset$ and $A \neq \omega_i$ then it will be considered as an imprecise classification. The error rate denoted by R_e is calculated by $R_e = U_e/N$, where U_e is number of misclassification errors, and N is the number of patterns in target domain. The imprecision rate denoted by R_i is calculated by $R_i = U_i/N$, where U_i is number of patterns committed to the meta-classes. The experiment is conducted with Matlab software.

TABLE 3. Classification results of different methods with different ϵ values (In %).

Data	n	MSTL R_e	KSTL R_e	LSTL R_e	MWTL R_e	KWTL R_e	LWTL R_e	CTL($\epsilon=0.05$) $[R_e, R_i]$	CTL($\epsilon=0.1$) $[R_e, R_i]$	CTL($\epsilon=0.15$) $[R_e, R_i]$	CTL($\epsilon=0.2$) $[R_e, R_i]$
Io	3	25.74	23.17	24.41	30.96	29.34	30.77	[22.32, 7.98]	[20.32, 11.49]	[18.99, 13.96]	[17.19, 16.52]
Io	6	28.68	26.02	27.16	32.10	29.44	31.53	[23.84, 6.93]	[21.56, 10.45]	[19.85, 13.49]	[18.14, 16.52]
Ma	1	34.56	34.16	34.28	34.55	33.84	33.88	[33.23, 0.03]	[33.21, 0.07]	[33.18, 0.12]	[33.17, 0.15]
Ma	2	35.66	35.92	36.04	35.87	35.90	35.71	[34.06, 0.10]	[34.02, 0.21]	[33.97, 0.33]	[33.95, 0.41]
Pen	2	34.26	21.42	24.91	33.94	20.98	24.49	[20.75, 0.18]	[20.69, 0.31]	[20.65, 0.42]	[20.58, 0.54]
Pen	4	53.05	26.28	35.64	53.27	26.03	35.52	[25.96, 0.37]	[25.87, 0.55]	[25.75, 0.79]	[25.61, 1.07]
Sp	5	27.86	25.84	26.73	36.57	33.35	34.84	[17.87, 11.57]	[15.50, 17.58]	[13.56, 23.41]	[11.70, 29.72]
Sp	10	32.71	26.70	29.68	39.02	33.55	37.23	[19.11, 14.79]	[16.87, 21.12]	[14.13, 27.83]	[12.03, 35.19]
Rwq	1	57.05	55.91	56.76	57.56	56.42	56.46	[53.17, 1.68]	[52.22, 3.33]	[51.47, 4.58]	[50.73, 5.69]
Rwq	2	61.34	57.33	59.54	57.80	58.50	59.28	[54.61, 3.16]	[53.32, 5.16]	[52.11, 7.13]	[50.84, 9.55]
Wwq	1	62.30	61.73	61.97	63.02	61.58	61.83	[57.25, 1.79]	[56.54, 2.98]	[55.72, 4.43]	[54.83, 5.74]
Wwq	2	66.26	61.91	63.10	63.99	63.37	63.51	[57.99, 2.85]	[56.42, 5.19]	[54.92, 7.65]	[53.52, 10.10]
Ve	3	54.77	51.38	53.43	54.65	50.83	53.62	[48.19, 0.08]	[48.15, 0.20]	[48.07, 0.28]	[48.07, 0.28]
Ve	5	62.33	57.68	59.22	61.94	56.82	58.79	[52.80, 0.20]	[52.76, 0.35]	[52.64, 0.47]	[52.52, 0.71]
Con	1	69.06	63.54	65.62	81.01	79.16	79.70	[57.54, 19.12]	[46.77, 31.45]	[38.18, 41.58]	[29.79, 52.69]
Con	2	74.58	68.08	72.86	82.96	80.03	82.90	[58.99, 20.77]	[46.67, 35.49]	[37.21, 27.83]	[28.29, 56.03]
Wall	2	32.31	28.88	29.85	50.06	47.23	47.18	[23.36, 6.20]	[21.98, 9.70]	[20.91, 12.61]	[19.90, 15.93]
Wall	4	43.33	33.94	38.41	52.73	45.77	49.80	[24.67, 9.48]	[22.75, 14.20]	[21.06, 18.27]	[19.46, 22.49]
Co	1	60.42	60.51	60.74	62.23	61.35	62.43	[57.61, 3.67]	[54.92, 8.67]	[53.16, 11.36]	[48.93, 18.19]
Co	2	62.64	63.91	65.38	62.34	61.71	62.37	[57.23, 2.63]	[56.35, 3.91]	[55.28, 5.77]	[49.38, 14.53]
Seg	3	56.48	40.40	46.58	62.74	54.46	56.26	[38.63, 0.63]	[38.37, 1.27]	[37.98, 1.99]	[37.63, 2.70]
Seg	5	64.66	40.97	48.73	67.46	56.51	58.90	[38.73, 0.97]	[38.28, 1.80]	[37.82, 2.71]	[37.14, 3.69]
MI	10	72.41	70.09	70.46	87.78	87.31	86.85	[64.35, 18.61]	[48.89, 35.56]	[38.52, 47.22]	[28.70, 58.70]
MI	20	74.72	73.43	72.69	87.50	87.13	87.31	[65.93, 15.19]	[52.50, 30.09]	[38.70, 45.28]	[30.83, 53.98]
Ave		51.97	46.22	48.51	56.34	52.11	53.80	[42.00, 6.21]	[38.95, 10.46]	[36.41, 14.10]	[33.87, 17.96]

A. EXPERIMENT 1

Twelve well-known UCI data sets are used to test the performance of CTL, and The basic information of the used data sets including number of classes (#Class.), attributes (#Attr.) and instances (#Inst.) are shown in Table 2. We divide the attributes of each data set into two parts corresponding to the source domain and target domain, to fit our transfer learning scenario. For instance, if a data set has 15 attributes, and we take 7 attributes as the source domain attributes, while the rest 8 attributes are regarded as target domain. We segment each data set into three partitions by the following steps.

