Combination of Transferable Classification with Multi-source Domain Adaptation Based on Evidential Reasoning

Zhun-ga Liu, Lin-qing Huang, Kuang Zhou, and Thierry Denœux

***Abstract*—In applications of domain adaptation, there may exist multiple source domains, which can provide more or less complementary knowledge for pattern classification in the target domain. In order to improve the classification accuracy, a decision-level combination method is proposed for the multi-source domain adaptation based on evidential reasoning. The classification results obtained from different source domains usually have different reliabilities/weights, which are calculated according to the domain-consistency. So the multiple classification results are discounted by the corresponding weights under belief functions framework, and then Dempster's rule is employed to combine these discounted results. In order to reduce errors, a neighborhood-based cautious decision making rule is developed to make the class decision depending on the combination result. The object is assigned to a singleton class if its neighborhoods can be (almost) correctly classified. Otherwise, it is cautiously committed to the disjunction of several possible classes. By doing this, we can well characterize the partial imprecision of classification, and reduce the error risk as well. A unified utility value is defined here to reflect the benefit of such classification. This cautious decision-making rule can achieve the maximum unified utility value, because partial imprecision is considered better than an error. Several real data sets are used to test the performance of proposed method, and the experimental results show that our new method can efficiently improve the classification accuracy with respect to other related combination methods.**

***Index Terms*—Evidential reasoning, belief functions, domain adaptation, evidence theory, pattern classification, cautious decision making.**

I. Introduction

In pattern classification, when there are not enough labeled training patterns in the target domain, traditional machine learning methods cannot build a reliable model. If there are abundant labeled patterns in the source domains that are related to the target domain but with different distributions or feature spaces, these patterns are expected to help classify objects in the target domain. Domain adaptation, as a special setting of transfer learning, aims to transfer knowledge in the source domain [1–3] to the target domain for improving the classification performance. The major issue of domain adaptation is how to reduce the distribution difference between the source and target domains.

Existing works can be summarized into two main categories: 1) instance re-weighting [4], which reuses samples in the source domain by some weighting techniques; 2) distribution match [5], which utilizes good feature representation to reduce the difference of distributions between domains and preserve important properties of original data. We mainly focus on distribution match in this work. Transfer Component Analysis (TCA) recently introduced by Pan et al. [5] consists in learning some transfer components and extracting new features of both domains to make the distributions close to each other. Wang et al. [6] come up with Stratified Transfer Learning (STL) considering conditional probability distribution match to obtain a new feature representation. The Joint Distribution Adaptation (JDA) [7] matches both marginal and conditional probability distributions and extracts a robust feature representation to reduce the difference between the source and target domains. Some methods considering new conditions based on matching marginal and conditional distributions have been proposed to achieve the best possible performance. A method called Visual Domain Adaptation (VDA) [8] takes into account not only the marginal and conditional probability distributions but also domain invariant clusters. Transfer Joint Matching (TJM) [9] reduces the difference of domains by jointly matching the distribution and re-weighting the instances. Balanced Distribution Adaptation (BDA) [10] considers both the importance of the marginal and conditional distribution discrepancies when minimizing the distribution distance between domains, and Weighted Balanced Distribution Adaptation (W-BDA) [10] for imbalance issues in classification is also proposed. Subspace Distribution Alignment (SDA) [11] as an extension of Subspace Alignment (SA) [12] employs subspace structure and feature alignment to obtain one robust feature representation. Sun et al. propose CORrelation ALignment (CORAL) [13] to minimize domain difference by aligning the second-order statistics of the source and target distributions. The Joint Geometrical and Statistical Alignment (JGSA) method [14] reduces the difference between domains both statistically and geometrically.

The previous methods mentioned above only consider one source domain. In applications, there may exist multiple source domains, and it has been an open problem [15–17] how to efficiently take advantage of the complementary knowledge among different source domains. For example, TrAdaBoost [18] is developed based on a different strategy compared

Manuscript received; revised.

Zhun-ga Liu and Lin-qing Huang are with the School of Automation, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: liuzhunga@nwpu.edu.cn, huanglinqing95@gmail.com).

Kuang Zhou is with the School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an 710072, China.

Thierry Denœux is with the Université de Technologie de Compiègne, CNRS, Heudiasyc, Compiègne and Institut Universitaire de France, Paris, France.

with AdaBoost [19]. It trains the base classifier using the weighted patterns in the source and target domains. At each iteration, the weights update for correctly classified patterns in the target domain is the same as that of AdaBoost, but the update is completely opposite for wrongly categorized patterns in the source domain. The MultiSource-TrAdBoost [20, 21] method is developed for solving the classification problem with multiple source domains. At every round of iteration, MultiSource-TrAdaBoost selects the individual classifier with minimum errors in these source domains as a base classifier.

The different source domains usually provide some complementary knowledge for the classification of objects (patterns) in the target domain. The fusion of such complementary knowledge is very important to improve the classification accuracy. In the MultiSource-TrAdaBoost method, the best individual classifier is selected without taking into account the complementary knowledge among multiple source domains. In this paper, we propose a new method called combination of transferable classification (CTC) with multi-source domain adaptation, which can take fully advantage of the complementary knowledge from multi-source domains to pursue the good classification performance. The main contributions of this work mainly lie in two parts: a new weighted combination method proposed for classification with multi-source domain, and a cautious decision making rule developed to reduce errors. They are briefly introduced as below:

1) A weighted decision-level combination method is proposed for dealing with multiple classification results produced by different source domains based on domain adaption techniques. When there exist multiple source domains in transfer leaning, the domain adaption technique is conducted for each source domain and target domain. Then multiple classification results can be obtained according to these multiple source domains for the object in target domain. There generally exist more or less complementary knowledge among these classification results. Evidential reasoning, which is good at managing the uncertain information, is employed here to combine these classification results. Because the classifiers learnt by using the data in different source domains can have different abilities on classification of objects in the target domain, we propose to estimate the weights of the classifiers depending on the distribution distances between the source and target domains. Then, these classification results are discounted with corresponding weights before the combination.

2) In the decision making phase, we develop a cautious decision making rule to reduce errors. It allows us to commit the object not only to singleton classes but also to the disjunction of several possible classes based on the K-nearest neighbors technique. In applications, some objects are hard to classify because of the insufficient attribute information, and such objects can be committed to the disjunction of classes. In this way, we can efficiently reduce errors in uncertain cases by properly modeling the partial imprecision.

This new CTC method is quite different from the traditional classifier fusion methods that work with different training data sets. In these traditional methods, the classifier can be directly learnt using each training data set, and the multiple classifiers are combined to obtain the classification results. In the new CTC method, the patterns in the source and target domains are represented in the same feature spaces but are assumed to be drawn from different distributions. In order to improve the classification accuracy, the domain adaptation technique must be employed here to transform these patterns into a new common feature space in which the distributions are as close as possible. Then the labeled patterns from each source domain can be used to learn a classifier for classifying the objects in target domain. When there exist multi-source domains, we propose to estimate the weights of the classification results (w.r.t. multi-source domains) based on a new measure of distribution distance (domain consistency). After that, the classification results discounted with the weights are combined by evidential reasoning method.

This paper is organized as follows. In section II, basic information about transfer learning and evidential reasoning is briefly introduced. The weighted combination method and cautious decision making rule are proposed in section III. The experimental applications are reported in section IV. Section V concludes this paper.

II. Background Knowledge

Transfer learning has been successfully applied to solve the pattern classification problem with few labeled or without labeled patterns. In applications, multiple source domains are often available. The fusion of complementary knowledge in these multiple source domains can efficiently improve the classification accuracy. Evidential reasoning (ER), as an effective tool to represent and combine uncertain information will be employed here to combine multi-source information. In this section, we will briefly introduce some basic concepts and notations for transfer learning and evidential reasoning.

A. Brief Introduction to Transfer Learning

Transfer learning aims to use the knowledge in the source domain to improve the classification performance in the target domain. It has been successfully applied in many applications, e.g., cross-domain image classification [22], image clustering [23], remote sensing image classification [24], indoor WiFi localization [4] and so on.

In transfer learning, there are two important concepts: *domain* and *task*. Domain $\mathcal{D}$ has two components: a feature space $\mathcal{X}$ and a marginal probability distribution $P(\boldsymbol{X})$. Similarly, task $\mathcal{T}$ consists of two elements: a label space $\mathcal{Y}$ and a prediction function $f(\cdot)$. This function is used to predict the corresponding label, $f(\boldsymbol{x})$, of one query object $\boldsymbol{x}$. The domain and task are denoted by $\mathcal{D} = \{\mathcal{X}, P(\boldsymbol{X})\}$ and $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$, where X is the set of patterns $\boldsymbol{x}$, and Y is the set of corresponding label y, i.e., $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\} \in \mathcal{X}$ and $Y = \{y_1, y_2, \cdots, y_n\} \in \mathcal{Y}$. We take an image classification task with SURF feature[1] as an example to explain the meanings of these notations: $\mathcal{X}$ and $\mathcal{Y}$ are the image feature space and object class space, respectively; $\boldsymbol{x}_i$ is one feature vector (pattern), and y_i is the corresponding label, i.e., $(\boldsymbol{x}_i, y_i)$ is an image pattern pair; $\boldsymbol{X}$ is the set of all image

[1]SURF feature means the image feature extracted by SURF algorithm.

feature vectors (patterns), and $P(\boldsymbol{X})$ stands for probability distribution. The domain and task with few or without labeled patterns is denoted by target domain $\mathcal{D}_T$ and target task $\mathcal{T}_T$, while the related domain and task are described as source domain $\mathcal{D}_S$ and source task $\mathcal{T}_S$. After introducing the concepts and notations for domain and task, the definition of transfer learning is described as follows.

Definition [25]: Given a source domain $\mathcal{D}_S$ and source task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and target task $\mathcal{T}_T$, transfer learning aims to use the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$ to help improve the learning of prediction function $f_T(\cdot)$ in $\mathcal{D}_T$, where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$.

From the domain perspective, there are two cases of $\mathcal{D}_S \neq \mathcal{D}_T$ as

$$\begin{cases} \mathcal{X}_S \neq \mathcal{X}_T, P(X_S) \neq P(X_T) \\ \mathcal{X}_S = \mathcal{X}_T, P(X_S) \neq P(X_T). \end{cases} \tag{1}$$

The first case with different feature spaces is referred to heterogeneous transfer learning, and the second is called homogeneous transfer learning or domain adaptation. For details, one can refer to papers [25–28].

In this work, we mainly focus on the classification problem in the second case, i.e., domain adaptation. The distribution of patterns in the source domain (training data) is different from that of objects in the target domain (test data), i.e., training and test data do not satisfy independent and identically distributed (i.i.d.) assumption. The classification performance on test data could be poor if directly using the training data. The influence of distribution is shown in Fig. 1. It seems not very reasonable to directly use labeled data in the source domain to classify the data in the target domain. The distribution match should be done before the classification procedure to make distributions of the source and target domains close to each other.

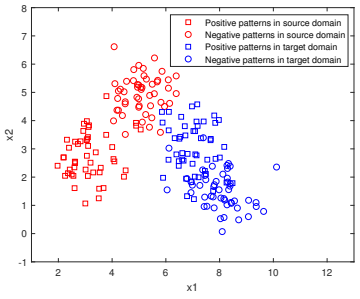


Fig. 1. Patterns in the source and target domains with different distributions.

B. Basics of Evidential Reasoning

Evidential reasoning (ER), also known as evidence theory, Dempster-Shafer theory (DST) or belief functions (BFs) was proposed by Dempster [29] and developed by Shafer [30]. It is widely used in real applications [31], such as classification [32, 33], clustering [34], information fusion [35, 36], decision making [37–41], etc.

Let $\Omega = \{\omega_1, \omega_2, \cdots, \omega_c\}$ be a finite set of mutually exclusive and exhaustive hypotheses about some problem domain, and the Ω is called the frame of discernment [42]. In pattern classification, the element ω_i can be considered as the i-th category in a c-class classification problem, and Ω is the label space. The power-set denoted by 2^Ω is the set of all subsets of Ω, and the cardinality of power-set is $2^{|\Omega|}$, e.g., if the frame of discernment is $\Omega = \{\omega_1, \omega_2, \omega_3\}$, then $|\Omega| = 3$, $2^{|\Omega|} = 8$ and $2^\Omega = \{\emptyset, \{\omega_1\}, \{\omega_2\}, \{\omega_3\}, \{\omega_1, \omega_2\}, \{\omega_1, \omega_3\}, \{\omega_2, \omega_3\}, \Omega\}$.

The *Basic belief assignment (BBA)* also called *mass function* $m(\cdot)$ is a mapping from 2^Ω to $[0, 1]$. It satisfies the condition:

$$\begin{cases} \sum\limits_{A \in 2^\Omega} m(A) = 1 \\ m(\emptyset) = 0, \end{cases} \tag{2}$$

where $m(A)$ measures the belief that one is willing to commit exactly to A, and not to any of its subsets. If $m(A) > 0$, A is a focal element. In pattern classification, if A is a single class ω_i, $m(A)$ represents the support degree of object associated to class ω_i. If A is a set of classes (e.g., $A = \{\omega_i, \omega_j\}$), $m(A)$ is used to reflect the imprecise (partial ignorance) degree among classes ω_i and ω_j. The quantity $m(\Omega)$ stands for the total ignorance degree, and it plays a neutral role in the combination. The *belief function* $Bel(\cdot)$ and *plausibility function* $Pl(\cdot)$ are also defined in [30] to represent the upper and lower of probability associated with BBA, respectively.

