

“© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Unconstrained fuzzy feature fusion for heterogeneous unsupervised domain adaptation

Feng Liu, Guangquan Zhang, Jie Lu

Decision Systems and e-Service Intelligence (DeSI) Lab,
Centre for Artificial Intelligence, Faculty of Engineering and Information Technology
University of Technology Sydney
Sydney, Australia

Feng.Liu-2@student.uts.edu.au; {Guangquan.Zhang, Jie.Lu}@uts.edu.au

Abstract— Domain adaptation can transfer knowledge from the source domain to improve pattern recognition accuracy in the target domain. However, it is rarely discussed when the target domain is unlabeled and heterogeneous with the source domain, which is a very challenging problem in the domain adaptation field. This paper presents a new feature reconstruction method: unconstrained fuzzy feature fusion. Through the reconstructed features of a source and a target domain, a geodesic flow kernel is applied to transfer knowledge between them. Furthermore, the original information of the target domain is also preserved when reconstructing the features of the two domains. Compared to the previous work, this work has two advantages: 1) the sum of the memberships of the original features to fuzzy features no longer must be one, and 2) the original information of the target domain is persevered. As a result of these advantages, this work delivers a better performance than previous studies using two public datasets.

Keywords—transfer learning, domain adaptation, fuzzy features, machine learning

I. INTRODUCTION

How to leverage the knowledge gained from a source domain to help improve the recognition performance of a target domain is an important and challenging problem in the field of machine learning. In the last decade, domain adaptation models have been proposed to address this problem and have been applied in many fields [1], [2]. For example, the knowledge to recognize trucks can be used to recognize cars; credit records collected from Germany can be applied to improve the assessment accuracy of Australia's credit records which suffer from data shortage; or bank failure data from the United States can be used to help predict Australian banks' failures. These examples clearly show that classification accuracy is improved while the cost of manually labeling a huge dataset in a new, almost unknown field is reduced if we use domain adaptation models [3]. The amount of data in the current era is growing fast, which means that data labelling comes at a high cost or is an impossible mission. Using the knowledge (labeled data) from source domains to explore unknown areas (unlabeled but related data) has thus become extremely important.

To date, five types of domain adaptation models have been well studied: 1) homogeneous supervised domain adaptation [4]; 2) homogeneous semi-supervised domain adaptation [5]; 3) homogeneous unsupervised domain

adaptation [6]; 4) heterogeneous supervised domain adaptation [7]; and 5) heterogeneous semi-supervised domain adaptation [8]. In each of these, "homogeneous" represents source domains and target domains have the same number of related features, and "heterogeneous" represents source domains and target domains have a different number of related features.

For supervised and semi-supervised domain adaptation, however, existing models need labeled instances in the target domain, which is an obstacle to applying these models to successfully accomplish a classification task in an unlabeled target domain. To deal with unsupervised domain adaptation, existing models are limited to source domains that have the homogeneous feature space as the target domain, which does not apply to some situations in the current environment. It is therefore necessary to consider another type of domain adaptation: heterogeneous unsupervised domain adaptation (HeUDA). This is a common situation in the current era because it is difficult to find a homogeneous source domain for a specific target domain. Kernel canonical correlation analysis (KCCA) [9] has been proposed in the existing work for heterogeneous unsupervised domain adaptation when there are paired instances in both domains. However, when the source domain and the target domain do not have paired instances, KCCA is invalid. In our previous work [10], we proposed a fuzzy feature fusion-geodesic flow kernel (FFF-GFK) model to address the heterogeneous unsupervised domain adaptation problem. But the FFF-GFK model has two shortcomings: 1) the sum of the memberships of each original feature to all fuzzy features must be one; and 2) the structure information of the unlabeled target domain is not preserved when constructing fuzzy features.

To overcome the shortcomings of FFF-GFK, this paper presents a novel HeUDA model based on unconstrained fuzzy features, which comprises two parts: 1) the unconstrained fuzzy feature fusion method which addresses the first shortcoming; and 2) a regularizer to preserve the original information of the unlabeled target domain which addresses the second shortcoming. After constructing unconstrained fuzzy features of each domain, knowledge, through these fuzzy features, can be transferred from a source domain to a target domain by the geodesic flow kernel (GFK) [6]. Compared to KCCA and FFF-GFK, the proposed model, referred to as UFFFG (unconstrained fuzzy feature fusion geodesic flow kernel), has the following advantages:

- 1) UFFFG does not require the source domain and the target domain to have the same number of instances;
- 2) UFFFG does not require the source domain and the target domain to have paired instances;
- 3) UFFFG can preserve the original information contained in the target domain.

These advantages result in the proposed model having better performance than KCCA, FFF-GFK and other benchmarks. The main contributions of this paper are as follows:

- 1) A novel unconstrained fuzzy feature fusion method to construct fuzzy features for each domain. This method does not require that the sum of memberships of each original feature to all fuzzy fusion features must be one;
- 2) Original information on the target domain is preserved via a regularizer, which avoids the loss of information of the target domain when constructing fuzzy features.

This paper is organized as follows. Section II reviews the related work. Section III introduces the UFFFG model. Section IV demonstrates the classification results of the UFFFG model and benchmarks. Section V concludes the paper and discusses future work.