- 1) In the two domains, 5% one-to-one corresponding patterns pairs are selected as bridge.
- 2) The rest patterns in the source domain are labeled training patterns.
- 3) The remaining patterns in the target domain are the test patterns with missing values, in which the test patterns randomly lose n attributes.

Table 2 shows the number of attributes of source domain and target domain, which are expressed as N_s and N_t respectively. Our CTL method and other comparison methods are used to classify test patterns with missing values in the target domain. Here, ten sets of source and target domain are randomly generated for the same data set, and the average values of the evaluation index are reported.

In this experiment, K-NN is selected as basic classifier. The average error rate R_e and imprecision rate R_i (for CTL) of the different methods with different meta-class threshold of ϵ (i.e., $\epsilon = 0.05, 0.1, 0.15, 0.2$), are given in Table 3.³

³The number of dimensions of each data set in target domain is shown in Table 2, and the n value in Table 3 is the number of missing dimensions in target domain

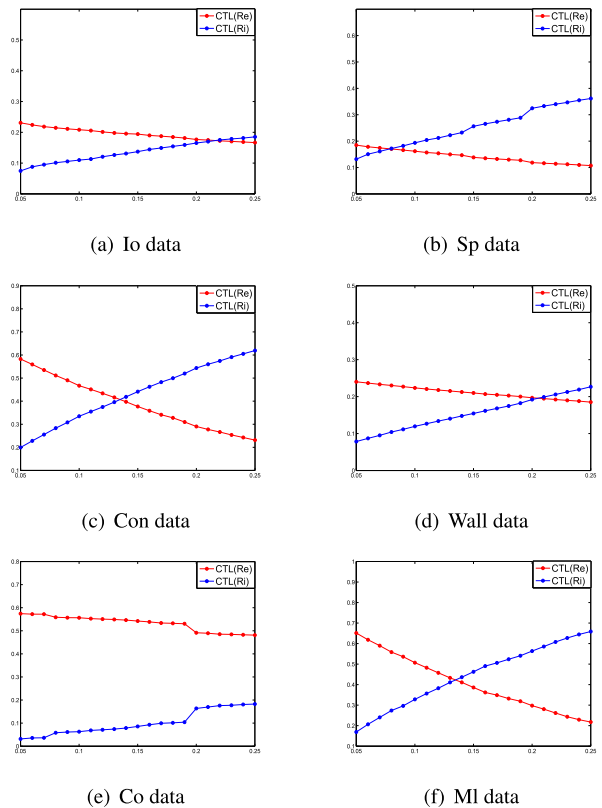


FIGURE 1. Classification results of different ϵ with K-NN classifier.

One can see from Table 3 that the CTL method generally yields lower error rate than other methods, but meanwhile some imprecision are appeared in the classification