Dempster's rule (called DS rule for short) for combining two pieces of evidence is defined by

$$m_1 \oplus m_2(A) = \frac{\sum\limits_{B \cap C = A | B, C \in 2^\Omega} m_1(B) m_2(C)}{1 - \sum\limits_{B \cap C = \emptyset | B, C \in 2^\Omega} m_1(B) m_2(C)}, \tag{3}$$

and $m(\emptyset) = 0$. The conflict between two pieces of evidence is defined by

$$k_{12} = \sum_{B \cap C = \emptyset | B, C \in 2^\Omega} m_1(B) m_2(C). \tag{4}$$

The DS rule is associative, and the combination order has no influence on the final combination result of multiple (more than two) pieces of evidence.

III. Pattern Classification via Combining Information in Different Source Domains

In this section, we will present the proposed method for the combination of transferable classification (CTC) with multi-source domain adaptation. Let us assume there exist n source domains $\mathcal{D}_{S_i}, i = 1, \cdots, n$ with many labeled patterns and one target domain $\mathcal{D}_T$ without labeled objects. The source and target domains are represented in the same feature space but are drawn from different probability distributions. This situation is formalized by

$$\begin{cases} \mathcal{X}_{S_1} = \cdots = \mathcal{X}_{S_n} = \mathcal{X}_T \\ P(X_{S_1}) \neq \cdots \neq P(X_{S_n}) \neq P(X_T). \end{cases} \tag{5}$$

In the existing domain adaptation methods [5–10], a mapping matrix is usually learnt to map the patterns in the source domain and the objects in the target domain into a new common feature space in which the distributions of the source and target domains become close to each other. Labeled patterns from the source domain are regarded as training data to learn a classifier, which can be employed

to classify the query objects in the common space. Each source domain can produce one piece of evidence about the classification for the object in the target domain. The different source domains generally provide more or less complementary knowledge for classifying an object, and the classification performance can be efficiently improved via the combination of these classification results. Nevertheless, the reliability of classification results (soft outputs) usually vary across different source domains. Evidential reasoning is a suitable formalism to deal with such uncertain information; it serves as a basis for the new weighted evidence combination method proposed here to combine multiple classification results.

A. Weighted Combination of Multiple Classification Results

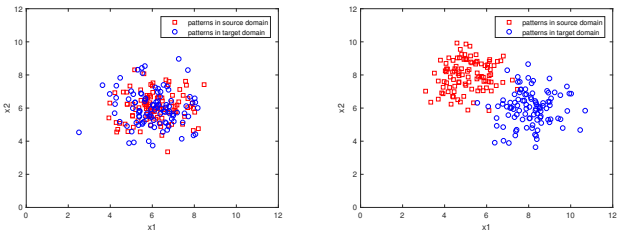
In the combination of multiple classification results, it is important to properly determine the weight of each result for achieving the best possible classification performance. In applications, the consistency between the source and target domains has significant influence on the classification result. If the distribution of the source domain is very consistent with that of the target domain, this source domain will be very useful for improving the classification performance on query objects in the target domain. However, the source domain cannot provide important information for classifying objects in the target domain when its distribution is quite distinct from that of the target domain. So we will attempt to estimate the reliability of each classification result according to the domain-consistency between the source and target domains.

There exist many metric methods to estimate the distribution difference. In some applications, the probability distribution function (PDF) of data in different domains is hard to obtain. The PDF-based metric methods like Kullback-Leibler (KL) divergence or Jensen-Shannon (JS) divergence cannot be directly applied in such case. The $\mathcal{A}$-distance proposed by Ben-David et al. in [28] has been widely used to measure distribution difference [43, 44], and it works well when the PDF is not available. So it is employed here to measure domain-consistency (distribution distance).

In the $\mathcal{A}$-distance method, the patterns from the source and target domains are annotated with pseudo labels (domain labels). The pseudo labels of patterns that have different real class labels in the source domain are all annotated by 0, whereas the labels become 1 when the objects come from the target domain. The two labels (i.e., 0, 1) do not represent the real class categories, and we call them domain labels for convenience. Such annotation considered as the benchmark is mainly used to reflect where the patterns come from (e.g., the source domain or the target domain) for calculating the domain-consistency. Then a classifier[2] can be learnt using these annotated patterns to distinguish whether the patterns are from the source or target domains. If the classification loss is big, it indicates that the patterns in the source and target domains are hard to distinguish, and the domain-consistency should be high. If the classification accuracy is high, it means the source and target domains are well separated, and the domain-consistency should be low. This is illustrated in Fig. 2. When the two distributions are close to each other, it is difficult to distinguish whether patterns are from the source or target domains. Nevertheless, the patterns are easily classified into the source and target domains if the distributions are quite separate. Formally, we denote $e(\mathcal{C})$ the average loss (i.e., error rate) of a linear classifier $\mathcal{C}$ discriminating the two domains $\mathcal{D}_S$ and $\mathcal{D}_T$. The $\mathcal{A}$-distance is defined by

$$d(\mathcal{D}_S, \mathcal{D}_T) = 2(1 - 2e(\mathcal{C})). \tag{6}$$

The theoretical derivation of this metric method is given in [28]. In applications, it achieves the biggest value (i.e., 2) when distributions of the source and target domains are completely different. If the distributions are the same, the average classification loss (error rate) for the source and target domain is 0.5, i.e., $e = 0.5$. Thus, the $\mathcal{A}$-distance value is equal to 0 according to Eq. (6) in such case. So distance value lies in the interval $[0, 2]$ in practice. This distance value varies with different linear classifiers, whereas the tendency of domain-consistency is similar. For instance,it is assumed that there exist two source domain $\mathcal{D}_{S_1}$, $\mathcal{D}_{S_2}$ and one target domain $\mathcal{D}_T$. Please note S_1 and S_2 represent the indexes of the two different source domains. When a Linear Discriminant Analysis (LDA) classifier is employed, one gets $d(\mathcal{D}_{S_1}, \mathcal{D}_T) > d(\mathcal{D}_{S_2}, \mathcal{D}_T)$, and a similar result can be also obtained using other linear classifiers, e.g., SVM with linear kernel. Now we will see how to exactly estimate the domain-consistency based on $\mathcal{A}$-distance.



(a) close distribution (b) quite different distribution

Fig. 2. The difference of patterns in the source and target domains with close distributions and quite different distributions.

The patterns in n source domains are denoted by $D_{S_1} = \{(\boldsymbol{x}_p^{S_1}, y_p^{S_1})\}_{p=1}^{N_1}, \cdots, D_{S_n} = \{(\boldsymbol{x}_p^{S_n}, y_p^{S_n})\}_{p=1}^{N_n}$, and the objects in the target domain are given by $D_T = \{\boldsymbol{x}_q^T\}_{q=1}^{N_T}$, where N_i and N_T are the number of patterns in the i-th source and target domains, and $\{y_p^{S_i}\}_{p=1}^{N_i}$ are corresponding real class labels of patterns in the i-th source domain. The patterns in the i-th source domain and objects in the target domain with domain labels are, respectively, denoted by $\tilde{D}_{S_i}^m = \{(\boldsymbol{x}_p^{S_i}, \tilde{y}_p^{S_i})\}_{p=1}^{N_i}$ and $\tilde{D}_T^m = \{(\boldsymbol{x}_q^T, \tilde{y}_q^T)\}_{q=1}^{N_T}$, where $\{\tilde{y}_p^{S_i}\}_{p=1}^{N_i} = 0$ and $\{\tilde{y}_q^T\}_{q=1}^{N_T} = 1$ are domain labels.

The new labeled data set can be obtained by merging these data sets as $\tilde{D}_{S_iT}^m = \tilde{D}_{S_i}^m \cup \tilde{D}_T^m$. A linear classifier $\mathcal{C}$ is learnt based on these labeled patterns to distinguish whether patterns are from the i-th source domain or the target domain. The average classification loss of the merged data set using

[2]This metric criterion [28] requires that the classifier is linear, e.g., Support Vector Machine (SVM) with liner kernel.

classifier $\mathcal{C}$ is

$$\tilde{e}_i\left(\mathcal{C}\right)=\frac{1}{N_i+N_T}\sum_{j=1}^{N_i+N_T}\left|\mathcal{C}\left(\boldsymbol{x}_j\right)-\tilde{y}_j\right|, i=1,\cdots,n \quad (7)$$

where $(\boldsymbol{x}_j,\tilde{y}_j)$ a pattern of the merged data set $\tilde{D}^m_{S_iT}$, and $\mathcal{C}(\boldsymbol{x}_j)\in\{0,1\}$. The distribution distance between the i-th source and target domains is

$$\tilde{d}_{S_iT}(\mathcal{D}_{S_i},\mathcal{D}_T)=2(1-2\tilde{e}_i(\mathcal{C})), i=1,\cdots,n. \quad (8)$$

The above distribution distance represents the consistency of the i-th source and target domains before distribution match. After matching via domain adaptation techniques, the distributions will become close, but these patterns still do not satisfy i.i.d., assumption. There may exist some differences between domains. The distribution distance after matching should also be taken into account in reliability evaluation. It is estimated in the similar way as above.

For distribution match of the source and target domains, there already exist many methods, e.g., TCA [5], JDA [7], TJM [9], BDA [10], to obtain a mapping matrix for transforming the patterns in the source and target domains into a common new feature space to make distributions drawn close. For example, in TCA [5], the dimensionality reduction technique can learn a transformed feature representation by minimizing the reconstruction error of the input data. Let us consider the input data matrix $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n]\in\mathbb{R}^{m\times n}$, where $\boldsymbol{X}$ is the set of patterns in the source and target domains, and $n=n_s+n_t$. The reconstruction error of transformed patterns in the source and target domains by mapping is defined by

$$\|\frac{1}{n_s}\sum_{i=1}^{n_s}\boldsymbol{A}^T\boldsymbol{x}_i-\frac{1}{n_t}\sum_{j=n_s+1}^{n_s+n_t}\boldsymbol{A}^T\boldsymbol{x}_j\|=tr(\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^T\boldsymbol{A}) \quad (9)$$

where

$$M_{ij}=\begin{cases}\frac{1}{n_sn_s},\boldsymbol{x}_i,\boldsymbol{x}_j\in\mathcal{D}_S\\ \frac{1}{n_tn_t},\boldsymbol{x}_i,\boldsymbol{x}_j\in\mathcal{D}_T\\ \frac{1}{n_sn_t},\text{otherwise.}\end{cases}$$

In Eq. (9), $\boldsymbol{A}$ denotes the mapping matrix, which is used to map the patterns in the source and target domains into a common new feature space. The value of $\boldsymbol{A}$ can be found in [5]. The distributions between domains become close under the new feature representation $\boldsymbol{A}^T\boldsymbol{X}$.

The complementary knowledge among the source domains is very helpful for improving the combination result to achieve the good classification performance. If all the source domains and the target domain are matched, some complementary information from individual source domain could be lost. In order to preserve the diversity (complementary knowledge) as much as possible, the distributions are matched between each source domain and the target domain. Original patterns in the i-th source and target domains are mapped into a low dimensionality feature space by the mapping matrix $\boldsymbol{A}_i$ to make the distributions close to each other. The new representation of these patterns are

$$\begin{cases}\hat{\boldsymbol{x}}_p^{S_i}=A_i\cdot\boldsymbol{x}_p^{S_i}, p=1,\cdots,N_i, i=1,\cdots,n\\ \hat{\boldsymbol{x}}_q^{T_i}=A_i\cdot\boldsymbol{x}_q^{T}, q=1,\cdots,N_T, i=1,\cdots,n.\end{cases} \quad (10)$$

These patterns in the i-th source and target domains after matching are denoted by $\hat{D}^m_{S_i}=\{(\hat{\boldsymbol{x}}_p^{S_i},\tilde{y}_p^{S_i})\}_{p=1}^{N_i}$ and $\hat{D}^m_{T_i}=\{(\hat{\boldsymbol{x}}_q^{T_i},\tilde{y}_q^{T})\}_{q=1}^{N_T}$. Similarly, these data sets are merged as $\hat{D}^m_{S_iT_i}=\hat{D}^m_{S_i}\cup\hat{D}^m_{T_i}$. The average classification loss is

$$\hat{e}_i(\mathcal{C})=\frac{1}{N_i+N_T}\sum_{k=1}^{N_i+N_T}\left|\mathcal{C}\left(\hat{\boldsymbol{x}}_k\right)-\tilde{y}_k\right|, i=1,\cdots,n \quad (11)$$

where $(\hat{\boldsymbol{x}}_k,\tilde{y}_k)$ is a pattern in the new data set $\hat{D}^m_{S_iT_i}$. The distribution distance between the i-th source and target domains after matching is calculated by

$$\hat{d}_{S_iT}(\mathcal{D}_{S_i},\mathcal{D}_T)=2(1-2\hat{e}_i(\mathcal{C})), i=1,\cdots,n. \quad (12)$$

The distribution distance $\tilde{d}_{S_iT}$ estimated by Eq. (8) before matching reflects the original difference between the i-th source and target domains. This distance will be large when distributions of the i-th source and target domains are quite different. When this distance is small, it implies that a lot of useful knowledge can be mined from the i-th source domain for classifying objects in the target domain. The distribution distance $\hat{d}_{S_iT}$ estimated by Eq. (12) generally becomes smaller than $\tilde{d}_{S_iT}$, since the domain adaptation algorithm aims to minimize the distribution distance between the i-th source and target domains. When $\hat{d}_{S_iT}$ is small, it means that the distributions are very close to each other in the new feature space. However, the mapping matrix transfers patterns by force, and the overfitting problem may happen. Consequently, the distance $\hat{d}_{S_iT}$ sometimes is very small even if $\tilde{d}_{S_iT}$ is quite large. In such case, the distance $\hat{d}_{S_iT}$ is not completely credible to reveal the classification ability of the i-th source domain. So both the distribution distances before and after matching should be taken into account. We use the geometric mean value to integrate them as

$$d_{S_iT}=\sqrt{\tilde{d}_{S_iT}\cdot\hat{d}_{S_iT}}. \quad (13)$$

Because both $\tilde{d}_{S_iT}$ and $\hat{d}_{S_iT}$ lie in $[0,2]$, the range of the geometric mean value d_{S_iT} will be also in $[0,2]$. If the i-th integrated distribution distance is large compared with others, it indicates that the distribution of the i-th source domain is quite different from that of the target domain, and the reliability of classification result obtained by the auxiliary of this source domain will not be high. Thus, a large distance will lead to a small weighting factor of the corresponding classification result. The classification result produced by the source domain with minimum distribution distance to the target domain will be considered with the biggest weighting factor (i.e., 1). The relative weighting factor for each classification result can be determined based on the integrated distribution distance by

$$\beta_i=\frac{\tilde{\beta}_i}{\max\left(\tilde{\beta}_1,\cdots,\tilde{\beta}_n\right)}, i=1,\cdots,n \quad (14)$$

where

$$\tilde{\beta}_i=e^{-d_{S_iT}}, i=1,\cdots,n.$$

In Eq. (14), the coefficient $\tilde{\beta}_i$ lies in the interval $[e^{-2},e^0]$, and the β value is in $(0,1]$ after normalization.