II. RELATED WORK

In this section, three types of domain adaptation models developed in recent years are reviewed, these being: supervised domain adaptation (SDA), semi-supervised domain adaptation (SSDA) and unsupervised domain adaptation (UDA).

A. Supervised domain adaptation

Many models have been proposed in relation to the traditional research topic of SDA. SDA models only use labeled instances in source and target domains to train prediction models, and unlabeled instances are unable to improve the training accuracy of the prediction models. There are two categories of SDA, based on the relationship between the feature spaces of the source domain and the target domain: homogeneous supervised domain adaptation (HoSDA) and heterogeneous supervised domain adaptation (HeSDA). The representative HoSDA models are transfer component analysis (TCA) [4], geodesic flow kernel (GFK), information-theoretical learning (ITL) [11], adaptive support vector machine (A-SVM) [12], projective model transfer SVM (PMT-SVM) [13], structure-aware adaptive structural SVM (SA-SSVM) [14] and stationary subspace analysis (SSA) [15]. The representative HeSDA models are asymmetric regularized cross-domain transformation (ARC-t) [16], and heterogeneous feature augmentation (HFA) [7].

B. Semi-supervised domain adaptation

SSDA models aim to apply the information of unlabeled data in the source and target domains to help improve classification accuracy. Similar to SDA models, there are two categories of SSDA: homogeneous semi-supervised domain adaptation (HoSSDA) and heterogeneous semi-supervised domain adaptation (HeSSDA). The representative HoSSDA models are covariate shift [17], self-labeling adaptation [5], semi-supervised kernel matching for domain adaptation (SSKMDA) [8] and fuzzy sets [18]–[20]. The representative HeSSDA models are heterogeneous spectral

mapping (HeMap) [21], manifold alignment-based models (MA) [22], SSKMDA [8], distant domain transfer learning (DDTL) [23] and the DASH-N method [24].

C. Unsupervised domain adaptation

When there are no labeled instances in the target domain, the knowledge transfer process is regarded as UDA. There is one basic assumption: two domains are related and have similar prediction tasks. Since it is difficult to transfer knowledge from heterogeneous source domains to an unlabeled target domain, the principal UDA technologies deal with the issue as if the source domain and target domain are homogeneous. This kind of technology is known as homogeneous unsupervised domain adaptation (HoUDA). The representative HoUDA models are similar to the HoSDA models and include TCA, GFK, ITL, SSA, transfer deep network [25], optimal transport [26], scatter component analysis [27], conditional transferable components [28] and correlation alignment [29]. Recently, the generative adversarial network is applied to develop HoUDA models and has satisfied performance in the computer vision field [30], [31]. As discussed in Section I, kernel canonical correlation analysis (KCCA) and FFF-GFK are HeUDA models.

III. HETEROGENEOUS UNSUPERVISED DOMAIN ADAPTATION VIA UNCONSTRAINED FUZZY FEATURE FUSION

This section introduces the proposed model based on unconstrained fuzzy feature fusion. To clearly demonstrate our model, we first introduce the notations used in this section, and the second subsection details the motivation of our paper. The unconstrained fuzzy feature fusion method is then proposed in subsection C. The final subsection presents a novel UFFFG model to address the situation in which the target domain is unlabeled and has a heterogeneous feature space with the source domain.

A. Notations and problem statement

We assume that there is one source domain D_S and one target domain D_T with different feature spaces, in which $D_S = \{X_{Si}, y_{Si}\}$ and $D_T = \{X_{Ti}\}$, where $X_{Si} \in \mathbb{R}^n$, $X_{Ti} \in \mathbb{R}^m$, $y_{Si} \in \{-1, 1\}$, $m \neq n$ and $i = 1, 2, \dots, L$ (L is the number of instances in each domain). We aim to label each instance in D_T using the knowledge from D_S . It is important to clarify that there are no labeled instances in the target domain.

B. Motivation

When a source domain and a target domain have different feature spaces, the model trained by the source domain cannot be applied to label instances of the target domain. A common idea is thus to map two domains onto the same feature space so that instances of the target domain can be predicted by the model trained by the source domain. There are three common ways to do this: 1) dimension reduction, 2) feature selection, and 3) feature fusion [32]. In our previous work, we used the feature fusion method to map two domains onto the same feature space. However, there are two shortcomings: 1) the sum of the memberships of each original feature to all fuzzy features must be one, and 2) the structure information of the unlabeled target domain is not preserved when constructing fuzzy features.

It is not necessary that the sum of the membership of each original feature to the constructed fuzzy features is one because memberships are possibilities rather than probabilities. For example, we assume that the membership of feature f_i to fuzzy feature f_i^{fuzzy} is 0.8 and it is probable that the membership of feature f_i to fuzzy feature f_{i+1}^{fuzzy} is 0.6. It is clear that $0.8+0.6>1$. If we constrain these memberships to be summed to one, fuzzy information among these features will be lost. So, it could be helpful to improve the performance of the target domain if we retain this fuzzy information. From the perspective of the structure of instances of the target domain, our previous work did not consider preserving the structure information of the target domain, which may result in the loss of useful information when constructing fuzzy features (if the target domain can be clearly clustered and we do not consider preserving the structure information, a lot of information will be lost).