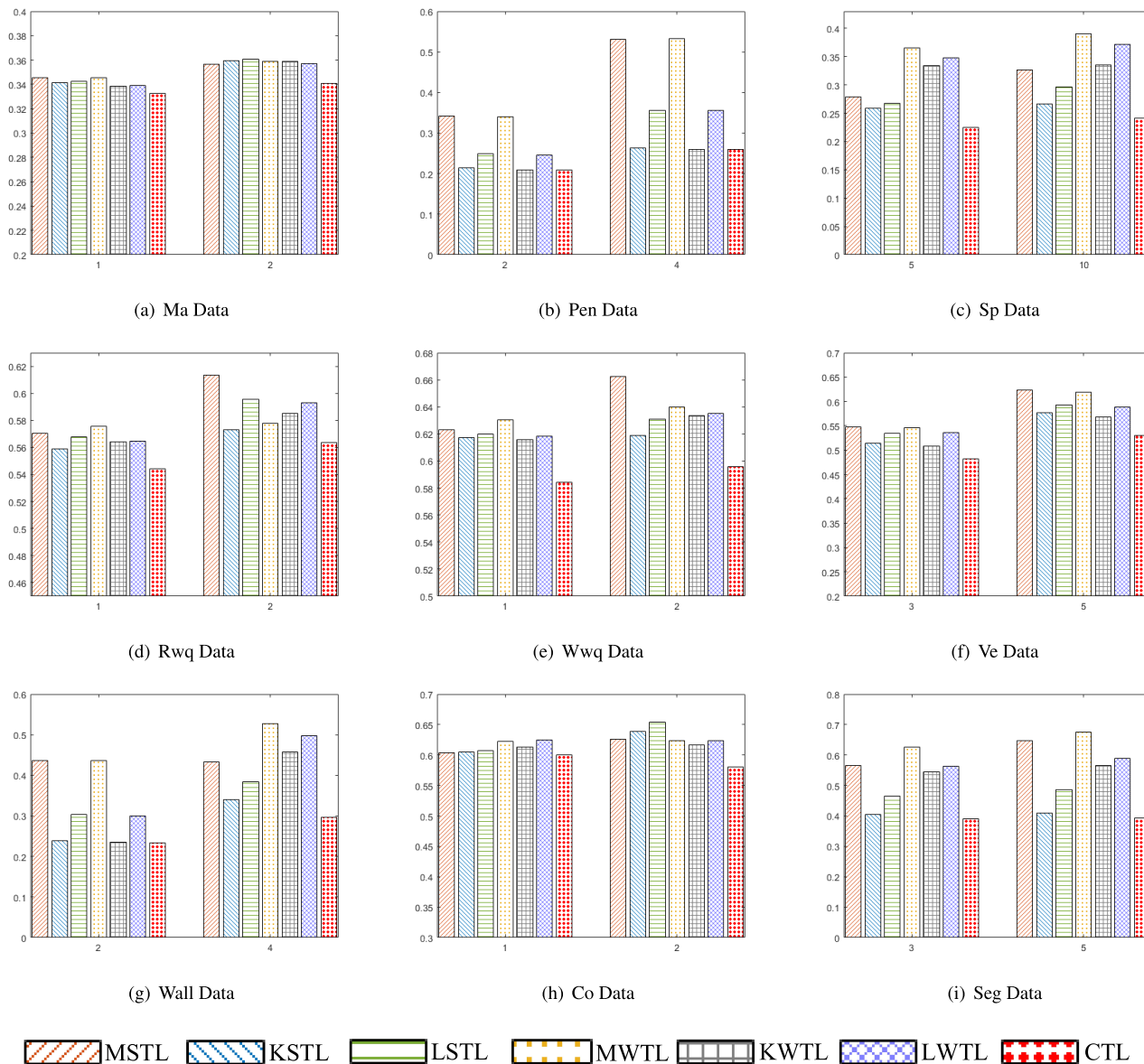


FIGURE 2. Classification results of different methods with K-NN classifier.

result due to bringing in meta-classes, which indicates that some incomplete patterns are very difficult to classify due to lack of attributes information. It is worth noting that the increase of the number of missing values (i.e., n) in target domain generally results in the increase of error rate, and the increment of imprecision in CTL, since the more missing values cause the bigger uncertainty in classification problem. So the credal classification fusion method, which includes meta-classes in CTL, is very effective in characterizing the imprecise degree, and helps to reduce the rate of misclassification. In CTL, with the increases of ϵ from $\epsilon = 0.05$ to $\epsilon = 0.2$, which causes the decrease of error rate but meanwhile it brings the increase of imprecision rate. The results show that the pattern attributes

used are not enough to accurately classify the patterns in meta-class. If one wants to obtain more precise classification results, some other source information with complementary characteristics.

Figure 1 shows the effect of different ϵ on CTL classification results, where the x-axis denotes the meta-class threshold ϵ , ranging from 0.05 to 0.25, and the y-axis represents the average of the classification results with scale of [0,1]. One can intuitively find that with the meta-class threshold ϵ increasing, the error rate of CTL method tends to decrease, whereas the imprecision rate tends to increment, which is consistent with the trend in Table 3. Figure 2 shows the average classification error rate of CTL with respect to other methods when the meta-class threshold $\epsilon = 0$,

TABLE 4. Classification results of different methods with EK-NN classifier (In %).

Data	n	MSTL R_e	KSTL R_e	LSTL R_e	MWTL R_e	KWTL R_e	LWTL R_e	CTL $[R_e, R_i]$
Io	3	25.74	23.17	24.41	30.96	29.34	30.77	[19.18, 13.01]
Io	6	28.68	26.02	27.16	32.00	29.34	31.43	[20.51, 11.97]
Ma	1	34.56	34.16	34.28	34.55	33.84	33.88	[33.05, 0.58]
Ma	2	35.66	35.92	36.04	35.87	35.90	35.71	[34.02, 0.70]
Pen	2	34.08	21.40	24.86	33.62	20.83	24.33	[20.69, 0.71]
Pen	4	53.22	26.28	35.74	53.18	25.90	35.42	[25.76, 1.02]
Sp	5	27.86	25.84	26.73	36.32	33.00	35.27	[11.53, 35.86]
Sp	10	32.71	26.70	29.68	38.86	33.20	37.38	[11.58, 39.90]
Rwq	1	56.88	56.02	56.54	57.86	56.93	56.69	[50.97, 6.62]
Rwq	2	61.53	58.37	60.51	58.29	59.07	59.86	[50.37, 12.88]
Wwq	1	61.62	61.15	61.34	62.30	60.97	61.10	[54.18, 7.13]
Wwq	2	65.65	61.14	62.56	62.99	62.43	62.67	[51.14, 14.72]
Ve	3	55.20	51.69	53.70	54.96	51.14	53.82	[48.19, 0.35]
Ve	5	62.49	57.57	59.50	62.10	56.86	59.18	[52.44, 0.71]
Con	1	68.01	62.53	64.55	80.71	78.75	79.63	[38.18, 40.34]
Con	2	73.97	67.47	72.15	82.93	79.93	82.63	[37.51, 46.26]
Wall	2	32.29	28.80	29.81	49.78	46.91	46.85	[20.07, 15.33]
Wall	4	43.40	34.20	38.53	52.10	44.83	49.13	[20.14, 22.38]
Co	1	60.19	60.44	60.47	60.35	61.05	61.51	[47.09, 20.73]
Co	2	60.85	63.63	60.60	60.74	60.53	60.99	[48.18, 18.35]
Seg	3	57.33	40.62	47.46	62.96	54.98	56.54	[37.92, 2.86]
Seg	5	65.24	41.23	49.87	67.95	57.04	59.22	[38.20, 3.48]
MI	10	71.94	69.54	70.09	87.87	86.85	86.39	[53.15, 29.26]
MI	20	75.00	73.70	72.87	87.96	87.04	87.87	[50.56, 31.48]
Ave		51.84	46.15	48.31	56.13	51.94	53.68	[36.44, 15.69]

TABLE 5. Classification results of different methods with NB classifier (In %).