Once the weighting factors are obtained, the classification results represented by basic belief assignments (BBA) are discounted as follows:

$$\begin{cases} \tilde{m}_i(A) = \beta_i \cdot m_i(A), A \in 2^{\Omega}, A \neq \Omega \\ \tilde{m}_i(\Omega) = 1 - \beta_i + \beta_i \cdot m_i(\Omega) \end{cases}, i = 1, \cdots, n. \tag{15}$$

The discounted BBA's also satisfy the condition as Eq. (2), i.e., the sum of the discounted BBA's is equal to 1; the proof is given as bellow.

Proof:

$$\begin{aligned} \sum_{A \in 2^{\Omega}} \tilde{m}(A) &= \sum_{A \in 2^{\Omega}, A \neq \Omega} \beta_i m(A) + 1 - \beta_i + \beta_i m(\Omega) \\ &= 1 - \beta_i + \beta_i \sum_{A \in 2^{\Omega}} m(A) \\ &= 1 - \beta_i + \beta_i = 1, \end{aligned}$$

and $\tilde{m}(\emptyset) = 0$.

One can see that the mass assigned to each focal element is proportionally transferred to Ω by the given weighting factor β_i. Thus, the small weighting factor will cause the big belief of ignorance. If $\beta_i = 1$, it means that this BBA is completely reliable, and the BBA remains the same after discounting as $\tilde{m}(A) = m(A)$. If $\beta_i = 0$, it means that this BBA is not reliable at all, and the discounted BBA becomes $\tilde{m}(\Omega) = 1$ and $\tilde{m}(A) = 0, A \neq \Omega$. This total ignorance plays a neutral role in the combination as $\tilde{\boldsymbol{m}} \oplus \boldsymbol{m}_j = \boldsymbol{m}_j$ ($\oplus$ being the DS combination operator), and it has no influence on the fusion.

These n discounted classification results can be combined using DS fusion rule as Eq. (3) by

$$\boldsymbol{m} = \tilde{\boldsymbol{m}}_1 \oplus \cdots \oplus \tilde{\boldsymbol{m}}_n. \tag{16}$$

The object in the target domain will be classified according to this combination result.

B. Cautious Decision Making

The class decision of the object in the target domain is made depending on the combination of multiple classification results provided by different source domains. For the traditional hard decision making support, the object is generally assigned to the class with maximum mass of belief (probability). In contrast, when the combination result is not very reliable, the hard class decision will be with high risk of error.

Many methods have been developed to make decisions with uncertainty, e.g., fuzzy sets [45], belief functions [39], interval number [46] and so on. In applications, it is usually considered that the partial imprecision should be better than an error, since the imprecision can be clarified with other (costly) techniques, but errors may cause serious damage. There already exist some rules [47–49] to assign an object to the set of classes. Maximum Expected Utility (MEU) principle in terms of the utility of decision is widely used to make decision with uncertainty [50–52]. Nevertheless, the utility matrix is hard to obtain in some applications. So we want to develop an alternative cautious decision making method based on K nearest neighbors to improve pattern classification performance. It allows us to commit the patterns not only to the singleton class but also to the disjunction of several classes for the pattern hard to correctly classify[3].

The normal base classifiers usually work within the probability framework. So the classifier output is represented by a simple Bayesian BBA, and the combination result of classifiers will contain singleton classes and one extra total ignorant class brought in the discounting procedure. The pignistic probability $BetP(\cdot)$ [53] of the singleton classes transferred from the combination result is computed by

$$BetP(\omega_i) = m(\omega_i) + \frac{1}{c} m(\Omega), i = 1, \cdots, c. \tag{17}$$

The cautious decision is able to reduce the errors at the price of partial imprecision (i.e., some patterns difficult to distinguish are committed to the disjunction of several classes), but it is not a good solution when this cautious decision making generates a high imprecise rate. A unified *utility value* $\mathcal{U}$ reflecting the benefit of classification is defined to balance the error and imprecision.

Let us consider a pattern $\boldsymbol{x}_i$ with real label ω is classified to A based on cautious decision making, and A may contain a singleton class or several classes. If $\{\omega\} \cap A = \emptyset$, this class decision is an error, and the utility value is 0. If $\{\omega\} = A$, this is a correct decision, and the utility value is 1. If $\{\omega\} \in A$ and $|A| \geq 2$, it means that the real label is included in the cautious decision, which consists of several classes. The cardinality of set A denoted by $|A|$ reflects the imprecision degree of decision. The larger the $|A|$, the higher the imprecision. The utility value should be small when $|A|$ is large and vice versa. The utility value is defined as $(\frac{1}{|A|})^{\alpha}$, where α is a coefficient to control the influence of $|A|$ on utility value.

The utility value should be bigger than random selection of one singleton class from A but smaller than correct classification, i.e., $\frac{1}{|A|} < \mathcal{U} < 1$. So the coefficient α must lie in $(0, 1)$, and it can be tuned according to contextual applications. If the error cost is rather large in applications, it indicates that the utility value of imprecision is big, and then the α should be small. The utility value generally expressed by a common formula for the class decision of a pattern $\boldsymbol{x}_i$ is

$$\mathcal{U}(A|\boldsymbol{x}_i) = \left(\frac{|A \cap \{\omega\}|}{|A \cup \{\omega\}|} \right)^{\alpha}. \tag{18}$$

We want to pursue the maximum utility value for one pattern in cautious decision making. Now we will see how to make the cautious decision for each pattern. After combining multiple classification results obtained from different source domain, the combined classification result of one object $\boldsymbol{x}_j^T$ is denoted by $\boldsymbol{m}(\boldsymbol{\omega}|\boldsymbol{x}_j^T)$.

The data in n source domains are merged as one data set (i.e., $\bigcup_{i=1}^{n} D_{S_i}$), and we can match the distributions of this new data set with the data in the target domain by traditional ways for the classification task. The patterns of the merged data set in the new low feature space are regarded as training patterns to learn a classifier. The soft outputs of a classifier for these training patterns $\boldsymbol{x}_g^S, g = 1, \cdots, \sum_{i=1}^{n} N_i$ in the merged data set

[3]We mainly consider the disjunction of two classes here, because the pattern is usually hard to classify among a few (e.g., two) classes in real applications.

are represented by the probabilities $P(\boldsymbol{\omega}|\boldsymbol{x}_g^S)$. For each object $\boldsymbol{x}_j^T$ to classify over the frame of $\Omega = \{\omega_1, \omega_2, \cdots, \omega_c\}$, its combined classification result denoted by $\boldsymbol{m}(\boldsymbol{\omega}|\boldsymbol{x}_j^T)$ can be transformed to pignistic probability by Eq. (17) as $P\left(\boldsymbol{\omega}|\boldsymbol{x}_j^T\right)$. Then we find the K nearest neighbors of $P\left(\boldsymbol{\omega}|\boldsymbol{x}_j^T\right)$[4] from $P(\boldsymbol{\omega}|\boldsymbol{x}_g^S), g = 1, \cdots, \sum_{i=1}^{n} N_i$.

The K nearest neighbors of $P\left(\boldsymbol{\omega}|\boldsymbol{x}_j^T\right)$ with real labels are given by $P_j(\boldsymbol{\omega}|\boldsymbol{x}_k^S), k = 1, \cdots, K$, and we can easily obtain the utility value for the classification of these neighbors. Because $P_j(\boldsymbol{\omega}|\boldsymbol{x}_k^S), k = 1, \cdots, K$ are close to $P\left(\boldsymbol{\omega}|\boldsymbol{x}_j^T\right)$, and the classification result $P_j(\boldsymbol{\omega}|\boldsymbol{x}_k^S)$ can provide important prior knowledge for the classification of object $\boldsymbol{x}_j^T$.

We will consider two cases for making the class decision of the K neighbors. In the first case, the patterns are directly classified to the singleton class with maximum probability, whereas the patterns are committed to the disjunction of two classes with top two probabilities in the second case. If the sum of the utility value for the classification of the K neighbors in the first case is bigger than that of the second case, it indicates that the hard decision generally produces better performance than that of the imprecise decision, and we will directly classify the object to the singleton class with maximum probability. If the sum of the utility value in the first case is smaller than that of the second case, it means that hard classification of these patterns likely causes high error risk, and the imprecise decision is preferred. Then we will cautiously assign the object to the disjunction of two classes in order to reduce the errors. In the calculation of the sum of utility values w.r.t. the K neighbors, the neighbors are considered with different weights in order to reduce the influence of the choice of K number on decision making. The bigger the distance between the neighbor $P_j(\boldsymbol{\omega}|\boldsymbol{x}_k^S)$ and $P\left(\boldsymbol{\omega}|\boldsymbol{x}_j^T\right)$, the smaller weight for the utility value of the class decision on the k-th neighbor. The neighbor far from the classification result of the object is assigned a small weight, and it has little influence on the decision making. Thus, this method is robust to the choice of K, and it is convenient for the applications. The class decision for one object $\boldsymbol{x}_j^T$ is given by

$$f_C(\boldsymbol{x}_j^T) = \begin{cases} \{\omega_a\}, & \bar{\mathcal{U}}_j^{\mathrm{S}}(\{\omega_a\}) \geq \bar{\mathcal{U}}_j^{\mathrm{S}}(\{\omega_a, \omega_b\}) \\ \{\omega_a, \omega_b\}, & \bar{\mathcal{U}}_j^{\mathrm{S}}(\{\omega_a\}) < \bar{\mathcal{U}}_j^{\mathrm{S}}(\{\omega_a, \omega_b\}) \end{cases} \tag{19}$$

with

$$\begin{cases} \bar{\mathcal{U}}_j^{\mathrm{S}}(\{\omega_a\}) = \sum_{k=1}^{K} e^{-d_{jk}} \cdot \mathcal{U}(\{\omega_1^k\}|\boldsymbol{x}_k^S) \\ \bar{\mathcal{U}}_j^{\mathrm{S}}(\{\omega_a, \omega_b\}) = \sum_{k=1}^{K} e^{-d_{jk}} \cdot \mathcal{U}(\{\omega_1^k, \omega_2^k\}|\boldsymbol{x}_k^S) \end{cases} \tag{20}$$

and

$$d_{jk} = \left\| P(\boldsymbol{\omega}|\boldsymbol{x}_j^T) - P_j(\boldsymbol{\omega}|\boldsymbol{x}_k^S) \right\|, k = 1, \cdots, K \tag{21}$$

where ω_a and ω_b are the classes of object $\boldsymbol{x}_j^T$ with top two pignistic probabilities computed by $BetP(\cdot)$; ω_1^k and ω_2^k are the two most possible classes for k-th neighbor; $\mathcal{U}(\{\omega_1^k\}|\boldsymbol{x}_k^S)$ and $\mathcal{U}(\{\omega_1^k, \omega_2^k\}|\boldsymbol{x}_k^S)$ are utility value of the k-th nearest neighbor with decision ω_1^k and $\{\omega_1^k, \omega_2^k\}$; $\bar{\mathcal{U}}_j^S(\{\omega_a\})$ and $\bar{\mathcal{U}}_j^S(\{\omega_a, \omega_b\})$ are weighted sum of utility values of K nearest neighbors based on two different decision strategies; $e^{-d_{jk}}$ is a weight to reduce influence of the tuning of K number; $P(\boldsymbol{\omega}|\boldsymbol{x}_j^T)$ and $P_j(\boldsymbol{\omega}|\boldsymbol{x}_k^S)$ are probabilities belonging to different classes of object $\boldsymbol{x}_j^T$ and pattern $\boldsymbol{x}_k^S$; d_{jk} is Euclidean distance between $P\left(\boldsymbol{\omega}|\boldsymbol{x}_j^T\right)$ and its k-th nearest neighbor $P_j(\boldsymbol{\omega}|\boldsymbol{x}_k^S)$. Thus, the cautious decision result of every pattern is captured. We define the average utility value $\mathcal{U}_{\mathrm{A}}$ of all objects in the target domain to measure the classification performance with cautious decision making rule as

$$\mathcal{U}_{\mathrm{A}} = \frac{1}{N_T} \sum_{j=1}^{N_T} \mathcal{U}(f_C(\boldsymbol{x}_j^T)|\boldsymbol{x}_j^T). \tag{22}$$

Obviously, the average utility value will be reduced to the normal accuracy rate if each pattern is classified into the singleton class by hard decision.