Hence, this paper presents an advanced fuzzy feature fusion method which preserves two kinds of information contained in the original domains when constructing fuzzy features. Through these fuzzy features, knowledge from a source domain can be transferred to a heterogeneous and unlabeled target domain.

C. Unconstrained fuzzy feature fusion

In this section, the proposed model is introduced in three parts: 1) unconstrained fuzzy features, 2) structure preservation regularizer, and 3) learning progress.

1) Unconstrained fuzzy features

A new feature fusion method, unconstrained fuzzy feature fusion, is proposed to overcome the drawbacks of the previous fuzzy feature fusion method. This new method has a higher probability of maintaining relationships between the label spaces of the source domain and the target domain than traditional dimension reduction methods, as shown in the experiment section of this paper. We first provide a definition of unconstrained fuzzy features.

Definition 1 (unconstrained fuzzy features). Consider that there are n original features denoted by f_1, \dots, f_n , the i^{th} fuzzy feature is constructed using the following formula ($i = 1, 2, \dots, r$).

$$f_i^{fuzzy} = \frac{\sum_{j=1}^n \mu_{ij} f_j}{\sum_{j=1}^n \mu_{ij}}, \sum_{j=1}^n \mu_{ij} > 0, \quad (1)$$

where μ_{ij} is the membership grade of the j^{th} original feature to the i^{th} fuzzy feature.

From Definition 1, it is clear that fuzzy features are combined by original features with corresponding membership grades. Calculating μ_{ij} is thus the key to fuzzy features. For a clear demonstration of μ_{ij} , the definition of an unconstrained fuzzy features matrix follows.

Definition 2 (unconstrained fuzzy features matrix). Given an instance matrix $X \in \mathbb{R}^{L \times n}$ where L is the number of instances and n is the number of features, an instance matrix based on fuzzy features can be expressed by the following term:

$$X^{fuzzy} = \begin{pmatrix} \frac{1}{\sum_{j=1}^n \mu_{1j}} & \dots & \frac{1}{\sum_{j=1}^n \mu_{rj}} \\ \vdots & \ddots & \vdots \\ \frac{1}{\sum_{j=1}^n \mu_{1j}} & \dots & \frac{1}{\sum_{j=1}^n \mu_{rj}} \end{pmatrix} \odot XU^T, \sum_{i=1}^r \mu_{ij} > 0.$$

$U \in [0, 1]^{r \times n}$ is a membership matrix, and μ_{ij} is the element in the i^{th} row and j^{th} column of U , and \odot represents the element-wise product of vectors.

Compared to the previous definition of fuzzy features [10], $\sum_{i=1}^r \mu_{ij}$ is more than 0 rather than being equal to 1. In addition, we hope the sequence $\sum_{i=1}^r \mu_{ij}$ ($j = 1, 2, \dots, n$) has a small variation value which means each original feature should be close to at least one fuzzy feature. Therefore, the basic cost function of unconstrained fuzzy feature fusion can be given in Definition 3.

Definition 3 (cost function J_1). Given an instance matrix $X \in \mathbb{R}^{L \times n}$ and a corresponding membership matrix $U \in [0, 1]^{r \times n}$ where L is the number of instances and n is the number of features, the cost function J_1 of the proposed model is defined as follows.

$$J_1(U, X) = \sum_{i=1}^r \sum_{j=1}^n \mu_{ij} \left\| X_{*j} - \frac{1}{\sum_{j=1}^n \mu_{ij}} \odot XU_{i*}^T \right\|^2 + \lambda_v Reg_v, \quad (2)$$

where

$$Reg_v = \sqrt{\frac{1}{n} \sum_{j=1}^n \left(\sum_{i=1}^r \mu_{ij} - \frac{1}{n} \sum_{j=1}^n \sum_{i=1}^r \mu_{ij} \right)^2},$$

X_{*j} is the j^{th} column of X , U_{i*} is the i^{th} row of U , $\lambda_v > 0$ is an appropriately chosen regularization parameter and μ_{ij} is the element in the i^{th} row and j^{th} column of U .

By minimizing J_1 , we can obtain the best unconstrained fuzzy features matrix and use this matrix to construct the fuzzy features (without structure preservation).

2) Structure preservation regularizer

In this section, we introduce how to preserve the structure of the original target domain. We can use the following term to express the situation of the structure preservation of the target domain.

$$\sum_{i=1}^L \sum_{j=1}^L w_{ij} \|X_{Ti} - X_{Tj}\|^2, \quad (3)$$

where w_{ij} is the similarity between instances X_{Ti} and X_{Tj} of the target domain. Then, we hope w_{ij} can be preserved, which means we need to minimize the following term.

$$\sum_{i=1}^L \sum_{j=1}^L w_{ij} \|X_{Ti}^{fuzzy} - X_{Tj}^{fuzzy}\|^2. \quad (4)$$

Based on [33], term (4) can be expressed by

$$Trace \left((X_T^{fuzzy})^T LX_T^{fuzzy} \right), \quad (5)$$

where $L = D - W$, D is a diagonal matrix whose (j, j) -entry is the sum of j^{th} row of similarity matrix W of the original target domain.