Data	n	MSTL R_e	KSTL R_e	LSTL R_e	MWTL R_e	KWTL R_e	LWTL R_e	CTL $[R_e, R_i]$
Ma	1	31.44	31.42	31.30	31.24	31.24	31.07	[28.47, 10.08]
Ma	2	31.88	32.22	31.73	31.65	32.00	31.53	[30.40, 5.15]
Pen	2	48.42	37.61	42.08	46.62	35.90	40.20	[36.39, 5.25]
Pen	4	62.20	41.12	50.72	61.26	39.26	48.97	[40.32, 3.12]
Sp	5	40.72	42.51	41.44	35.58	38.14	36.47	[31.17, 22.78]
Sp	10	41.06	41.43	41.03	35.25	36.23	36.49	[31.40, 24.61]
Rwq	1	60.68	59.75	60.00	57.22	56.06	56.72	[54.32, 3.08]
Rwq	2	63.87	63.97	62.51	62.30	60.90	60.96	[58.18, 6.17]
Wwq	1	62.51	63.16	62.69	64.37	65.61	65.32	[58.66, 2.84]
Wwq	2	64.66	63.63	64.00	66.48	65.67	66.26	[58.31, 5.72]
Ve	3	66.59	62.02	64.38	66.51	61.47	63.67	[54.69, 15.17]
Ve	5	70.92	63.67	68.20	70.45	63.83	67.42	[56.38, 12.45]
Con	1	76.73	73.84	75.35	85.52	84.21	85.25	[55.02, 27.21]
Con	2	80.54	75.29	79.80	86.20	83.70	86.60	[57.85, 27.07]
Wall	2	50.93	52.24	51.15	54.80	54.51	54.48	[44.56, 6.61]
Wall	4	54.03	52.79	53.78	53.60	52.71	53.21	[44.50, 8.97]
Co	1	66.51	65.40	65.90	62.98	61.62	62.96	[57.21, 10.41]
Co	2	65.24	63.32	64.25	65.13	59.95	62.86	[57.66, 8.37]
MI	10	76.57	74.91	74.54	85.46	84.91	84.44	[48.24, 35.46]
MI	20	76.67	74.81	75.00	89.26	88.06	88.06	[48.24, 35.65]
Ave		59.61	56.76	57.99	60.59	57.80	59.15	[47.60, 13.81]

where the x-axis denotes the number of missing attributes and the y-axis denotes the error rate. In this situation, the CTL method only obtains a specific result. One can find that the CTL method still has a significant effect on these data sets. In real applications, the parameter ϵ should be tuned according to the imprecision rate one can accept in the classification. The CTL method allows the patterns that are really difficult to be classified correctly to be assigned to proper meta-class, and they should be cautiously treated in applications.

B. EXPERIMENT 2

In this experiment, we use the twelve real data sets in Table 2 to evaluate the performance of CTL with respect to MSTL, KSTL, LSTL, MWTL, KWTL and LWTL. EK-NN, Adaboost and NB⁴ are selected as basic classifiers and the meta-class threshold $\epsilon = 0.1$ is selected here. The average error rate R_e and imprecision rate R_i (for CTL) of different

⁴Naive Bayes is not applicable to Io and Seg data sets, because the within-class class variance of several attributes is not positive.

TABLE 6. Classification results of different methods with Adaboost classifier (In %).

Data	n	MSTL	KSTL	LSTL	MWTL	KWTL	LWTL	CTL
		R_e	R_e	R_e	R_e	R_e	R_e	$[R_e, R_i]$
Io	3	25.74	23.17	24.41	29.72	28.21	29.82	[19.18, 13.01]
Io	6	28.68	26.02	27.16	31.62	28.58	31.05	[20.51, 11.97]
Ma	1	32.00	31.19	31.65	31.26	30.47	30.90	[29.62, 2.20]
Ma	2	32.85	32.44	32.98	32.20	31.75	32.16	[30.37, 1.15]
Pen	2	76.78	74.47	75.60	76.34	74.16	75.10	[66.18, 11.96]
Pen	4	79.73	75.29	77.11	79.58	74.97	76.75	[69.81, 7.47]
Sp	5	20.29	18.08	19.82	18.25	16.95	17.85	[12.11, 14.01]
Sp	10	22.48	17.95	21.19	19.76	16.40	18.90	[11.58, 20.07]
Rwq	1	55.91	55.34	55.21	56.21	55.78	55.84	[53.70, 1.68]
Rwq	2	56.63	56.19	56.21	57.31	56.48	56.40	[54.78, 3.78]
Wwq	1	54.65	54.95	54.72	54.29	54.63	54.62	[54.44, 0.52]
Wwq	2	55.14	55.48	55.37	54.92	54.99	54.91	[54.67, 0.81]
Ve	3	58.59	54.33	56.78	58.63	54.37	56.70	[53.78, 0.32]
Ve	5	64.26	59.10	63.59	63.99	59.38	62.96	[56.78, 0.51]
Con	1	78.45	78.32	77.95	85.66	85.02	85.59	[31.35, 57.37]
Con	2	82.73	81.78	83.06	87.10	85.12	87.88	[32.29, 59.29]
Wall	2	50.63	50.46	50.33	52.58	52.78	52.72	[48.99, 2.24]
Wall	4	54.93	52.12	53.53	56.21	55.48	56.07	[49.65, 3.55]
Seg	3	70.75	64.19	68.30	70.43	64.49	66.18	[64.52, 0.40]
Seg	5	73.39	63.97	66.91	74.10	63.81	66.25	[64.91, 0.13]
MI	10	88.70	88.70	88.89	91.11	90.83	90.74	[40.00, 53.15]
MI	20	87.41	87.04	87.69	90.65	90.37	90.74	[45.19, 46.76]
Ave		56.85	54.57	55.84	57.81	55.68	56.82	[43.84, 14.19]