The pseudo-code of the whole method is given in Algorithm 1 to clearly illustrate how to implement the proposed method.

C. Discussion on Parameter Tuning

Two important parameters, i.e., K number of nearest neighbors and utility coefficient α, are involved in the proposed method. The K neighbors in Eq. (19) are used for cautious decision making. The neighbor far from the object will be given a small weight, and the neighbor far away plays a small role in decision making. So the choice of K has small influence on the decision result, as will be shown experimentally in Section IV. Coefficient $\alpha \in (0, 1)$ in Eq. (18) is used to control the utility value of imprecision; the tuning of this parameter mainly depends on the actual applications. If the error cost is considered rather big, the imprecision is much preferable to errors. Then α should be small to make the utility value big for the partially imprecise decision. By doing this, we can efficiently reduce errors. Coefficient α should be big when the error cost is relatively small and the specific decision is necessary. The exact value of α can be determined according to the experiences of experts (end users).

IV. Experiments

A. Data Sets

We selected four widely used benchmark data sets Office+Caltech10, PIE, Office-31 and VLSC with multiple domains to validate the effectiveness of the proposed method. These data sets[5] have been considered as benchmarks to test domain adaptation techniques in [5, 7, 9, 10]. There are four different domains in Office+Caltech10, i.e., Amazon (A, images downloaded from Amazon), Caltech (C, images downloaded from google), DSLR (D, high-resolution images obtained by a digital SLR camera), Webcam (W, low-resolution images obtained by a web camera), and each domain has 10 real-world categories images. PIE consists of five domains,

[4]The pignistic probability transformation is employed here for convenience of finding the K close neighbors of the object from training data, because the classifier output for the training data in source domain is usually represented by a probability distribution.

[5]They can be downloaded from http://transferlearning.xyz.

Algorithm 1 :Combination of Transferable Classification

Input:

Data:

The labeled patterns in n source domains:
$D_{S_1} = \{(\boldsymbol{x}_p^{S_1}, y_p^{S_1})\}_{p=1}^{N_1}, \cdots, D_{S_n} = \{(\boldsymbol{x}_p^{S_n}, y_p^{S_n})\}_{p=1}^{N_n}$
and the unlabeled objects in the target domain:
$D_T = \{\boldsymbol{x}_q^T\}_{q=1}^{N_T}$.

Parameters:

α: Coefficient of utility value.

K: Number of nearest neighbors for decision making.

1: Compute the distribution distance before matching by Eqs. (7) and (8).
2: Match the distributions in traditional ways.
3: Calculate the distribution distance after matching by Eqs. (11) and (12).
4: Estimate the weighting factors by Eq. (14).
5: Obtain n pieces of evidence about the classification results of object with the auxiliary of n source domains.
6: Discount these classification results with the corresponding weighting factors by Eq. (15).
7: **for** $q = 1$ to N_t **do**
8: Combine the n discounted classification results by DS fusion rule as Eq. (16).
9: Transform the combination result in the form of BBA to pignistic probability by Eq. (17).
10: Find K nearest neighbors of combined results for cautious decision making.
11: Compute the sum of utility value of K nearest neighbors with hard and cautious decision by Eq. (20).
12: Make the final decision by comparing with the above sum of utility value as Eq. (19).
13: **end for**
14: Compute the average utility value by Eq. (22) and save the final class decisions.

Output:

Class decision results.

i.e., PIE_C05 (PIE1, left pose), PIE_C07 (PIE2, upward pose), PIE_C09 (PIE3, downward pose), PIE_C27 (PIE4, frontal pose), PIE_C29 (PIE5, right pose), and every domain has 68 individual face images. Office-31 contains of 3 different domains, i.e., Amazon (A), DSLR (D), Webcam (W), and 3,973 images with 31 classes. VLSC includes four domains, i.e., VOC2007 (V), LabelMe (L), SUN09 (S) and Caltech (C) and 10,729 pictures with five classes. The basic information of the data sets is shown in Table I.

TABLE I
BASIC INFORMATION OF THE BENCHMARK DATA SETS

Data set	Domain	Feature	Sample	Class
Office+Caltech10	Amazon (A)	800	958	10
	Caltech (C)	800	1123	10
	DSLR (D)	800	157	10
	Webcam (W)	800	295	10
PIE	PIE_C05 (PIE1)	1024	3332	68
	PIE_C07 (PIE2)	1024	1629	68
	PIE_C09 (PIE3)	1024	1632	68
	PIE_C27 (PIE4)	1024	3329	68
	PIE_C29 (PIE5)	1024	1632	68
Office-31	Amazon (A)	800	2715	31
	DSLR (D)	800	482	31
	Webcam (W)	800	776	31
VLSC	VOC2007 (V)	4096	3376	5
	LabelMe (L)	4096	2656	5
	SUN09 (S)	4096	3282	5
	Caltech101 (C)	4096	1415	5

B. Domain Adaptation Approaches and Classifier Fusion Methods

We have used some state-of-the-art basic domain adaptation methods to match the source and target domains. These method are briefly introduced here. Transfer Component Analysis (TCA) [5] adopts the marginal distribution discrepancy using Maximum Mean Discrepancy (MMD) to discover new representations. Joint Distribution Adaptation (JDA) [7] matches both the marginal and conditional distributions to learn a robust feature space. Transfer Joint Matching (TJM) [9] reuses some similar patterns and discovers new representation for transferring knowledge. Balanced Distribution Adaptation and Weighted Balanced Distribution Adaptation (BDA, WBDA) [10] consider the importance of marginal and conditional distributions, and imbalance issue respectively. The other two domain adaption methods as Geodesic Flow Kernel (GFK) [3] and CORrelation ALignment (CORAL) [13] are also included for comparison.

The k-Nearest Neighbor (k-NN) classifier is often used in these domain adaptation methods for classification task [7, 9], and it is also employed as base classifier in our experiments. In this work, we mainly focus on how to develop an efficient combination method with a given base classifier. We do not put emphasis on the selection of base classifier. The choice of the optimal k value in k-NN classifier is out of the scope of this work. We take $k = 5$ in k-NN base classifier. We found that the classification performance of 5NN is good in general. Moreover, this k value is not big, and thus the computation burden for seeking the k nearest neighbors is not very heavy. Our proposed method is compared with other methods using the same base classifier (i.e., k-NN, $k = 5$) for fair comparison in the experiments. The influence of parameter tuning of k ($k \in \{5, \ldots, 15\}$) on the classification performance of the proposed method will be tested in the sequel. In order to evaluate the performance of regular classification approaches, the regular classifiers k-NN and SVM with liner kernel built by the labeled data in the source domain are directly applied to classify the query patterns in the target domain.

Several often-used classifier fusion methods including Majority Vote (MV) method, Weighted Majority Vote (WMV) method, Average Fusion (AF) method, Weighted Average Fusion (WAF) method and DS combination rule have been used for comparison with the proposed Weighted DS (WDS) combination method. The weights of classification results in WMV and WAF are determined as in the proposed WDS method. In MV and WMV, the object is assigned to the class with the maximum voting score. In AF and WAF, the mean and weighted mean of multiple classification results are, respectively, calculated by $\boldsymbol{m} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{m}_i$ and $\boldsymbol{m} = \frac{1}{n}\sum_{i=1}^{n} w_i\boldsymbol{m}_i$, n being the number of pieces of evidence. The DS method is used to directly combine the classification results by Eq. (3). In the proposed WDS method, the multiple classification results are discounted with corresponding weights using Eq. (15) before the combination by Eq. (3).

C. Implementation Details

In this experiment, we selected one domain of the benchmark data set as the target domain, and the rest is considered as the source domains. The classification performance of the proposed method and other comparative methods are evaluated based on these classical domain adaptation methods as TCA, JDA, TJM, BDA and W-BDA for cross-domain image classification task, i.e., C $\rightarrow$ A, D $\rightarrow$ A, W $\rightarrow$ A, $\cdots$, A $\rightarrow$ W, C $\rightarrow$ W, D $\rightarrow$ W and PIE2$\rightarrow$PIE1, PIE3$\rightarrow$PIE1, PIE4$\rightarrow$PIE1, PIE5$\rightarrow$PIE1, $\cdots$, PIE1$\rightarrow$PIE5, PIE2$\rightarrow$PIE5, PIE3$\rightarrow$PIE5, PIE4$\rightarrow$PIE5, etc. The classification accuracy[6] $\mathcal{R}$ and average utility value $\mathcal{U}$ (w.r.t. cautious decision making rule) for different methods are shown in Tables II-XII. The meaning of acronyms (corresponding to different methods) in Tables II-XIII are explained in detail as follows.

Experiments without combining multi-source domains:

- SVM/k-NN: Traditional (regular) classification models (i.e., SVM, k-NN) built by labeled data in the source domain were directly used to classify query patterns in the target domain.
- GFK/CORAL/TCA/JDA/TJM/BDA/W-BDA: These classical domain adaptation approaches were employed to reduce the distribution difference between source and target domains, and then the base classifier k-NN was used to classify query patterns in the target domain.

Experiments for the combination of multi-source domains:

- MV/AF/DS: The query patterns in target domain were directly classified by the base classifier k-NN using the labeled patterns in source domain without domain adaption, and the combination methods MV/AF/DS were used to combine the multiple classification results.
- Combined Multiple Source Domain (CMSD): The multiple source domains were considered as one combined source domain, and the query patterns in target domain were directly classified by the base classifier k-NN using the labeled patterns in the combined source domain.
- TCA/JDA/TJM/BDA/WBDA+CMSD: The domain adaption techniques TCA/JDA/TJM/BDA/WBDA were applied to match distribution between the combined source domain and the target domain. Then the query patterns were classified by the base classifier k-NN.
- TCA/JDA/TJM/BDA/WBDA+MV/WMV/AF/WAF/DS /WDS: The domain adaption techniques as TCA/JDA/TJM/BDA/WBDA were operated between each source domain and target domain before the classification of query patterns by k-NN. Then the multiple classification results produced by different source domains were combined by the methods of MV, WMV, AF, WAF, DS and WDS.

Experiments for the combination of multi-source domains joint with cautious decision making rule:

- TCA/JDA/TJM/BDA/WBDA+AFC/DSC/WDSC: The query patterns were classified based on the combination of multi-source domains after the domain adaption by TCA/JDA/TJM/BDA/WBDA. Then the cautious class decision was made according to the combination results of AF/DS/WDS.

We have considered two cases for classifier fusion in experiments. In Case 1, the classifiers were directly learnt using the labeled patterns in each source domain without matching distribution for the classification of query patterns in the target domain. The classification accuracies denoted by $\mathcal{R}_{\rm MV}$, $\mathcal{R}_{\rm AF}$, $\mathcal{R}_{\rm DS}$, $\mathcal{R}_{\rm CMSD}$ was reported in Tables II-XII. In Case 2, the domain adaption techniques were implemented for matching distribution at first, and then the classifier was learnt in the new feature space to classify the query patterns. The classification results yielded by the classifier are combined finally. The accuracies are denoted by $\mathcal{R}_{\mathcal{A}+\mathcal{B}}$, where $\mathcal{A}$ and $\mathcal{B}$, respectively, stand for domain adaptation techniques as TCA/JDA/TJM/BDA/WBDA and combination methods as MV/WMV/AF/WAF/DS/WDS. These mentioned classifier fusion methods (e.g., MV, AF and DS) were applied to combine the classifiers in both cases for comparison.

The hyper parameters in domain adaptation techniques were all determined in the same manner as previous references [5, 7, 9, 10], i.e., the iteration number T = 10, balance parameter $\mu = 0.4$ and regularization parameter $\lambda = 1$, and the linear kernel employed as well.

In the decision phase, the new cautious decision making rule can be applied joint with the proposed weighted combination method (WDS) as well as other combination methods (e.g., AF, DS). The K number of nearest neighbors and utility coefficient α are tuning parameters involved in cautious decision making rule. The value of α should be determined according to the error cost in actual application. The bigger the error cost, the smaller the value of α. It is considered $\alpha = 0.6$ as default value here. For fair comparison, we reported the average utility values with variance (mean$\pm$variance) of the cautious decision making result by WDS method and other methods (e.g., AF, DS) with K ranging from 5 to 15 in Tables II-XII denoted by $\mathcal{U}_{\rm AFC}$, $\mathcal{U}_{\rm DSC}$ and $\mathcal{U}_{\rm WDSC}$. The maximum accuracy[7] and average utility value are marked in bold for convenience. We also tested the influence of parameter tuning on the performance of cautious decision making rule. The average utility value curves of the proposed method with the different K number of nearest neighbors as $K \in \{5, \ldots, 15\}$ and the different utility coefficients $\alpha \in [0.1, 1]$ are shown in Figs. 4 and 5.

The k-NN was employed as the base classifier in the experiments. In order to test the influence of tuning of k on the classification performance of our proposed weighted combination method (WDS) and other comparative methods, we have shown the classification results of these methods with different k values ranging from 5 to 15 in Fig. 3, and the mean classification accuracy with variance is reported in Table XIII.