For the target domain, one could compute the pair-wise similarity w_{ij} using different functions, of which two typical choices are cosine measurement and RBF measurement, $w_{ij} = \exp(\|x_i - x_j\|^2 / \sigma^2)$. The choice of similarity function is usually domain-specific. For example, with text data (term frequency), cosine measurement is empirically often a better choice in characterizing documents with similar (proportional) word counts. In this paper, we choose the RBF as the similarity function and set σ as the median value of the distance matrix $\|x_i - x_j\|$.

3) Learning progress

After constructing cost function \mathbf{J}_1 and structure preservation regularizer, the final cost function is given in Definition 4.

Definition 4 (cost function). Given instance matrixes $X_S \in \mathbb{R}^{L \times n}$, $X_T \in \mathbb{R}^{L \times m}$ and corresponding membership matrixes $U_S \in [0, 1]^{r \times n}$ and $U_T \in [0, 1]^{r \times m}$ where L is the number of instances and n, m are the number of features, the cost function of the proposed model is defined as follows:

$$J(U_S, X_S, U_T, X_T) = \mathbf{J}_1(U_S, X_S) + \mathbf{J}_1(U_T, X_T) + \lambda_s \text{Trace} \left((X_T^{\text{fuzzy}})^T L X_T^{\text{fuzzy}} \right), \quad (6)$$

where

$$X_T^{\text{fuzzy}} = \begin{pmatrix} 1 & & 1 \\ \frac{1}{\sum_{j=1}^n \mu_{1j}^T} & \cdots & \frac{1}{\sum_{j=1}^m \mu_{rj}^T} \\ \vdots & \ddots & \vdots \\ 1 & & 1 \\ \frac{1}{\sum_{j=1}^n \mu_{1j}^T} & \cdots & \frac{1}{\sum_{j=1}^m \mu_{rj}^T} \end{pmatrix} \odot X_T U_T^T,$$

μ_{ij}^T is the element in the i^{th} row and j^{th} column of U_T .

We can thus establish our model by minimizing the cost function \mathbf{J} . In detail, the proposed method can be expressed as:

Unconstrained fuzzy feature fusion. Given instance matrixes $X_S \in \mathbb{R}^{L \times n}$, $X_T \in \mathbb{R}^{L \times m}$ where L is the number of instances and n, m are the number of features, the proposed method, denoted by unconstrained fuzzy feature fusion, aims to obtain the fuzzy feature matrixes by finding the best membership matrix U_S^* and U_T^* . They satisfy the following equation:

$$J(U_S^*, X_S, U_T^*, X_T) = \min_{U_S, U_T} J(U_S, X_S, U_T, X_T). \quad (7)$$

To obtain U_S^* and U_T^* , we adopt evolutionary algorithms to optimize the cost function \mathbf{J} . Two evolutionary algorithms are considered in this paper: the cuckoo search (CS) algorithm and the differential evolution (DE) algorithm. To use evolutionary algorithms, we need to change U_S and U_T to one vector, which can be expressed by

$$V = (\mu_{11}^S, \dots, \mu_{1n}^S, \dots, \mu_{r1}^S, \dots, \mu_{rn}^S, \mu_{11}^T, \dots, \mu_{1m}^T, \dots, \mu_{r1}^T, \dots, \mu_{rm}^T).$$

By choosing the most appropriate parameters of the CS and DE, we can obtain the best V^* , and turn V^* into U_S^* and U_T^* .

Algorithm 1 briefly demonstrates the unconstrained fuzzy feature fusion method.

Algorithm 1. Unconstrained fuzzy feature fusion

Input:

\mathbf{X}_S : a matrix representing the source domain
 \mathbf{X}_T : a matrix representing the target domain
 r : the number of fuzzy features
 ϵ : minimum posteriori error

Output:

FFM_S (fuzzy features matrix) generated by \mathbf{X}_S
FFM_T (fuzzy features matrix) generated by \mathbf{X}_T

- 1 $[l, n] = \text{size}(\mathbf{X}_S)$; % to obtain the size of \mathbf{X}_S .
 - 2 $[l, m] = \text{size}(\mathbf{X}_T)$; % to obtain the size of \mathbf{X}_T .
 - 3 $V = \text{rand}(1, r * (m + n))$; % to initialize parameters.
 - 4 $V^* = \text{CS}(V, \mathbf{X}_S, \mathbf{X}_T, \epsilon)$; % to optimize V .
 - 5 $U_S = \text{reshape}(V^*(1:n*r), r, n)$; % to generate U_S .
 - 6 $U_T = \text{reshape}(V^*(n*r+1:end), r, m)$; % to generate U_T .
 - 7 FFM_S = $\mathbf{X}_S U_S^T$;
 - 8 FFM_T = $\mathbf{X}_T U_T^T$;
 - 9 **Return** FFM_S and FFM_T.
-

D. A novel fuzzy-based HeUDA model

This section details the four procedures of the model presented in this paper, namely: 1) data pre-processing, 2) unconstrained fuzzy feature fusion, 3) domain adaptation, and 4) prediction. Figure 1 shows the steps of the proposed model.