methods with different basic classifiers (i.e., EK-NN [40], NB [41], Adaboost [45]), are reported in Tables 4-6.

In Tables 4-6, we can see that error rates of CTL method with EK-NN, Adaboost and NB classifiers are smaller than the other applied methods in most situations. In parallel, some incomplete patterns that are very difficult to classify into a specific class have been submitted to the meta-classes. With the number of missing values n increases, it may cause the increment of error rates in the classifiers, and the imprecision rate generally becomes higher in CTL, which is reasonable. In the process of credal transfer learning, meta-classes are introduced to reasonably characterize the imprecision caused by missing values, so the proposed method is able to effectively reduce classification error. The average error rate and imprecision rate denoted by Ave of different methods on different data sets with the same classifier is given in the last row of Tables 4-6 to express the general performance of the corresponding method. It can be seen that CTL method has good adaptability in three basic classifiers: EK-NN, Adaboost and NB. In other words, CTL method has good robustness and can be applied to various basic classifiers. However, in the situation of large amount of data, we find that NB classifier takes less time than EK-NN and Adaboost, because EK-NN and Adaboost classifiers will bring heavy computational burden.

C. EXPERIMENT 3

In this experiment, we adopted five public high dimensional data sets: MNIST + USPS, COIL20 and Office + Caltech. These data sets have been widely used in most of the existing TL works. Table 7 shows the details of the data sets. MNIST (M) and USPS (U) are two handwritten

TABLE 7. Basic information of the high dimensional data sets.

Data	Domain	#Class	#Attr.	#Inst.
MNIST	M	10	256	2000
USPS	U	10	256	1800
Office	A	10	800	958
Caltech	C	10	800	1123
COIL20	CO1, CO2	20	1024	1440

digits recognition data sets that follow very different distributions. MNIST includes 60000 training images and 10000 test images, USPS contains 7291 training images and 2007 test images. The Office-Caltech data set contains 10 classes of images from four domains, and we select Amazon (A) and Caltech (C) for testing. COIL20 (CO) contains 20 classes and 1440 images, with 72 images in each class. Detailed descriptions about these data sets can be found in [14]. Here, we use $A \rightarrow B$ to express the knowledge transfer from source domain A to target domain B.

In experiment, we randomly select 10 patterns in each class from both the source domain and the target domain as one-to-one corresponding pattern pairs. Then, the rest patterns in the source domain are considered as labeled training set, and the patterns in the target domain as test set, in which each pattern randomly loses n attributes. EK-NN is selected as basic classifier, and the parameter $\epsilon = 0.1$ is selected here. In order to fully prove the validity of CTL method for high dimensional data, we also use two other classic TL methods (i.e., TCA [13] and JDA [14]) for complete patterns. In TCA and JDA, we choose RBF kernel uniformly, and the

TABLE 8. Classification results of different methods with high dimensional data sets (In %).