[6]The accuracy can be computed by $\frac{N}{T}$,where N and T is the number of correctly classified patterns and total patterns.

[7]The accuracy is calculated based on the traditional hard decision making criteria that the pattern is directly assigned to the class with maximum probability. There is no tuning parameter involved in such hard decision making way. So we just report the classification accuracy without variance for different methods with the given base classifier (i.e., k-NN).

TABLE II
CLASSIFICATION PERFORMANCE OF DIFFERENT FUSION METHODS BASED ON JDA IN OFFICE+CALTECH10 DATA SETS

Task	$\mathcal{R}_{SVM}$	$\mathcal{R}_{k-NN}$	$\mathcal{R}_{CMSD}$	$\mathcal{R}_{MV}$	$\mathcal{R}_{AF}$	$\mathcal{R}_{DS}$	$\mathcal{R}_{GFK}$	$\mathcal{R}_{CORAL}$	$\mathcal{R}_{JDA}$	$\mathcal{R}_{JDA+CMSD}$	$\mathcal{R}_{JDA+MV}$	$\mathcal{R}_{JDA+WMV}$	$\mathcal{R}_{JDA+AF}$	$\mathcal{R}_{JDA+WAF}$	$\mathcal{U}_{JDA+AFC}$	$\mathcal{R}_{JDA+DS}$	$\mathcal{U}_{JDA+DSC}$	$\mathcal{R}_{JDA+WDS}$	$\mathcal{U}_{JDA+WDSC}$
C→A	30.53	22.76					41.02	20.15	45.93										
D→A	29.85	26.62	24.32	26.10	29.12	28.91	32.05	30.69	32.67	41.78	40.08	45.30	44.15	46.66	*45.28±0.08*	44.15	*46.21±0.03*	**46.76**	***49.48±0.05***
W→A	29.54	22.65					31.84	26.20	39.77										
A→C	44.08	24.04					40.25	23.06	41.05										
D→C	28.50	26.09	24.04	26.63	26.09	22.80	30.10	32.06	29.12	44.61	37.76	40.69	42.56	44.52	*43.21±0.02*	43.37	*44.01±0.02*	**45.75**	***46.60±0.04***
W→C	26.51	18.08					30.72	25.73	31.97										
A→D	40.13	23.57					36.61	30.75	42.04										
C→D	45.22	24.84	37.58	26.11	36.94	33.76	41.40	26.75	49.68	73.85	58.60	73.25	77.71	80.25	*77.73±0.17*	74.52	*77.12±0.04*	**80.25**	***82.57±0.05***
W→D	62.82	44.59					77.90	73.25	79.62										
A→W	39.66	27.12					40.00	26.10	38.64										
C→W	42.37	26.10	29.49	32.20	39.32	35.25	40.68	19.66	46.10	62.71	52.24	68.47	61.69	66.10	*62.38±0.83*	62.03	*63.91±0.10*	**70.51**	***72.57±0.12***
D→W	65.42	43.73					64.41	63.56	68.81										
Average	40.39	27.52	28.85	27.76	32.87	30.18	40.25	33.16	45.45	55.74	47.17	56.92	56.53	59.38	*57.15*	56.01	*57.60*	**60.84**	***62.59***

TABLE III
CLASSIFICATION PERFORMANCE OF DIFFERENT FUSION METHODS BASED ON TCA IN OFFICE+CALTECH10 DATA SETS

	$\mathcal{R}_{SVM}$	$\mathcal{R}_{k-NN}$	$\mathcal{R}_{CMSD}$	$\mathcal{R}_{MV}$	$\mathcal{R}_{AF}$	$\mathcal{R}_{DS}$	$\mathcal{R}_{GFK}$	$\mathcal{R}_{CORAL}$	$\mathcal{R}_{TCA}$	$\mathcal{R}_{TCA+CMSD}$	$\mathcal{R}_{TCA+MV}$	$\mathcal{R}_{TCA+WMV}$	$\mathcal{R}_{TCA+AF}$	$\mathcal{R}_{TCA+WAF}$	$\mathcal{U}_{TCA+AFC}$	$\mathcal{R}_{TCA+DS}$	$\mathcal{U}_{TCA+DSC}$	$\mathcal{R}_{TCA+WDS}$	$\mathcal{U}_{TCA+WDSC}$
C→A	30.53	22.76					41.02	20.15	**47.91**										
D→A	29.58	26.62	24.32	26.10	29.12	28.91	32.05	30.69	20.88	46.35	33.51	41.23	36.95	40.92	*40.75±0.27*	38.00	*39.15±0.03*	45.20	***48.72±0.12***
W→A	29.54	22.65					31.84	26.20	33.30										
A→C	44.08	24.04					40.25	23.06	41.41										
D→C	28.50	26.09	24.04	26.63	26.09	22.80	30.10	32.06	27.16	43.19	37.13	40.61	41.23	42.83	*43.60±0.72*	42.65	*43.33±0.01*	**43.46**	***45.89±0.03***
W→C	26.51	18.08					30.72	25.73	31.79										
A→D	40.13	23.57					36.61	30.75	36.94										
C→D	45.22	24.84	37.58	26.11	36.94	33.76	41.40	26.75	49.68	70.06	59.87	73.89	77.07	80.89	*77.18±1.84*	73.89	*75.44±0.14*	79.62	***82.96±0.31***
W→D	62.82	44.59					77.90	73.25	**81.53**										
A→W	39.66	27.12					40.00	26.10	40.00										
C→W	42.37	26.10	29.49	32.20	39.32	35.25	40.68	19.66	45.08	57.29	52.88	64.41	63.39	65.76	*64.29±0.97*	61.69	*64.40±0.06*	**72.20**	***74.26±0.18***
D→W	65.42	43.73					64.41	63.56	67.46										
Average	40.39	27.52	28.85	27.76	32.87	30.18	40.25	33.16	43.57	54.22	45.84	55.04	54.66	57.60	*56.55*	54.06	*55.42*	**60.21**	***62.75***

TABLE IV
CLASSIFICATION PERFORMANCE OF DIFFERENT FUSION METHODS BASED ON TJM IN OFFICE+CALTECH10 DATA SETS

Task	$\mathcal{R}_{SVM}$	$\mathcal{R}_{k-NN}$	$\mathcal{R}_{CMSD}$	$\mathcal{R}_{MV}$	$\mathcal{R}_{AF}$	$\mathcal{R}_{DS}$	$\mathcal{R}_{GFK}$	$\mathcal{R}_{CORAL}$	$\mathcal{R}_{TJM}$	$\mathcal{R}_{TJM+CMSD}$	$\mathcal{R}_{TJM+MV}$	$\mathcal{R}_{TJM+WMV}$	$\mathcal{R}_{TJM+AF}$	$\mathcal{R}_{TJM+WAF}$	$\mathcal{U}_{TJM+AFC}$	$\mathcal{R}_{TJM+DS}$	$\mathcal{U}_{TJM+DSC}$	$\mathcal{R}_{TJM+WDS}$	$\mathcal{U}_{TJM+WDSC}$
C→A	30.53	22.76					41.02	20.15	46.55										
D→A	29.58	26.62	24.32	26.10	29.12	28.91	32.05	30.69	29.33	47.29	37.06	42.69	42.69	45.62	*44.90*±0.07	42.07	*43.51±0.05*	**48.12**	***51.09±0.02***
W→A	29.54	22.65					31.84	26.20	35.49										
A→C	44.08	24.04					40.25	23.06	42.21										
D→C	28.50	26.09	24.04	26.63	26.09	22.80	30.10	32.06	30.10	44.08	37.85	41.50	43.01	44.97	*43.50±0.52*	44.43	*44.66±0.01*	**45.59**	***47.02±0.01***
W→C	26.51	18.08					30.72	25.73	31.88										
A→D	40.13	23.57					36.61	30.75	40.76										
C→D	24.84	45.22	37.58	26.11	36.94	33.76	41.40	26.75	48.41	72.61	64.97	75.16	76.43	77.71	*77.52±0.11*	75.16	*75.83±0.12*	**81.53**	***83.92±0.42***
W→D	44.59	62.82					77.90	73.25	80.25										
A→W	39.66	27.12					40.00	26.10	39.66										
C→W	42.37	26.10	29.49	32.20	39.32	35.25	40.68	19.66	49.49	63.39	54.58	65.42	63.72	68.14	*66.15±0.58*	61.69	*65.34±0.15*	**71.53**	***74.19±0.02***
D→W	65.42	43.73					64.41	63.56	67.80										
Average	40.39	27.52	28.85	27.76	32.87	30.18	40.25	33.16	45.16	58.84	48.62	56.19	56.46	59.11	*58.16*	55.84	*57.02*	**61.62**	***64.13***

TABLE V
CLASSIFICATION PERFORMANCE OF DIFFERENT FUSION METHODS BASED ON BDA IN OFFICE+CALTECH10 DATA SETS

	$\mathcal{R}_{SVM}$	$\mathcal{R}_{k-NN}$	$\mathcal{R}_{CMSD}$	$\mathcal{R}_{MV}$	$\mathcal{R}_{AF}$	$\mathcal{R}_{DS}$	$\mathcal{R}_{GFK}$	$\mathcal{R}_{CORAL}$	$\mathcal{R}_{BDA}$	$\mathcal{R}_{BDA+CMSD}$	$\mathcal{R}_{BDA+MV}$	$\mathcal{R}_{BDA+WMV}$	$\mathcal{R}_{BDA+AF}$	$\mathcal{R}_{BDA+WAF}$	$\mathcal{U}_{BDA+AFC}$	$\mathcal{R}_{BDA+DS}$	$\mathcal{U}_{BDA+DSC}$	$\mathcal{R}_{BDA+WDS}$	$\mathcal{U}_{BDA+WDSC}$
C→A	30.53	22.76					41.02	20.15	47.60										
D→A	29.85	26.62	24.32	26.10	29.12	28.91	32.05	30.69	33.72	48.02	40.40	46.24	43.74	44.26	*45.81±0.17*	42.80	*43.96±0.07*	**48.64**	***50.19±0.03***
W→A	29.54	22.64					31.84	26.20	38.83										
A→C	44.08	24.04					40.25	23.06	41.23										
D→C	28.50	26.09	24.04	26.63	26.09	22.80	30.10	32.06	32.68	40.46	37.85	41.14	38.56	41.05	*39.68±0.62*	39.63	*39.73±0.01*	**42.92**	***44.81±0.03***
W→C	26.51	18.08					30.72	25.73	30.81										
A→D	40.13	23.57					36.61	30.75	40.76										
C→D	45.22	24.84	37.58	26.11	36.94	33.76	41.40	26.75	54.14	71.34	61.15	75.80	73.89	80.25	*75.42±1.22*	75.16	*75.75±0.07*	82.17	***83.50±0.20***
W→D	62.82	44.59					77.90	73.25	**83.44**										
A→W	39.66	27.12					40.00	26.10	40.00										
C→W	42.37	26.10	29.49	32.20	39.32	35.25	40.68	19.66	49.83	63.73	56.27	71.53	66.10	73.90	*66.86±0.24*	63.39	*64.94±0.10*	**75.25**	***75.69±0.21***
D→W	65.42	43.73					64.41	63.56	74.58										
Average	40.39	27.52	28.85	27.76	32.87	30.18	40.25	33.16	47.30	55.89	48.92	58.68	55.57	59.87	*56.94*	55.25	*55.92*	**62.25**	***63.57***

TABLE VI
CLASSIFICATION PERFORMANCE OF DIFFERENT FUSION METHODS BASED ON JDA IN PIE DATA SETS

Task	$\mathcal{R}_{SVM}$	$\mathcal{R}_{k-NN}$	$\mathcal{R}_{CMSD}$	$\mathcal{R}_{MV}$	$\mathcal{R}_{AF}$	$\mathcal{R}_{DS}$	$\mathcal{R}_{GFK}$	$\mathcal{R}_{CORAL}$	$\mathcal{R}_{JDA}$	$\mathcal{R}_{JDA+CMSD}$	$\mathcal{R}_{JDA+MV}$	$\mathcal{R}_{JDA+WMV}$	$\mathcal{R}_{JDA+AF}$	$\mathcal{R}_{JDA+WAF}$	$\mathcal{U}_{JDA+AFC}$	$\mathcal{R}_{JDA+DS}$	$\mathcal{U}_{JDA+DSC}$	$\mathcal{R}_{JDA+WDS}$	$\mathcal{U}_{JDA+WDSC}$
PIE2→PIE1	30.51	41.63					47.35	42.96	34.90										
PIE3→PIE1	21.53	43.27	61.94	50.71	56.94	55.92	33.57	47.24	51.53	65.92	61.43	65.85	70.51	70.61	*71.83±0.02*	68.47	*68.64±0.02*	**70.92**	***71.23±0.02***
PIE4→PIE1	41.33	61.63					47.14	62.04	69.08										
PIE5→PIE1	34.08	42.14					30.31	41.12	37.55										
PIE1→PIE2	29.79	24.17					38.54	34.17	44.37										
PIE3→PIE2	27.29	46.04	72.92	59.17	67.08	66.46	52.08	56.67	45.83	70.42	60.00	69.79	72.92	75.00	*72.77±0.08*	68.75	*69.05±0.00*	**76.25**	***76.26±0.02***
PIE4→PIE2	55.21	62.08					73.75	73.54	71.88										
PIE5→PIE2	25.62	33.96					30.42	35.00	34.79										
PIE1→PIE3	37.71	27.92					41.46	37.71	45.83										
PIE2→PIE3	42.08	54.58	76.25	72.29	75.21	73.33	64.58	60.83	47.71	80.00	69.58	77.29	77.71	78.96	*78.35±0.16*	77.71	*77.81±0.02*	**81.46**	***81.60±0.01***
PIE4→PIE3	59.79	73.96					70.72	75.42	78.54										
PIE5→PIE3	29.17	41.25					34.58	45.21	33.75										
PIE1→PIE4	40.71	20.71					42.04	20.10	63.57										
PIE2→PIE4	53.27	59.93	67.86	60.31	66.02	67.65	64.39	61.53	62.45	77.04	76.43	78.08	73.27	74.49	***78.94±0.02***	77.96	*78.18±0.01*	**78.78**	*78.73±0.01*
PIE3→PIE4	41.43	56.73					46.02	61.73	64.69										
PIE5→PIE4	35.61	47.55					36.53	43.27	44.18										
PIE1→PIE5	22.71	12.29					30.21	20.42	36.88										
PIE2→PIE5	23.33	29.38	56.04	43.54	46.67	40.00	35.83	34.17	30.83	55.42	53.75	51.88	60.42	**61.04**	***60.52±0.05***	59.79	*60.51±0.01*	59.79	*60.24±0.01*
PIE3→PIE5	29.58	40.00					32.29	42.08	39.58										
PIE4→PIE5	38.75	50.42					41.88	53.96	54.79										
Average	35.98	36.24	67.00	57.20	62.38	60.67	44.68	47.49	49.64	69.76	64.24	68.58	72.97	72.02	*73.28*	70.54	*70.84*	**73.44**	***73.61***