1) Data pre-processing

For two feature spaces X_S and X_T , we adopt the same method of normalizing them, which can be expressed using the following functions:

$$f_{pp}(X_{*j}) = \frac{X_{*j} - \min(X_{*j})}{\max(X_{*j}) - \min(X_{*j})}, j = 1, \dots, n,$$

$$F_{pp}(X) = (f_{pp}(X_{*1}) f_{pp}(X_{*2}) \dots f_{pp}(X_{*n})).$$

X_{*j} is the j^{th} column of X and n is the number of features of X . $f_{pp}(X_{*j})$ is an L by 1 vector and $F_{pp}(X)$ is an L by n matrix.

We can then obtain two pre-processed feature spaces using $X_S = F_{pp}(X_S)$ and $X_T = F_{pp}(X_T)$.

2) Unconstrained fuzzy feature fusion

For X_S and X_T obtained in procedure 1, we can use Algorithm 1 to obtain the corresponding unconstrained fuzzy feature matrixes, denoted by FFM_S and FFM_T.

3) Domain adaptation

After obtaining FFM_S and FFM_T, they still have different distributions. We therefore need to map FFM_S and FFM_T onto another feature space where both have the same distribution. This process can be regarded as homogeneous unsupervised domain adaptation.

We apply GFK to map FFM_S and FFM_T onto another feature space where they have the same distribution. There are five steps in GFK [6]:

- 1) standardize (z-score) FFM_S and FFM_T;

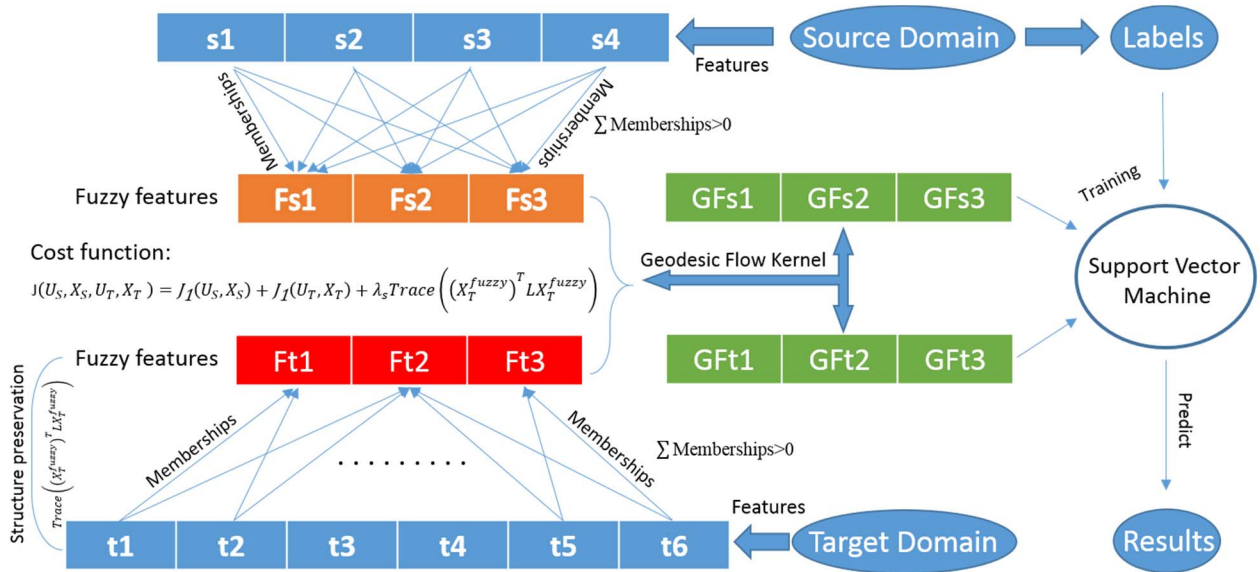


Fig. 1. Main steps of the proposed model. The source domain (normalized) comprises two parts: the feature space (s_1, \dots, s_4) and instance labels. The target domain comprises the feature space (t_1, \dots, t_6). Both feature spaces are used to construct fuzzy feature spaces ($Fs_1, Fs_2, Fs_3, Ft_1, Ft_2, Ft_3$), after which the fuzzy feature spaces are mapped by GFK ($GFs_1, GFs_2, GFs_3, GFt_1, GFt_2, GFt_3$). On the mapped space, knowledge can be transferred to the target domain.

2) calculate subspaces of FFM_S and FFM_T ;

3) use geodesic flow defined on Grassmann manifold to calculate kernel parameter G ;

4) calculate square root of G such that $R^T R = G$;

5) map FFM_S and FFM_T using R : $FFM_{GS} = R \times FFM_S$ and $FFM_{GT} = R \times FFM_T$ (in step 1 and step 2, FFM_S and FFM_T are replaced with corresponding calculation results).

4) Label Prediction

After adapting FFM_S and FFM_T to FFM_{GS} and FFM_{GT} , we can use FFM_{GS} to train a support vector machine (SVM) and predict labels of instances in the target domain using trained SVM and FFM_{TG} .