Data	n	MSTL R_e	KSTL R_e	LSTL R_e	KMWTL R_e	KWTL R_e	LWTL R_e	MTCA R_e	KTCA R_e	LTCA R_e	MJDA R_e	KJDA R_e	LJDA R_e	CTL $[R_e, R_i]$
M → U	30	28.17	25.33	26.89	28.56	24.94	26.44	43.33	38.44	38.78	34.98	31.67	32.54	[16.89, 12.00]
M → U	50	28.44	24.78	27.28	30.67	23.33	27.61	46.72	37.94	38.89	37.00	30.69	32.46	[17.33, 10.89]
M → U	70	29.72	23.11	27.06	31.89	22.11	27.67	54.94	37.61	39.72	44.43	29.06	32.39	[17.89, 11.44]
U → M	30	35.30	31.15	33.30	36.10	32.20	35.10	57.15	53.60	53.90	51.33	49.63	49.73	[22.25, 19.40]
U → M	50	36.35	30.25	33.80	38.40	29.75	35.00	57.75	54.10	54.85	53.42	49.10	50.27	[23.00, 19.85]
U → M	70	36.95	29.85	34.70	39.85	28.55	35.25	61.20	52.95	55.35	54.40	48.33	50.40	[23.15, 20.95]
A → C	50	69.55	68.74	69.81	71.15	70.97	70.61	68.92	68.83	69.55	68.63	68.48	69.55	[63.13, 5.25]
A → C	100	69.99	72.04	70.88	71.68	71.68	71.77	70.70	72.22	71.50	70.76	71.71	71.48	[66.16, 4.54]
A → C	200	71.42	73.73	71.95	72.40	73.55	72.93	72.22	74.18	70.88	72.31	73.85	70.94	[68.48, 4.72]
C → A	50	71.19	71.61	71.92	70.88	71.29	70.77	63.36	65.55	63.78	64.34	66.60	64.61	[62.42, 3.97]
C → A	100	71.40	73.80	71.92	71.29	73.28	71.40	63.05	67.01	65.14	64.23	67.12	64.61	[64.61, 5.22]
C → A	200	71.40	74.74	73.28	71.71	75.16	72.03	65.34	69.52	68.48	65.83	70.04	67.78	[67.12, 4.49]
CO1 → CO2	100	21.94	19.03	21.53	25.00	22.08	23.75	20.42	17.22	17.22	21.53	18.61	18.56	[17.08, 5.56]
CO1 → CO2	200	21.94	19.03	21.53	25.00	22.08	23.75	28.75	18.33	17.78	30.05	19.31	18.98	[17.64, 5.69]
CO1 → CO2	300	23.75	18.61	21.39	25.56	21.11	23.75	33.33	20.69	20.00	34.86	20.74	20.09	[17.08, 6.39]
CO2 → CO1	100	24.72	23.06	24.31	26.94	25.42	26.53	21.53	20.14	20.83	22.64	21.53	21.18	[20.69, 5.00]
CO2 → CO1	200	25.28	21.39	24.44	27.64	24.44	26.81	23.75	21.39	20.56	24.86	22.64	21.30	[21.36, 4.44]
CO2 → CO1	300	25.14	21.11	24.44	27.08	24.31	26.67	23.89	22.22	20.83	25.05	23.10	21.34	[20.42, 5.42]
Ave		42.37	40.08	41.69	43.99	40.90	42.66	48.69	45.11	44.89	46.70	43.46	43.23	[34.82, 8.62]

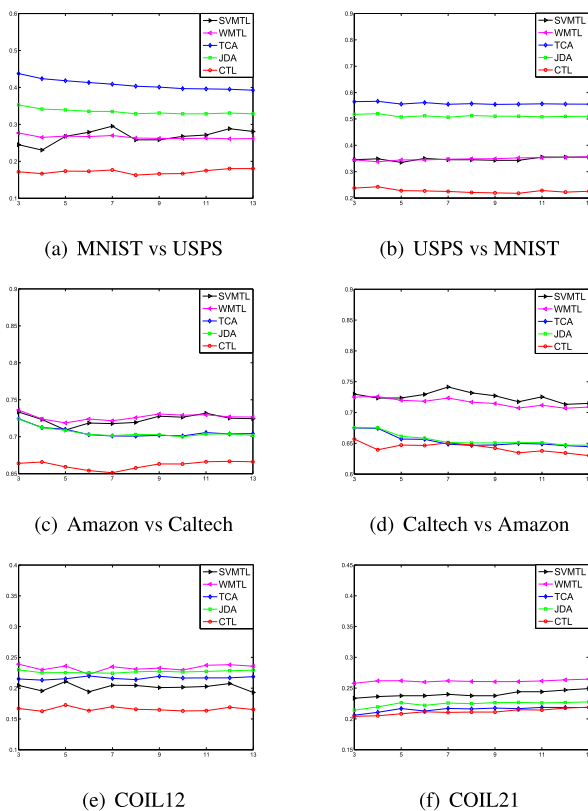


FIGURE 3. Classification results of different methods with different K values.

implementation details of TCA and JDA follow [14]. The average error rate R_e and imprecision rate R_i (for CTL) of different methods is given in Table 8.

From Table 8, we observe that CTL has better performance than the rest of methods in high dimensional applications. The average classification error rate of CTL on these data sets is 34.82%, which is 5.26% lower than best baseline method KSTL. This proves that CTL can construct a more

effective representation for the task of cross domain incomplete pattern classification.

Figure 3 shows more clearly and intuitively the influence of different K values in EK-NN classifier on classification results. The x-axis corresponds to the K value, ranging from 3 to 13, and the y-axis the average error rate in different methods, in interval of [0, 1]. SVMTL, WMTL, TCA and JDA respectively denote the average results under three imputation methods. We can observe that the error rate of CTL is always lower than that of other methods, and different K values has little impact on the classification results in CTL. This shows that CTL method has strong robustness for K values selection, which is a good feature for CTL method in practical classification applications.

V. CONCLUSION

In this paper, A new credal transfer learning (CTL) method, based on the belief function theory, is proposed to classify missing data. CTL method is able to effectively address the classification problem of missing values, which training and test sets come from different distribution domains. CTL uses observed attributes to search for K -nearest neighbors (KNNs), and estimates K versions of mapping patterns in the source domain for incomplete patterns the target domain according to some given one-to-one pattern pairs, which can effectively represent the uncertainty of estimation caused by missing values. The K pieces of classification results then are discounted by the discounting (reliable) factors depending on the distance between the corresponding KNNs and the (incomplete) pattern, and they are adaptively fused by a originally proposed method under the framework of belief function theory. The non classifiable patterns are reasonably submitted to the relative meta-class regarded as the union of some specific classes, representing classification with imprecision. The reasoning of imprecision is able to reduce the risk of error and characterize the uncertainly due to the lack of attributes information. Further technique (possibly costly) or

extra informative sources can be used if more precise results are required. Finally, CTL the effectiveness of CTL is justified by three experiments, in which comparison with other methods is executed on real data sets. The results show that CTL is able to reduce mis-classification rates, and captures and represents well the imprecision of classification caused by missing values.