TABLE VII
CLASSIFICATION PERFORMANCE OF DIFFERENT FUSION METHODS BASED ON TCA IN PIE DATA SETS

Task	$\mathcal{R}_{SVM}$	$\mathcal{R}_{k-NN}$	$\mathcal{R}_{CMSD}$	$\mathcal{R}_{MV}$	$\mathcal{R}_{AF}$	$\mathcal{R}_{DS}$	$\mathcal{R}_{GFK}$	$\mathcal{R}_{CORAL}$	$\mathcal{R}_{TCA}$	$\mathcal{R}_{TCA+CMSD}$	$\mathcal{R}_{TCA+MV}$	$\mathcal{R}_{TCA+WMV}$	$\mathcal{R}_{TCA+AF}$	$\mathcal{R}_{TCA+WAF}$	$\mathcal{U}_{TCA+AFC}$	$\mathcal{R}_{TCA+DS}$	$\mathcal{U}_{TCA+DSC}$	$\mathcal{R}_{TCA+WDS}$	$\mathcal{U}_{TCA+WDSC}$
PIE2→PIE1	30.51	41.63					47.35	42.96	39.69										
PIE3→PIE1	21.53	43.27	61.94	50.71	56.94	55.92	33.57	47.24	49.90	66.43	64.29	69.59	71.43	71.33	*72.21±0.01*	70.00	*70.31±0.01*	**72.65**	***72.76±0.01***
PIE4→PIE1	41.33	61.63					47.14	62.04	68.06										
PIE5→PIE1	34.08	42.14					30.31	41.12	37.65										
PIE1→PIE2	29.79	24.17					38.54	34.17	44.79										
PIE3→PIE2	27.29	46.04	72.92	59.17	67.08	66.46	52.08	56.67	47.71	73.54	67.08	77.50	79.17	80.83	*77.38±0.11*	76.04	*76.03±0.00*	**81.67**	***81.69±0.04***
PIE4→PIE2	55.21	62.08					73.75	73.54	76.67										
PIE5→PIE2	25.62	33.96					30.42	35.00	42.08										
PIE1→PIE3	37.71	27.92					41.46	37.71	49.58										
PIE2→PIE3	42.08	54.58	76.25	72.29	75.21	73.33	64.58	60.83	52.50	81.67	72.50	79.58	80.42	81.87	*79.97±0.04*	77.71	*78.03±0.02*	**83.33**	***83.64±0.03***
PIE4→PIE3	59.79	73.96					70.72	75.42	80.00										
PIE5→PIE3	29.71	41.25					34.58	45.21	37.71										
PIE1→PIE4	40.71	20.71					42.04	20.10	57.35										
PIE2→PIE4	53.27	59.93	67.86	60.31	66.02	67.65	64.39	61.53	64.08	77.86	74.69	75.41	81.43	**81.73**	***81.36±0.01***	79.18	*79.16±0.00*	74.80	*75.12±0.01*
PIE3→PIE4	41.43	56.73					46.02	61.73	65.92										
PIE5→PIE4	35.61	47.55					36.53	43.27	45.00										
PIE1→PIE5	22.71	12.29					30.21	20.42	33.54										
PIE2→PIE5	23.33	29.38	56.04	43.54	46.67	40.00	35.83	34.17	35.21	54.58	51.67	59.58	59.58	59.58	*59.68±0.04*	59.38	*59.27±0.00*	**60.83**	***60.83±0.01***
PIE3→PIE5	29.58	40.00					32.29	42.08	43.33										
PIE4→PIE5	38.75	50.42					41.88	53.96	58.13										
Average	35.98	36.24	67.00	57.20	62.38	60.67	44.68	47.49	51.45	70.82	66.05	72.41	74.41	75.07	*74.12*	72.46	*72.56*	**74.66**	***74.81***

D. Performance Analysis

In Tables II-XIII, we can see that the performance of regular classification methods (i.e., SVM, k-NN) is poor with respect to the domain adaptation algorithms and the proposed combination method. Meanwhile, we also find that the classification accuracy of CMSD and other classifier fusion methods in Case 1 (i.e., MV, AF, DS) is not so high as those of other combination methods joint with domain adaption techniques. This is because the labeled training patterns in the source domain and test patterns in the target domain are drawn from quite different distributions, and the classifiers learnt using the labeled patterns in source domain are not very effective for dealing with query patterns in target domain. It indicates that the distribution difference affects the classification performance a lot, and the implementation of domain adaption technique is very important to reduce the distribution difference for achieving high classification accuracy.

The classification accuracy of the majority voting methods as TCA/JDA/TJM/BDA/WBDA+MV usually lies between the maximum and minimum of accuracy of the multiple individuals, while the performance of the average fusion methods as TCA/JDA/TJM/BDA/WBDA+AF are close to that of majority voting method. In the DS combination method with TCA/JDA/TJM/BDA/WBDA, the multiple classification results derived from different source domains are considered as equal weight in the combination, and this cannot well reflect the different reliabilities of these results. The proposed Weighted DS combination method joint with domain adaption technique as TCA/JDA/TJM/BDA/WBDA+WDS usually produces significantly higher accuracy than other methods in most cases. This is because the proposed WDS method can fully take advantage of the complementary information provides by multiple source domains, and it can also effectively control the influence of different classification results using the weights calculated depending on the distribution distance between each source and target domain. If data distribution of one source

TABLE VIII
CLASSIFICATION PERFORMANCE OF DIFFERENT FUSION METHODS BASED ON TJM IN PIE DATA SETS

Task	$\mathcal{R}_{SVM}$	$\mathcal{R}_{k-NN}$	$\mathcal{R}_{CMSD}$	$\mathcal{R}_{MV}$	$\mathcal{R}_{AF}$	$\mathcal{R}_{DS}$	$\mathcal{R}_{GFK}$	$\mathcal{R}_{CORAL}$	$\mathcal{R}_{TJM}$	$\mathcal{R}_{TJM+CMSD}$	$\mathcal{R}_{TJM+MV}$	$\mathcal{R}_{TJM+WMV}$	$\mathcal{R}_{TJM+AF}$	$\mathcal{R}_{TJM+WAF}$	$\mathcal{U}_{TJM+AFC}$	$\mathcal{R}_{TJM+DS}$	$\mathcal{U}_{TJM+DSC}$	$\mathcal{R}_{TJM+WDS}$	$\mathcal{U}_{TJM+WDSC}$
PIE2→PIE1	30.51	41.63					47.35	42.96	49.90										
PIE3→PIE1	21.53	43.27	61.94	50.71	56.94	55.92	33.57	47.24	49.69	73.75	65.31	72.86	76.33	76.43	*75.82±0.09*	73.98	*74.21±0.01*	**76.53**	***76.87±0.010***
PIE4→PIE1	41.33	61.63					47.14	62.04	74.80										
PIE5→PIE1	34.08	42.14					30.31	41.12	39.49										
PIE1→PIE2	29.79	24.17					38.54	34.17	59.79										
PIE3→PIE2	27.29	46.04	72.92	59.17	67.08	66.46	52.08	56.67	52.08	85.62	76.67	85.42	88.96	88.96	*86.86±0.09*	83.96	*84.08±0.01*	**90.00**	***90.11±0.01***
PIE4→PIE2	55.21	62.08					73.75	73.54	87.50										
PIE5→PIE2	25.62	33.96					30.42	35.00	31.25										
PIE1→PIE3	37.71	27.92					41.46	37.71	59.17										
PIE2→PIE3	42.08	54.58	76.25	72.29	75.21	73.33	64.58	60.83	67.29	85.83	80.63	84.17	84.58	85.00	*84.20±0.05*	83.33	*83.47±0.01*	**86.88**	***86.90±0.01***
PIE4→PIE3	59.79	73.96					70.72	75.42	86.04										
PIE5→PIE3	29.17	41.25					34.58	45.21	46.67										
PIE1→PIE4	40.71	20.71					42.04	20.10	74.39										
PIE2→PIE4	53.27	59.93	67.86	60.31	66.02	67.65	64.39	61.53	78.16	82.74	83.37	85.10	84.53	84.80	*85.29±0.12*	85.82	*85.89±0.01*	**85.92**	***86.81±0.01***
PIE3→PIE4	41.43	56.73					46.02	61.73	66.22										
PIE5→PIE4	35.61	47.55					36.53	43.27	51.73										
PIE1→PIE5	22.71	12.29					30.21	20.42	50.42										
PIE2→PIE5	23.33	29.38	56.04	43.54	46.67	40.00	35.83	34.17	42.17	60.42	66.25	65.21	69.37	**69.58**	***70.88±0.05***	70.21	*70.47±0.01*	64.79	*65.41±0.00*
PIE3→PIE5	29.58	40.00					32.29	42.08	52.71										
PIE4→PIE5	38.75	50.42					41.88	53.96	64.17										
Average	35.98	36.24	67.00	57.20	62.38	60.67	44.68	47.49	59.21	77.67	74.45	78.55	81.32	80.95	*81.01*	79.46	*79.62*	**80.82**	***81.22***

TABLE IX
CLASSIFICATION PERFORMANCE OF DIFFERENT FUSION METHODS BASED ON BDA IN PIE DATA SETS

Task	$\mathcal{R}_{SVM}$	$\mathcal{R}_{k-NN}$	$\mathcal{R}_{CMSD}$	$\mathcal{R}_{MV}$	$\mathcal{R}_{AF}$	$\mathcal{R}_{DS}$	$\mathcal{R}_{GFK}$	$\mathcal{R}_{CORAL}$	$\mathcal{R}_{BDA}$	$\mathcal{R}_{BDA+CMSD}$	$\mathcal{R}_{BDA+MV}$	$\mathcal{R}_{BDA+WMV}$	$\mathcal{R}_{BDA+AF}$	$\mathcal{R}_{BDA+WAF}$	$\mathcal{U}_{BDA+AFC}$	$\mathcal{R}_{BDA+DS}$	$\mathcal{U}_{BDA+DSC}$	$\mathcal{R}_{BDA+WDS}$	$\mathcal{U}_{BDA+WDSC}$
PIE2→PIE1	30.51	41.63					47.35	42.96	40.10										
PIE3→PIE1	21.53	43.27	61.94	50.71	56.94	55.92	33.57	47.24	48.88	70.51	64.29	63.57	70.71	70.51	*70.64±0.04*	70.20	*70.19±0.01*	**71.33**	***71.39±0.01***
PIE4→PIE1	41.33	61.63					47.14	62.04	69.49										
PIE5→PIE1	34.08	42.14					30.31	41.12	39.59										
PIE1→PIE2	29.79	24.17					38.54	34.17	44.37										
PIE3→PIE2	27.29	46.04	72.92	59.17	67.08	66.46	52.08	56.67	43.96	73.58	57.50	70.00	70.21	71.25	*70.30±0.06*	67.71	*68.33±0.01*	**74.17**	***74.69±0.03***
PIE4→PIE2	55.21	62.08					73.75	73.54	73.33										
PIE5→PIE2	25.62	33.96					30.42	35.00	26.04										
PIE1→PIE3	37.71	27.92					41.46	37.71	47.08										
PIE2→PIE3	42.08	54.58	76.25	72.29	75.21	73.33	64.58	60.83	49.79	80.63	69.37	78.33	79.17	80.21	*78.20±0.11*	76.25	*76.19±0.01*	**82.29**	***82.53±0.04***
PIE4→PIE3	59.79	73.96					70.52	75.42	78.75										
PIE5→PIE3	29.17	33.96					34.58	45.21	37.92										
PIE1→PIE4	40.71	20.71					42.04	20.10	60.51										
PIE2→PIE4	53.27	59.93	67.86	60.31	66.02	67.65	64.39	61.53	65.31	77.35	73.67	77.35	82.35	**82.86**	***82.37±0.05***	79.08	*79.22±0.01*	79.29	*79.96±0.01*
PIE3→PIE4	41.43	56.73					46.02	61.73	62.45										
PIE5→PIE4	35.61	47.55					36.53	43.27	41.02										
PIE1→PIE5	22.71	12.29					30.21	20.42	36.46										
PIE2→PIE5	23.33	29.38	56.04	43.54	46.67	40.00	35.83	34.17	25.62	57.08	52.71	56.04	61.67	62.50	*59.63±0.23*	59.79	*59.90±0.01*	**61.88**	***62.38±0.02***
PIE3→PIE5	29.58	40.00					32.29	42.08	38.96										
PIE4→PIE5	38.75	50.42					41.88	53.96	54.58										
Average	38.98	36.24	67.00	57.20	62.38	60.67	44.68	47.49	48.96	71.83	63.51	69.06	72.82	73.47	*72.23*	70.61	*70.77*	**73.79**	***74.19***