The source and target domains will first be normalized by step 1, and both feature spaces are separately used to construct fuzzy feature spaces. GFK is then applied to map the two fuzzy feature spaces onto a latent space. Finally, the model trained by the mapped fuzzy feature space can be used to predict labels of instances in the target domain.

IV. EXPERIMENTS

In this section, the UFFFG model is applied to two public datasets to test its classification performance. We first introduce the relevant datasets. We then introduce the related benchmarks. Lastly, the prediction result will be analyzed.

A. Datasets for HeUDA

We select the same datasets with our previous work, from the UCI Machine Learning Repository (UMLR, <http://archive.ics.uci.edu/ml/index.html>), to test the proposed model and benchmarks. Both datasets are related to personal credit assessment, namely German Credit Data and Australian Credit Approval. They have totally different

feature spaces. Our aim is to address two HeUDA tasks based on both datasets:

Task 1: G2A: Assume that the German data is labeled and the Australian data is unlabeled. Label “1” means “good credit” and label “-1” means “bad credit”. This task is equivalent to the question: “Can we use knowledge from German credit records to label unlabeled Australian data?”

Task 2: A2G: Assume that the Australian data is labeled and the German data is unlabeled. Label “1” means “good credit” and label “-1” means “bad credit”. This task is equivalent to the question: “Can we use knowledge from Australian credit records to label unlabeled German data?”

Table I shows the details of two HeUDA tasks. It should be clarified that the reason why the domain adaptation technique works in this instance (G2A or A2G) is because the two domains are similar (both are datasets for evaluating personal credit).

TABLE I. DETAILS OF THE TWO HEUDA TASKS

Source Domain	Target Domain	Labels	Task Name
German Credit Data	Australian Credit Approval	1: Good	G2A
Australian Credit Approval	German Credit Data	1: Good	A2G

B. Benchmarks and implementation details

To demonstrate the superiority of the proposed model, we select several benchmarks for comparison purposes. Dimensional reduction technology can be applied to force the two domains to have the same number of features. We denote this method as dimension reduction geodesic flow kernel (DG) where the dimensions of the mapped features are set as r . KCCA is claimed to be a HeUDA model in the literature and is selected as another benchmark model. The

third benchmark model is our previous model, FFF-GFK. DG, KCCA and FFF-GFK are regarded as models which have a knowledge transfer ability. In addition, we select two non-transfer models: all_1 (A1) and the clustering method (CM). A1 labels all instances as “1” and CM clusters the instances and randomly labels the categories using the k -means algorithm. Because there are no labeled instances in the target domain, CM naturally has extremely unstable prediction results.

Following [1], [4], [7], [8], SVM was trained on the unconstrained fuzzy features of the source data, then tested on the unlabeled target data. The order of features and instances in each dataset was randomly permuted before knowledge transfer. The German Credit dataset contains some bias, with 70% of the dataset labeled “1” and 30% labeled “-1”; however, the Australian Credit Approval dataset is unbiased. Given the basic assumption that both domains are similar, we needed to offset this dissimilarity by changing the implementation of the experiments with this dataset. Hence, we randomly selected 600 unbiased instances from the German Credit dataset for every experiment and ran the experiment 50 times for each model and each task.

Since the target domains do not contain any labeled data, it was impossible to automatically tune the optimal parameters for the target classifier using cross-validation. As a result, we used LIBSVM’s¹ default parameters for all classification tasks: the γ of the radial basis function kernel was set to $1/r$ (where r is the dimension of the feature space of the homogeneous representations); the cost C was set to 1; and the tolerance of the termination criterion was set to 0.001. The parameters for Algorithm 1, CS and DE are listed in Table II. Because there were no existing pairs for the two transfer learning tasks, we randomly matched instances from each domain as pairs for the KCCA model. Two datasets have different clustering performances (different structure), hence we give different λ_v and λ_s for two tasks. For task G2A, λ_v is 20 and λ_s is 0.05. For task A2G, λ_v is 5 and λ_s is 0.1.

Accuracy was used as the test metric as it has been widely adopted in the literature [4], [7], [27]. The definition follows.

$$Accuracy = \frac{|\mathbf{x}: \mathbf{x} \in \mathcal{X}_T \wedge g(\mathbf{x}) = y(\mathbf{x})|}{|\mathbf{x}: \mathbf{x} \in \mathcal{X}_T|},$$

where $y(\mathbf{x})$ is the ground truth label of \mathbf{x} , while $g(\mathbf{x})$ is the label predicted by the SVM classification algorithm. All experiments were conducted on an Intel(R) Core(TM) i7-4770 CPU at 3.40Ghz with a memory of 64 GB running Windows 7 professional 64-bit operating system and MATLAB 9.2.0. We ran the experiments 50 times, preprocessing the instances with the zscore function.