In this work, we assume that some one-to-one pattern pairs are given. In some situations, however, the pattern pairs may not be available. Therefore, in the future, we will consider a more general TL method instead of using pattern pairs. In parallel, we will further study the problem of cross domain incomplete pattern classification from the perspective of deep learning and data-driven methods [46], [47].

REFERENCES

- [1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
- [2] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, pp. 1–40, Dec. 2016.
- [3] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," 2019, *arXiv:1911.02685*. [Online]. Available: <http://arxiv.org/abs/1911.02685>
- [4] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Scholkopf, "Domain adaptation with conditional transferable components," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2016, pp. 2839–2848.
- [5] C. Pan, J. Huang, J. Gong, and X. Yuan, "Few-shot transfer learning for text classification with lightweight word embedding based models," *IEEE Access*, vol. 7, pp. 53296–53304, 2019.
- [6] B. Myagmar, J. Li, and S. Kimura, "Cross-domain sentiment classification with bidirectional contextualized transformer language models," *IEEE Access*, vol. 7, pp. 163219–163230, 2019.
- [7] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 193–200.
- [8] L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for heterogeneous domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 711–718.
- [9] S. Sukhija, N. C. Krishnan, and G. Singh, "Supervised heterogeneous domain adaptation via random forests," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2039–2045.
- [10] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1134–1148, Jun. 2014.
- [11] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 668–675.
- [12] R. K. Sanodiya, J. Mathew, S. Saha, and M. D. Thalakkottur, "A new transfer learning algorithm in semi-supervised setting," *IEEE Access*, vol. 7, pp. 42956–42967, 2019.
- [13] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.
- [14] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 2200–2207.
- [15] Y. Zhang, N. Wang, S. Cai, and L. Song, "Unsupervised domain adaptation by mapped correlation alignment," *IEEE Access*, vol. 6, pp. 44698–44706, 2018.
- [16] M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [17] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: A review," *Neural Comput. Appl.*, vol. 19, no. 2, pp. 263–282, Mar. 2010.
- [18] R. J. Little and D. B. Rubin, *Statistical Analysis With Missing Data*. Hoboken, NJ, USA: Wiley, 2014.
- [19] D. J. Mundfrom and A. Whitcomb, "Imputing missing values: The effect on the accuracy of classification," *MLRV*, vol. 25, no. 1, pp. 13–19, 1998.
- [20] G. Batista and M. C. Monard, "A study of K -nearest neighbour as an imputation method," in *Proc. Int. Conf. Hybrid Intell. Syst.*, 2002, pp. 251–260.
- [21] C.-H. Cheng, C.-P. Chan, and Y.-J. Sheu, "A novel purity-based K nearest neighbors imputation method and its application in financial distress prediction," *Eng. Appl. Artif. Intell.*, vol. 81, pp. 283–299, May 2019.
- [22] J. Luengo, J. A. Sáez, and F. Herrera, "Missing data imputation for fuzzy rule-based classification systems," *Soft Comput.*, vol. 16, no. 5, pp. 863–881, May 2012.
- [23] D. Li, J. Deogun, W. Spaulding, and B. Shuart, "Towards missing data imputation: A study of fuzzy K -means clustering method," in *Proc. Int. Conf. Rough Sets Current Trends Comput.*, Uppsala, Sweden, Jun. 2004, pp. 573–579.
- [24] J. C. Bezdek, "Pattern recognition with fuzzy objective function algorithms," *Adv. Appl. Pattern Recognit.*, vol. 22, no. 1171, pp. 203–239, 1981.
- [25] J. Dai, H. Hu, Q. Hu, W. Huang, N. Zheng, and L. Liu, "Locally linear approximation approach for incomplete data," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1720–1732, Jun. 2018.
- [26] C.-B. Lu and Y. Mei, "An imputation method for missing data based on an extreme learning machine auto-encoder," *IEEE Access*, vol. 6, pp. 52930–52935, 2018.
- [27] K. Pelckmans, J. De Brabanter, J. A. K. Suykens, and B. De Moor, "Handling missing values in support vector machine classifiers," *Neural Netw.*, vol. 18, nos. 5–6, pp. 684–692, Jul. 2005.
- [28] Z. Zhang, H. Fang, and H. Wang, "A new MI-based visualization aided validation index for mining big longitudinal Web trial data," *IEEE Access*, vol. 4, pp. 2272–2280, 2016.
- [29] C. de Bodt, D. Mulders, M. Verleysen, and J. A. Lee, "Nonlinear dimensionality reduction with missing data using parametric multiple imputations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1166–1179, Apr. 2019.
- [30] G. Shafer, *A Mathematical Theory of Evidence*, vol. 1. Princeton, NJ, USA: Princeton Univ. Press, 1976.
- [31] Z.-G. Liu, Q. Pan, G. Mercier, and J. Dezert, "A new incomplete pattern classification method based on evidential reasoning," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 635–646, Apr. 2015.
- [32] Z.-G. Liu, Y. Liu, J. Dezert, and Q. Pan, "Classification of incomplete data based on belief functions and K -nearest neighbors," *Knowl.-Based Syst.*, vol. 89, pp. 113–125, Nov. 2015.
- [33] L. Jiao, X. Geng, and Q. Pan, "BPkNN: K -nearest neighbor classifier with pairwise distance metrics and belief function theory," *IEEE Access*, vol. 7, pp. 48935–48947, 2019.
- [34] Z.-G. Liu, G. Qiu, G. Mercier, and Q. Pan, "A transfer classification method for heterogeneous data based on evidence theory," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Oct. 21, 2019, doi: [10.1109/TSMC.2019.2945808](https://doi.org/10.1109/TSMC.2019.2945808).
- [35] M.-H. Masson and T. Denœux, "ECM: An evidential version of the fuzzy c -means algorithm," *Pattern Recognit.*, vol. 41, no. 4, pp. 1384–1397, Apr. 2008.
- [36] Z.-G. Su and T. Denœux, "BPEC: Belief-peaks evidential clustering," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 1, pp. 111–123, Jan. 2019.
- [37] B.-C. Chen, X. Tao, M.-R. Yang, C. Yu, W.-M. Pan, and V. C. M. Leung, "A saliency map fusion method based on weighted DS evidence theory," *IEEE Access*, vol. 6, pp. 27346–27355, 2018.
- [38] J. Xia, Y. Feng, L. Liu, D. Liu, and L. Fei, "An evidential reliability indicator-based fusion rule for Dempster–Shafer theory and its applications in classification," *IEEE Access*, vol. 6, pp. 24912–24924, 2018.
- [39] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.
- [40] L. M. Zouhal and T. Denœux, "An evidence-theoretic k -NN rule with parameter optimization," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 28, no. 2, pp. 263–271, May 1998.
- [41] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ, USA: Prentice-Hall, 2003.
- [42] R. R. Yager, "On the Dempster–Shafer framework and new combination rules," *Inf. Sci.*, vol. 41, no. 2, pp. 93–137, Mar. 1987.
- [43] D. Dubois and H. Prade, "Representation and combination of uncertainty with belief functions and possibility measures," *Comput. Intell.*, vol. 4, no. 3, pp. 244–264, Sep. 1988.
- [44] F. Smarandache and J. Dezert, "On the consistency of PCR6 with the averaging rule and its application to probability estimation," in *Proc. IEEE Int. Conf. Inf. Fusion*, Jul. 2013, pp. 1119–1126.