TABLE X
CLASSIFICATION PERFORMANCE OF DIFFERENT FUSION METHODS BASED ON WBDA IN OFFICE-CALTECH10 DATA SETS

Task	$\mathcal{R}_{SVM}$	$\mathcal{R}_{k-NN}$	$\mathcal{R}_{CMSD}$	$\mathcal{R}_{MV}$	$\mathcal{R}_{AF}$	$\mathcal{R}_{DS}$	$\mathcal{R}_{GFK}$	$\mathcal{R}_{CORAL}$	$\mathcal{R}_{WBDA}$	$\mathcal{R}_{WBDA+CMSD}$	$\mathcal{R}_{WBDA+MV}$	$\mathcal{R}_{WBDA+WMV}$	$\mathcal{R}_{WBDA+AF}$	$\mathcal{R}_{WBDA+WAF}$	$\mathcal{U}_{WBDA+AFC}$	$\mathcal{R}_{WBDA+DS}$	$\mathcal{U}_{WBDA+DSC}$	$\mathcal{R}_{WBDA+WDS}$	$\mathcal{U}_{WBDA+WDSC}$
C→A	30.53	22.76					41.02	20.15	48.23										
D→A	29.85	26.62	24.32	26.10	29.12	28.91	32.05	30.69	32.88	48.23	39.77	46.24	41.54	44.26	*45.06±0.16*	41.02	*42.37±0.05*	**48.54**	***50.26±0.06***
W→A	29.54	22.65					31.84	26.20	38.20										
A→C	44.08	24.04					40.25	23.06	41.67										
D→C	28.50	26.09	24.04	26.63	26.09	22.80	30.10	32.06	33.48	43.63	38.02	41.14	40.07	41.05	*40.96±0.03*	40.61	*41.19±0.02*	**44.52**	***45.40±0.03***
W→C	26.51	18.08					30.72	25.73	30.10										
A→D	40.13	23.57					36.61	30.75	38.22										
C→D	45.22	24.84	37.58	26.11	36.94	33.76	41.40	26.75	53.50	71.34	59.24	75.80	75.16	80.25	*75.77±0.37*	75.16	*75.69±0.09*	79.62	***83.50±0.55***
W→D	62.82	44.59					77.90	73.25	**82.80**										
A→W	39.66	27.12					40.00	26.10	42.37										
C→W	42.37	26.10	29.49	32.20	39.32	35.25	40.68	19.66	40.34	58.64	51.86	71.53	67.46	73.90	*67.27±0.90*	66.10	*67.15±0.02*	**75.59**	***75.65±0.08***
D→W	65.42	43.73					64.41	63.56	74.24										
Average	40.39	27.52	28.85	27.76	32.87	30.18	40.25	33.16	46.33	55.46	47.22	58.68	56.05	59.86	*57.13*	55.72	*55.99*	**62.38**	***63.61***

TABLE XI
CLASSIFICATION PERFORMANCE OF DIFFERENT FUSION METHODS BASED ON TCA IN OFFICE-31 DATA SETS

Task	$\mathcal{R}_{\text{SVM}}$	$\mathcal{R}_{k-\text{NN}}$	$\mathcal{R}_{\text{CMSD}}$	$\mathcal{R}_{\text{MV}}$	$\mathcal{R}_{\text{AF}}$	$\mathcal{R}_{\text{DS}}$	$\mathcal{R}_{\text{GFK}}$	$\mathcal{R}_{\text{CORAL}}$	$\mathcal{R}_{\text{TCA}}$	$\mathcal{R}_{\text{TCA+CMSD}}$	$\mathcal{R}_{\text{TCA+MV}}$	$\mathcal{R}_{\text{TCA+WMV}}$	$\mathcal{R}_{\text{TCA+AF}}$	$\mathcal{R}_{\text{TCA+WAF}}$	$\mathcal{U}_{\text{TCA+AFC}}$	$\mathcal{R}_{\text{TCA+DS}}$	$\mathcal{U}_{\text{TCA+DSC}}$	$\mathcal{R}_{\text{TCA+WDS}}$	$\mathcal{U}_{\text{TCA+WDSC}}$
D→A	24.51	22.45	27.06	22.94	24.64	24.88	24.76	25.97	19.78	37.99	28.16	31.89	32.77	38.59	*36.15±1.09*	31.31	*32.41±0.38*	**39.44**	***41.27±0.15***
W→A	32.11	27.06					36.04	38.96	38.35										
A→D	27.92	18.83	55.19	24.03	33.77	33.77	24.07	25.97	37.66	57.53	42.21	58.44	56.42	57.79	60.43±0.10	57.14	60.11±0.06	**61.04**	***63.26±0.45***
W→D	54.73	50.65					57.14	39.42	59.74										
A→W	45.96	21.70	46.38	29.36	31.49	34.47	33.62	35.32	46.81	64.47	44.68	59.57	65.11	66.81	67.91±1.02	64.26	67.01±0.08	**67.66**	***69.03±0.05***
D→W	49.36	48.09					39.15	33.89	62.98										
Average	39.10	31.46	42.88	25.44	29.97	31.04	35.80	33.26	44.22	59.99	38.35	49.97	51.43	54.40	*54.83*	50.90	*53.18*	**56.05**	***57.85***

TABLE XII
CLASSIFICATION PERFORMANCE OF DIFFERENT FUSION METHODS BASED ON TCA IN VLSC DATA SETS

Task	$\mathcal{R}_{\text{SVM}}$	$\mathcal{R}_{k-\text{NN}}$	$\mathcal{R}_{\text{CMSD}}$	$\mathcal{R}_{\text{MV}}$	$\mathcal{R}_{\text{AF}}$	$\mathcal{R}_{\text{DS}}$	$\mathcal{R}_{\text{GFK}}$	$\mathcal{R}_{\text{CORAL}}$	$\mathcal{R}_{\text{TCA}}$	$\mathcal{R}_{\text{TCA+CMSD}}$	$\mathcal{R}_{\text{TCA+MV}}$	$\mathcal{R}_{\text{TCA+WMV}}$	$\mathcal{R}_{\text{TCA+AF}}$	$\mathcal{R}_{\text{TCA+WAF}}$	$\mathcal{U}_{\text{TCA+AFC}}$	$\mathcal{R}_{\text{TCA+DS}}$	$\mathcal{U}_{\text{TCA+DSC}}$	$\mathcal{R}_{\text{TCA+WDS}}$	$\mathcal{U}_{\text{TCA+WDSC}}$
L→V	42.64	24.42					40.42	35.07	48.29										
S→V	55.42	27.64	37.30	26.89	36.11	39.52	45.91	37.44	57.06	65.08	58.69	63.60	64.78	**66.67**	*64.87±0.06*	63.75	*65.04±0.04*	65.68	***65.82±0.28***
C→V	23.03	54.09					47.55	52.01	48.14										
V→L	55.58	50.28					47.83	50.47	55.58										
S→L	51.23	47.83	52.74	48.60	51.06	51.44	44.05	39.70	50.66	54.63	52.17	53.69	56.71	54.81	*59.92±0.47*	52.74	*55.20±0.17*	**56.71**	***59.69±0.31***
C→L	47.30	48.41					25.14	51.80	34.78										
V→S	54.53	54.53					50.50	51.80	59.28										
L→S	39.14	29.21	34.96	41.15	44.17	41.87	32.66	34.53	37.41	56.69	45.61	53.24	49.50	56.40	*53.92±0.21*	49.64	*50.74±0.15*	**59.71**	***61.76±0.14***
C→S	17.84	42.01					21.51	42.16	29.64										
V→C	71.07	51.46					74.41	63.84	79.83										
L→C	32.27	16.83	31.02	22.67	28.23	36.44	27.26	22.25	59.25	80.84	77.47	78.58	77.89	80.53	*74.26±0.26*	75.38	*74.83±0.10*	**81.22**	*81.00±0.07*
S→C	38.53	21.56					34.24	32.28	64.81										
Average	44.05	37.81	39.01	34.83	39.89	42.32	40.96	38.46	52.06	64.31	58.49	62.28	62.22	64.60	63.25	60.38	61.45	**65.83**	***67.05***

domain is quite different from that of target domain, it means this source domain cannot provide a lot of useful knowledge for the classification of query patterns, and the the corresponding classification result represented by evidence should be discounted by a small weight before the combination. The experimental results also prove that it is desirable to discount evidence based on the domain-consistency between the source and target domains.

We find the classification accuracy with individual source domain quite vary in a few cases. In such extreme cases, the classification results with very low accuracy may have harmful influence on the combination, but it is hard to completely eliminate this negative influence only by tuning the weighting factor. So the accuracy of combined result is lower than the maximum accuracy of individual results in such case. Nevertheless, the proposed Weighted DS combination method still produces higher accuracy than the other combination methods in general.

For decision making support, we find the average utility value of cautious decision making rule is usually higher than the traditional hard decision making rule using $BetP(\cdot)$, i.e., TCA/JDA/TJM/BDA/WBDA+WDS. This is because some patterns difficult to clearly distinguish are assigned to the disjunction of two possible classes by the cautious decision. The utility value of an imprecise decision is bigger than that of an error. Therefore, the cautious decision making rule can efficiently reduce errors by properly modeling the partial imprecision. Moreover, we can see that the variance with different numbers of neighbors is very small, because the neighbors far from the classification result of an object play a minor role in decision making. It means that the choice of K has a small influence on the decision making performance, and the proposed method is robust to the choice of K value. This is an interesting property for applications. Overall, the proposed method Weighted DS combination method jointly working with the cautious decision making usually captures the highest average utility value. It shows that the proposed method generally outperforms the other methods. If we want to further improve the classification accuracy, some more prior or training information or other techniques should be used to deal with these patterns hard to specifically classify.

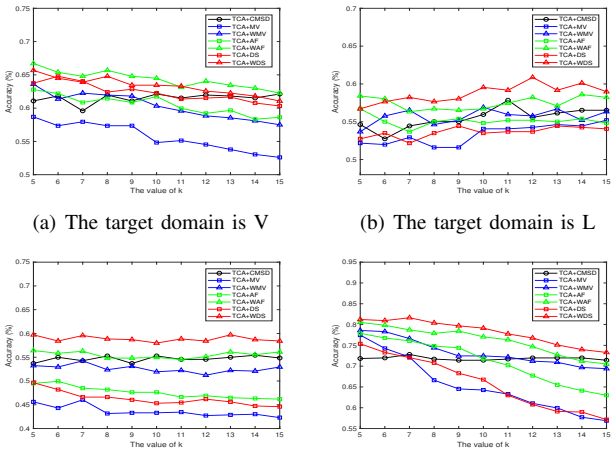
In this experiment, k-NN is employed as the base classifier, and the selection of k remains an open problem. We have tested the influence of tuning of $k \in \{5, \ldots, 15\}$ on the classification performance. The classification accuracy curves of some combination methods (i.e., CMSD, MV, WMV, AF, WAF, DS and WDS) with different k values are shown in Fig. 3, and the mean classification accuracy with variance are reported in Table XIII. In Fig. 3, one can see the accuracy of the proposed method and the others varies with different k values, but the accuracy of our proposed WDS method does not change too much. Moreover, WDS method generally produces higher accuracy than other methods with different k values in most cases.