TABLE II. PARAMETER SETTINGS

Parameters	A1	CM	DG	FFF-GFK	UFFFG	CS	DE
r	-	-	10	10	10	-	-

¹ <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

ϵ	-	0.00001	-	0.00001	0.00001
Mutation factor	-	0.5			
Crossover rate	-	0.9			
Discovery rate			-	0.25	-
# of nest (population)				30	30
Max Iteration				1500	1500

C. Classification results and analysis

The classification results are listed in Table III. From this table, it is clear that UFFFG (with CS or DE) outperforms the other models. KCCA is better than DG, CM and A1 for task G2A. CM is better than A1, DG and KCCA for A2G, indicating that KCCA and DG are invalid for task A2G. FFF-GFK and UFFFG are better than the other models, mainly due to the lower probability of causing negative transfer. For example, KCCA has high Max accuracy when the task is G2A but it always has the situation of negative transfer, resulting in its average accuracy being much lower and having higher standard derivation. The UFFFG model has better performance than FFF-GFK for task G2A because the Australian data has better clustering results, indicating that regularizers, Reg , and $Trace((X_T^{fuzzy})^T L X_T^{fuzzy})$, have positive effects on improving classification performance.

TABLE III. PREDICTION RESULTS OF FFF-GFK AND BENCHMARKS

Model	Criteria	A2G	G2A
A1	Average Accuracy	50%	50%
	Min Accuracy	50%	50%
	Max Accuracy	50%	50%
CM	Average Accuracy	50.96%±5.22%	44.86%±0.43%
	Min Accuracy	43.67%	43.77%
	Max Accuracy	57.17%	56.23%
DG	Average Accuracy	50.91%±1.03%	45.17%±1.95%
	Min Accuracy	48.50%	41.16%
	Max Accuracy	53.00%	52.61%
KCCA	Average Accuracy	50.53%±4.61%	51.08%±9.78%
	Min Accuracy	41.00%	31.50%
	Max Accuracy	59.67%	71.00%
FFF-GFK	Average Accuracy	59.48%±2.91%	69.53%±6.90%
	Min Accuracy	50.33%	47.83%
	Max Accuracy	65.00%	77.50%
UFFFG with DE	Average Accuracy	59.73%±2.88%	72.11%±6.12%
	Min Accuracy	50.67%	56.00%
	Max Accuracy	65.00%	83.50%

		Accuracy	
UFFFG with CS	Average Accuracy	59.82%±2.88%	72.89%±6.20%
	Min Accuracy	50.67%	56.00%
	Max Accuracy	65.00%	83.83%

To show the negative transfer situation, we use the following criterion [10] to show the situations of negative transfer of each model.

$$P_{ng} = \frac{1}{T} \text{Card}(\{i | a_i \leq 0.5, i = 1, \dots, T\}),$$

where T is the number of times the experiments are run ($T = 50$ in this paper) and a_i is the accuracy of the i^{th} experiment. Lower P_{ng} values represent the higher probability of avoiding negative transfer. Table IV shows the values of P_{ng} in each model. The P_{ng} values of these models clearly show that the UFFFG model is better than the other models. In particular, UFFFG is better than FFF-GFK for task G2A (in 50 experiments, the proposed model completely avoids negative transfer).

TABLE IV. P_{ng} VALUES OF EACH MODEL

Model	A2G	G2A
A1	100%	100%
CM	22.00%	94.00%
DG	16.00%	98.00%
KCCA	42.00%	46.00%
FFF-GFK	0.00%	2.00%
UFFFG	0.00%	0.00%

Table III also shows that both the CS and DE algorithms effectively optimize the cost function J and obtain similar classification results. The following table shows the running time of the benchmarks and UFFFG (optimized by CS and DE). It is clear that CS can optimize the cost function J faster than DE.

TABLE V. RUNNING TIME OF EACH MODEL

	A1	CM	DG	FFF-GFK	UFFFG+CS	UFFFG+DE
Time (s)	<1	<1	1.8	1.5	344	489

V. CONCLUSIONS AND FURTHER STUDY

This paper presents a novel heterogeneous unsupervised domain adaptation model, called the unconstrained fuzzy feature fusion geodesic flow kernel. We first construct the unconstrained fuzzy features and, through these constructed fuzzy features, knowledge from the heterogeneous source domain is transferred to an unlabeled target domain. Different from previous work, we relax the condition that memberships must sum to one and retain the original information contained in the target domain when transferring knowledge across domains. The proposed model is tested on two public datasets and delivers satisfactory classification results.

Based on the proposed model, our future work includes: 1) determining how to automatically select the dimension of the mapped feature space, and 2) investigating a more effective algorithm to optimize cost function J . The first aspect aims to make UFFFG widely used in many datasets without dimension selection. The second aspect will provide a way to use the proposed model on a very large dataset.

ACKNOWLEDGMENT

The work presented in this paper was supported by the Australian Research Council (ARC) under Discovery Grant DP170101632.