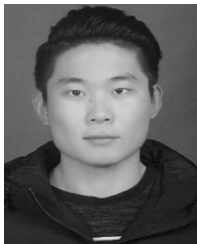
- [45] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.
- [46] S. Ding, S. Qu, Y. Xi, and S. Wan, "Stimulus-driven and concept-driven analysis for image caption generation," *Neurocomputing*, early access, Jul. 27, 2019, doi: [10.1016/j.neucom.2019.04.095](https://doi.org/10.1016/j.neucom.2019.04.095).
- [47] R. Zhang, P. Xie, C. Wang, G. Liu, and S. Wan, "Classifying transportation mode and speed from trajectory data via deep multi-scale learning," *Comput. Netw.*, vol. 162, Oct. 2019, Art. no. 106861.



ZONGFANG MA was born in Anhui, China, in 1980. He received the bachelor's and master's degrees from the Xi'an University of Architecture and Technology (XAUAT), China, in 2002 and 2006, respectively, and the Ph.D. degree from Northwestern Polytechnical University (NPU), Xi'an, China, in 2011.

He is currently an Associate Professor with the School of Information and Control Engineering, XAUAT. His current research interests include

machine vision and pattern recognition.



ZHE LIU was born in Shaanxi, China, in 1996. He received the bachelor's degree from the Xi'an University of Architecture and Technology Huaqing College, China, in 2018. He is currently pursuing the master's degree with the Xi'an University of Architecture and Technology, China.

His current research interest includes pattern recognition.



YIRU ZHANG was born in Shandong, China, in 1991. He received the bachelor's degree from the Huazhong University of Science and Technology (HUST), China, in 2013, the engineer's and master's degrees from the Université de Pierre et Marie Curie (UPMC) and Saclay University, France, in 2016, and the Ph.D. degree from Université Rennes 1, France, in 2020.

His current research interests include the imperfect information reasoning with the theory of belief function, decision theory, preference learning, and pattern recognition.



LIN SONG was born in Liaoning, China, in 1983. He received the master's degree from the Xi'an University of Architecture and Technology (XAUAT), China, in 2009, and the Ph.D. degree from Northwestern Polytechnical University (NPU), Xi'an, China, in 2015.

He is currently a Teacher with the School of Information and Control Engineering, XAUAT. His current research interests include multisource information fusion and pattern recognition.



JIHUAN HE was born in China. He is currently a Distinguished Professor with the National Engineering Laboratory for Modern Silk, College of Textile and Clothing Engineering, Soochow University, and also a Distinguished Professor with the Xi'an University of Architecture and Technology. He is an Expert on nonlinear science and nanotechnology, and he is the Owner of some famous analytical methods, such as the semi-inverse method, the variational iteration method, the homotopy per-

turbation method, the exp-function method, and He's frequency formulation. He has published more than 400 articles with an H-index of 67, he has been listed as a highly cited researcher by Clarivate Analytics and Elsevier for many years, and was also one of the World's Hottest Researchers. His present interests mainly include the fractal calculus and fractional calculus.

...