E. Influence of Parameter Tuning

Two parameters α and K are involved in the proposed cautious decision making. We validate the influence of the parameters on the classification performance using several benchmarks on JDA. The average utility value with respect to α and K are shown in Figs. 4-5. For parameter α, one can see that the average utility value gets smaller when the value of α increases. This is because $\frac{|A \cap \{\omega\}|}{|A \cup \{\omega\}|}$ is smaller than one in Eq. (18), and the bigger α value yields a small utility value. The choice of α mainly depends on the error cost in actual applications. If the error cost is high, the partial imprecision will be considered much better than error, and a small α value is preferred. For parameter K, we can see the average utility

TABLE XIII
CLASSIFICATION ACCURACY (MEAN ± VARIANCE) OF DIVERSE FUSION METHODS BASED ON TCA IN VLSC DATA SETS WITH DIFFERENT k VALUES

Task	$\mathcal{R}_{k-\mathrm{NN}}$	$\mathcal{R}_{\mathrm{CMSD}}$	$\mathcal{R}_{\mathrm{MV}}$	$\mathcal{R}_{\mathrm{AF}}$	$\mathcal{R}_{\mathrm{DS}}$	$\mathcal{R}_{\mathrm{CORAL}}$	$\mathcal{R}_{\mathrm{TCA}}$	$\mathcal{R}_{\mathrm{TCA+CMSD}}$	$\mathcal{R}_{\mathrm{TCA+MV}}$	$\mathcal{R}_{\mathrm{TCA+WMV}}$	$\mathcal{R}_{\mathrm{TCA+AF}}$	$\mathcal{R}_{\mathrm{TCA+WAF}}$	$\mathcal{R}_{\mathrm{TCA+DS}}$	$\mathcal{R}_{\mathrm{TCA+WDS}}$
L→V	23.40±1.72					35.36±0.87	44.83±4.25							
S→V	27.39±0.14	37.94±2.04	27.93±2.13	39.90±3.12	44.74±6.93	37.88±0.37	55.59±0.48	61.55±0.54	55.69±4.50	60.39±3.93	62.52±2.16	**64.34±1.74**	62.34±1.97	63.34±1.85
C→V	53.55±0.22					53.94±1.15	45.37±2.97							
V→L	52.59±1.02					52.59±1.43	57.74±1.32							
S→L	46.45±0.47	54.29±1.07	48.27±0.49	50.78±0.22	51.54±0.71	35.04±4.03	50.66±0.28	55.49±1.82	53.36±1.76	55.70±0.93	55.11±0.50	57.48±0.71	53.63±0.49	**58.74±1.47**
C→L	48.05±0.26					51.81±0.27	36.40±1.75							
V→S	51.11±5.01					50.75±1.84	56.29±1.50							
L→S	29.01±1.03	35.36±2.62	41.49±0.26	43.56±0.14	43.30±0.30	34.91±0.75	38.18±0.58	57.68±0.36	43.65±1.72	52.65±0.65	47.63±1.65	55.50±0.46	46.28±2.25	**58.85±0.34**
C→S	42.42±0.12					42.83±0.22	26.79±2.29							
V→C	48.11±3.08					60.32±4.61	75.88±1.79							
L→C	19.43±2.45	32.12±5.24	22.73±0.71	34.37±1.77	40.04±1.67	23.73±1.53	48.70±2.65	74.48±0.16	65.29±4.58	73.30±10.61	71.12±2.86	76.18±1.18	65.96±4.18	**78.18±9.00**
S→C	20.60±0.17					31.89±0.28	52.08±3.95							

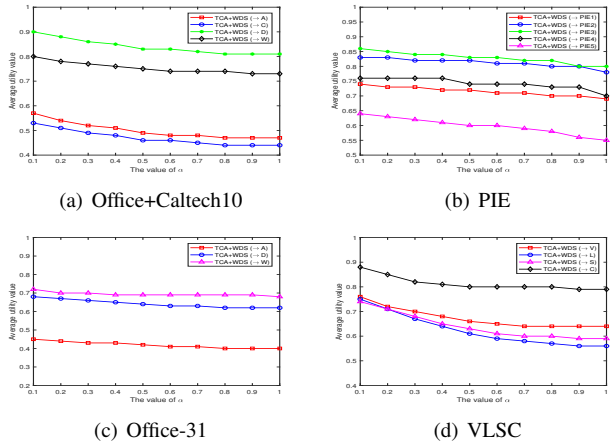


(a) The target domain is V (b) The target domain is L

(c) The target domain is S (d) The target domain is C

Fig. 3. Classification accuracy of different methods based on TCA and k-NN classifier with tuning of k values.

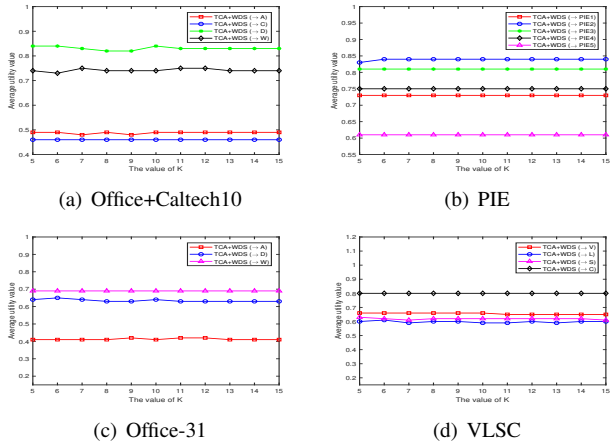
value is not sensitive to K. Since the influence of distance from the query object to the neighbors has been considered in the proposed cautious decision rule, and the neighbor far from object will play a small role in computing the utility value. So the performance of the cautious decision making rule is robust to the tuning of K, which is convenient for applications.



(a) Office+Caltech10 (b) PIE

(c) Office-31 (d) VLSC

Fig. 4. Average utility value of different TCA-based methods with tuning of α.



(a) Office+Caltech10 (b) PIE

(c) Office-31 (d) VLSC

Fig. 5. Average utility value of different TCA-based methods with tuning of K.

V. CONCLUSION

A decision-level combination method for multi-source domain adaptation based on evidential reasoning has been proposed to improve the classification accuracy in the target domain. The classification results (soft outputs) obtained by the auxiliary of different source domains usually can provide complementary knowledge in different quality to the target domain for pattern classification. The distribution distance between the source and target domains is employed to evaluate the reliability/weight of corresponding classification results. The larger the distance, the smaller the weight. This weight is used to discount the corresponding classification results, and the discounted results are combined by Dempster's rule. In order to further reduce the error rate, a cautious decision rule committing the objects hard to classify to the disjunction of several classes is presented. This rule can significantly reduce the error rate by properly modeling the partial imprecision of classification. Real data sets have been used in the experimental application to test the performance of the proposed method compared with other related fusion methods. The experimental results show that our method considering multiple source domains generally produces higher accuracy than that of any individual source domains, and the proposed Weighted DS fusion method outperforms the traditional DS fusion method, Average Fusion method and Majority Vote method. These

results demonstrate the effectiveness of the new method. In some applications, the source and target domains may have different class labels. In the future work, we will attempt to solve such challenging and interesting transfer classification problem via the combination of multiple source domains with different label spaces.

ACKNOWLEDGMENT

This work has been partially supported by National Natural Science Foundation of China (No. 61672431, No. 61790552, No. 61790554), Shaanxi Science Fund for Distinguished Young Scholars (No. 2018JC-006) and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] Y. Cao, M. Long, and J. Wang, "Unsupervised domain adaptation with distribution matching machines," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[2] R. Gopalan, R. Li, and R. Chellappa, "Unsupervised adaptation across domain shifts by generating intermediate data representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2288–2302, 2014.

[3] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proceedings of the Computer Vision and Pattern Recognition*, 2012, pp. 2066–2073.

[4] Y. Xu, S. J. Pan, H. Xiong, Q. Wu, R. Luo, H. Min, and H. Song, "A unified framework for metric transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 6, pp. 1158–1171, 2017.

[5] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Network*, vol. 22, no. 2, pp. 199–210, 2011.

[6] J. Wang, Y. Chen, L. Hu, X. Peng, and P. S. Yu, "Stratified transfer learning for cross-domain activity recognition," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*, 2018.

[7] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2200–2207.

[8] J. Tahmoresnezhad and S. Hashemi, "Visual domain adaptation via transfer feature learning," *Knowledge and Information Systems*, vol. 50, no. 2, pp. 585–605, 2017.

[9] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1410–1417.

[10] J. Wang, Y. Chen, S. Hao, W. Feng, and Z. Shen, "Balanced distribution adaptation for transfer learning," in *Proceedings of the IEEE International Conference on Data Mining*, 2017, pp. 1129–1134.

[11] B. Sun and K. Saenko, "Subspace distribution alignment for unsupervised domain adaptation." in *Proceedings of the British Machine Vision Conference*, 2015.

[12] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2960–2967.

[13] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

[14] Z. Jing, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[15] L. Duan, D. Xu, and I. W.-H. Tsang, "Domain adaptation from multiple sources: A domain-dependent regularization approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 504–518, 2012.

[16] S. Sun, H. Shi, and Y. Wu, "A survey of multi-source domain adaptation," *Information Fusion*, vol. 24, pp. 84–92, 2015.

[17] Z. Ding, M. Shao, and Y. Fu, "Incomplete multi-source transfer learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 2, pp. 310–323, 2018.

[18] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the International Conference on Machine learning*, 2007, pp. 193–200.

[19] Y. Freund and R. E. Schapire, *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*. Springer Berlin Heidelberg, 1995.

[20] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 2010, pp. 1855–1862.

[21] P. Huang, G. Wang, and S. Qin, "Boosting for transfer learning from multiple data sources," *Pattern Recognition Letters*, vol. 33, no. 5, pp. 568–579, 2012.

[22] Y. Zhu, Y. Chen, Z. Lu, S. J. Pan, G. R. Xue, Y. Yu, and Q. Yang, "Heterogeneous transfer learning for image classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2011, pp. 1304–1309.

[23] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the International Conference on Machine Learning.*, 2007, pp. 759–766.

[24] G. Matasci, M. Volpi, M. Kanevski, L. Bruzzone, and D. Tuia, "Semisupervised transfer component analysis for domain adaptation in remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 7, pp. 3550–3564, 2015.

[25] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[26] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1019–1034, 2017.

[27] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine Learning*, vol. 79, no. 1-2, pp. 151–175, 2010.

[28] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2007, pp. 137–144.

[29] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *The annals of mathematical statistics*, pp. 325–339, 1967.

[30] G. Shafer, *A mathematical theory of evidence*. Princeton university press, 1976, vol. 42.

[31] T. Denœux, "Logistic regression, neural networks and dempster-shafer theory: a new perspective," *Knowledge-Based Systems*, vol. 176, pp. 54–67, 2019.

[32] ——, "A k-nearest neighbor classification rule based on dempster-shafer theory," *IEEE Transactions on System, Man and Cybernetics.*, vol. 25, no. 5, pp. 804–813, 1995.

[33] Z.-g. Liu, Y. Liu, J. Dezert, and F. Cuzzolin, "Evidence combination based on credal belief redistribution for pattern classification," *IEEE Transactions on Fuzzy Systems*, 2019.

[34] Z.-g. Liu, Q. Pan, J. Dezert, and G. Mercier, "Credal c-means clustering method based on belief functions," *Knowledge-based systems*, vol. 74, pp. 119–132, 2015.

[35] Z.-g. Liu, Q. Pan, J. Dezert, and A. Martin, "Combination of classifiers with optimal weight based on evidential reasoning," *IEEE Transaction on Fuzzy System*, vol. 26, no. 3, pp. 1217–1230, 2018.

[36] Z.-g. Liu, Q. Pan, J. Dezert, J.-W. Han, and Y. He, "Classifier fusion with contextual reliability evaluation," *IEEE Transactions on Cybernetics*, vol. 48, no. 5, pp. 1605–1618, 2018.

[37] M. Beynon, B. Curry, and P. Morgan, "The dempster-shafer theory of evidence: an alternative approach to multicriteria decision modelling," *Omega*, vol. 28, no. 1, pp. 37–50, 2000.

[38] J. Dezert and J.-M. Tacnet, "Evidential reasoning for multi-criteria analysis based on dsmt-ahp," *Proceedings of the Advances and Applications of DSmT for Information Fusion*, p. 95, 2015.

[39] T. Denœux, "Analysis of evidence-theoretic decision rules for pattern classification," *Pattern Recognition.*, vol. 30, no. 7, pp. 1095–1107, 1997.

[40] V. N. Huynh, Y. Nakamori, T. B. Ho, and T. Murai, "Multiple-attribute decision making under uncertainty: The evidential reasoning approach revisited," *IEEE Transactions on Systems Man and Cybernetics - Part A Systems and Humans*, vol. 36, no. 4, pp. 804–822, 2006.

[41] J. B. Yang and D. Xu, "On the evidential reasoning algorithm for multiple attribute decision analysis under uncertainty," *IEEE Transactions on Systems Man and Cybernetics - Part A Systems and Humans*, vol. 32, no. 3, pp. 289–304, 2002.

[42] T. Denœux, "A neural network classifier based on dempster-shafer theory," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 30, no. 2, pp. 131–150, 2000.

[43] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2006, pp. 120–128.

[44] J. Wang, W. Feng, Y. Chen, Y. Han, M. Huang, and P. S. Yu, "Visual domain adaptation with manifold embedded distribution alignment," in *Proceedings of the ACM Multimedia Conference*, 2018.

[45] R. M. Rodrłguez, L. Martłnez, and F. Herrera, "Hesitant fuzzy linguistic term sets for decision making," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 109–119, 2012.

[46] C. Fu, W. Chang, W. Liu, and S. Yang, "Data-driven group decision making for diagnosis of thyroid nodule," *Sciece China Information Sciences*, vol. 62, no. 11, pp. 151–173, 2019.

[47] J. D. C. Juan, J. Dłez, and A. Bahamonde, "Learning nondeterministic classifiers," *Journal of Machine Learning Research*, vol. 10, pp. 2273–2293, 2009.

[48] M. Zaffalon, G. Corani, and D. Mau, "Evaluating credal classifiers by utility-discounted predictive accuracy," *International Journal of Approximate Reasoning*, vol. 53, no. 8, pp. 1282–1301, 2012.

[49] L. Ma and T. Denoeux, "Making set-valued predictions in evidential classification: A comparison of different approaches," in *Proceedings of the International Symposium on Imprecise Probabilities: Theories and Applications*, 2019, pp. 276–285.

[50] J. L. V. Neumann and O. V. Morgenstern, "The theory of games and economic behavior," *Princeton University Press Princeton N J*.

[51] D. D. Mau, "Equivalences between maximum a posteriori inference in bayesian networks and maximum expected utility computation in influence diagrams," *International Journal of Approximate Reasoning*.

[52] T. Denœux, "Decision-making with belief functions: A review," *International Journal of Approximate Reasoning*, vol. 109, pp. 87–110, 2019.

[53] P. Smets, "Decision making in the tbm: the necessity of the pignistic transformation," *International Journal of Approximate Reasoning*, vol. 38, no. 2, pp. 133–147, 2005.