REFERENCES

- [1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [2] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowledge-Based Syst.*, vol. 80, pp. 14–23, 2015.
- [3] T. Liu, Q. Yang, and D. Tao, "Understanding how feature structure transfers in transfer learning," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 2365–2371.
- [4] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.
- [5] J. Jiang and C. Zhai, "A two-stage approach to domain adaptation for statistical classifiers," in *Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management*, 2007, pp. 401–410.
- [6] B. Gong, K. Grauman, and F. Sha, "Learning kernels for unsupervised domain adaptation with applications to visual object recognition," *Int. J. Comput. Vis.*, vol. 109, no. 1–2, pp. 3–27, 2014.
- [7] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1134–1148, 2014.
- [8] M. Xiao and Y. Guo, "Feature space independent semi-supervised domain adaptation via kernel matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 54–66, 2015.
- [9] Y. R. Yeh, C. H. Huang, and Y. C. F. Wang, "Heterogeneous domain adaptation and classification by exploiting the correlation subspace," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2009–2018, 2014.
- [10] F. Liu, G. Zhang, and J. Lu, "Heterogeneous unsupervised domain adaptation based on fuzzy feature fusion," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, 2017, pp. 1–6.
- [11] Y. Shi, L. Angeles, and F. Sha, "Information-theoretical learning of discriminative clusters for unsupervised domain adaptation," in *Proceedings of the 29th International Conference on Machine Learning*, 2012, pp. 1079–1086.
- [12] Y. Aytar and A. Zisserman, "Tabula Rasa: Model Transfer for Object Category Detection," in *Proceedings of the 13th International Conference on Computer Vision*, 2011, pp. 2252–2259.
- [13] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video

- concept detection using adaptive SVMs,” in *Proceedings of the 15th ACM International Conference on Multimedia*, 2007, pp. 188–197.
- [14] J. Xu, S. Ramos, D. Vazquez, and A. M. Lopez, “Domain adaptation of deformable part-based models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2367–2380, 2014.
- [15] J. S. Muller, P. von Bunau, F. C. Meinecke, F. J. Kiraly, and K. R. Muller, “The stationary subspace analysis toolbox,” *J. Mach. Learn. Res.*, vol. 12, pp. 3065–3069, 2011.
- [16] B. Kulis, K. Saenko, and T. Darrell, “What you saw is not what you get: Domain adaptation using asymmetric kernel transforms,” in *Proceedings of the 24th IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1785–1792.
- [17] T. Kanamori, S. Hido, and M. Sugiyama, “A least-squares approach to direct importance estimation,” *J. Mach. Learn. Res.*, vol. 10, pp. 1391–1445, 2009.
- [18] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, “Fuzzy regression transfer learning in Takagi-Sugeno fuzzy models,” *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1795–1807, 2017.
- [19] V. Behbood, J. Lu, G. Zhang, and W. Pedrycz, “Multistep fuzzy bridged refinement domain adaptation algorithm and its application to bank failure prediction,” *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 6, pp. 1917–1935, 2015.
- [20] H. Zuo, G. Zhang, V. Behbood, J. Lu, W. Pedrycz, and T. Zhang, “Fuzzy transfer learning in data-shortage and rapidly changing environments,” in *Proceedings of the 12th International Fuzzy Logic and Intelligent Technologies in Nuclear Science Conference*, Belfast, UK: World Scientific, 2016, pp. 175–180.
- [21] X. Shi, Q. Liu, W. Fan, and P. S. Yu, “Transfer across completely different feature spaces via spectral embedding,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 906–918, 2013.
- [22] C. Wang and S. Mahadevan, “Heterogeneous domain adaptation using manifold alignment,” in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 2011, pp. 1541–1546.
- [23] B. Tan, Y. Zhang, S. J. Pan, and Q. Yang, “Distant Domain Transfer Learning,” in *Proceedings of the 31st AAAI conference on Artificial Intelligence*, 2017, pp. 2604–2610.
- [24] H. V. Nguyen, H. T. Ho, S. Member, and V. M. Patel, “DASH-N: Joint hierarchical domain adaptation and feature learning,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5479–5491, 2015.
- [25] M. Long, J. Wang, Y. Cao, J. Sun, and P. S. Yu, “Deep learning of transferable representation for scalable domain adaptation,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2027–2040, 2016.
- [26] N. Courty, R. Flamary, D. Tuia, S. Member, and A. Rakotomamonjy, “Optimal transport for domain adaptation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1853–1865, 2017.
- [27] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, “Scatter component analysis: A unified framework for domain adaptation and domain generalization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1414–1430, 2017.
- [28] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and I. Systems, “Domain adaptation with conditional transferable components,” in *Proceedings of the 33rd International Conference on Machine Learning*, 2016, pp. 2839–2848.
- [29] B. Sun, J. Feng, and K. Saenko, “Return of frustratingly easy domain adaptation,” in *Proceedings of the 30th AAAI conference on Artificial Intelligence*, 2016, pp. 2058–2065.
- [30] W. Deng, L. Zheng, G. Kang, Y. Yang, Q. Ye, and J. Jiao, “Image-Image Domain Adaptation with Preserved Self-Similarity and Domain-Dissimilarity for Person Re-identification,” *arXiv Prepr. arXiv 1711.07027 [cs.CV]*, pp. 1–10, 2017.
- [31] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, “Camera Style Adaptation for Person Re-identification,” *arXiv Prepr. arXiv 1711.10295 [cs.CV]*, pp. 1–10, 2017.
- [32] Q. Wu, Z. Wang, F. Deng, Z. Chi, and D. D. Feng, “Realistic human action recognition with multimodal feature selection and fusion,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 43, no. 4, pp. 875–885, 2013.
- [33] R. K. Ando and T. Zhang, “Learning on Graph with Laplacian Regularization,” in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems*, 2006, pp. 25–32.