Transfer Learning by Sample Selection Bias Correction and Its Application in Communication Specific Emitter Identification

Zhen Liu, Jun-an Yang, Hui Liu, and Wei Wang
Electronic Engineering Institute, Hefei 230037, China
Key Laboratory of Electronic Restriction of Anhui Province, Hefei 230037, China
Email: {ahulz, liuhui1983eei}@163.com; yangjunan@ustc.edu.cn; wwei009@mail.ustc.edu.cn

Abstract —In many traditional machine learning algorithms, a major assumption is that the training samples and the test samples have the same distribution. However, this assumption does not hold in many real applications. In recent years, transfer learning has attracted a significant amount of attention to solve this problem. In this paper, a novel transfer learning method based on clustering analysis and re-sampling is proposed, which can correct different types of domain differences and does not need to estimate the different distribution directly. The method explores the data structure by clustering analysis, and then uses the obtained structure information to generate a new training set for target learning under a re-sampling strategy. To explore more data structure information and be more robust to data sets with various shapes and densities, the method introduces the fuzzy neighborhood membership degree to improve the performance of clustering analysis. It also applies the Gaussian kernel function to measure the similarities between samples to improve the reliability of the new training samples. The proposed method can transfer more useful knowledge from the source domain to the target domain. Experimental results on toy datasets demonstrate that the proposed method can effectively and stably enhance the learning performance. Finally, the proposed algorithm is applied to the communication specific emitter identification task and the result is also satisfying.

Index Terms—Transfer learning, clustering analysis, density-based clustering, fuzzy neighborhood, communication specific emitter identification

I. Introduction

Machine learning technologies have already achieved a significant success in many knowledge engineering areas and computational fields, most of these technologies work under the two common assumptions: 1) the training data (source domain) and the test data (target domain) are drawn from the identical feature space with underlying distribution; 2) there are enough available samples building a reliable classification model. However, in many real-world applications, these assumptions do not hold, which means that the joint distribution $P(\boldsymbol{x},y)$ of the data $\boldsymbol{x}$ and its corresponding label y is different, i.e., the source domain distribution $P_S(\boldsymbol{x},y)$ is not equal to $P_T(\boldsymbol{x},y)$ in the target domain. Besides, collecting enough homogeneous labeled data is very expensive and sometimes not practically possible. In such cases, transfer learning [1]-[4] is truly beneficial, which can effectively exploit and transfer the knowledge for target domain learning from similar but not identical labeled source domain data. Recently, transfer learning has been applied in many real world applications, such as text processing [5]-[9], image processing [10]-[13], indoor localization [14], network identification [15], and automatic control [16], etc.

In terms of "what to transfer", transfer learning can be categorized in three settings: instance transfer learning, feature transfer learning, and parameter transfer learning. Due to the different distribution between the source domain and the target domain, some labeled samples in the source domain may not be useful for the target task. How to select the 'right' samples for the target task is the issue that instance transfer learning mainly solves. Based on the empirical risk minimization, Zadrozny [17] used the estimated $P_T(\boldsymbol{x},y)/P_S(\boldsymbol{x},y)$ to measure the importance of the samples to the target task and proposed to estimate the terms $P_T(\boldsymbol{x})$ and $P_S(\boldsymbol{x})$ independently by constructing simple classification problems after assuming $P_T(y|\boldsymbol{x})=P_S(y|\boldsymbol{x})$. Huang *et al*. [18] proposed a kernel mean matching algorithm to learn $P_T(\boldsymbol{x})/P_S(\boldsymbol{x})$ directly by matching the means of data between the source domain and the target domain in a reproducing kernel Hilbert space. Sugiyama *et al*. [19] proposed an algorithm known as Kullback-Leibler Importance Estimation Procedure to estimate $P_T(\boldsymbol{x})/P_S(\boldsymbol{x})$ based on the minimization of the Kullback-Leibler divergence.

However, it is often neither the case that the assumption of $P_T(y|\boldsymbol{x})=P_S(y|\boldsymbol{x})$ is always appropriate, nor easy to estimate $P_T(\boldsymbol{x})/P_S(\boldsymbol{x})$ accurately. Motivated from the Bias Reduction via Structure Discovery (BRSD) [20] and fuzzy neighborhood membership degree [21], we propose a new transfer learning method called TL-FNDCR (Transfer Learning based on Fuzzy

Manuscript received November 6, 2015; revised April 15, 2016.
This work was supported by the National High-tech R&D Program (863 Program) of China, and Anhui Provincial Natural Science Foundation (NO.1308085QF99, NO.1408085MKL46).
Corresponding author email: ahulz@163.com.
doi:10.12720/jcm.11.4.417-427

Neighborhood Density Clustering and Re-sampling) which does not need to estimate the $P_T(\boldsymbol{x},y)/P_S(\boldsymbol{x},y)$ and can correct different types of domain bias. The method extracts the information about the data structure by clustering analysis, and then uses the information to generate a new training set for the target learning under a re-sampling strategy. By introducing a fuzzy neighborhood membership degree into the clustering analysis, TL-FNDCR can explore more data structure information and be more robust to datasets with various shapes and densities. To further improve the reliability of the new training samples, it also applies the Gaussian kernel function to measure the similarities between samples. The proposed method can transfer more useful knowledge from the source domain to the target domain. Validation of the proposed method is performed with toy datasets. Results demonstrate that the proposed method can more effectively and stably enhance the learning performance. Finally, the proposed algorithm is applied to the specific emitter identification task and the result is also satisfying.

The rest of the paper is organized as follows: In section II, the overviews of the transfer learning framework based on clustering analysis and re-sampling are reviewed; In section III, a novel transfer learning method called TL-FNDCR is proposed; In section IV, extensive experiments are performed; Some conclusions and possible future work are given in section V.

II. Preliminary

Let us represent the samples from the source domain as $\boldsymbol{X}_S=\{(\boldsymbol{x}_1^S,y_1^S),(\boldsymbol{x}_2^S,y_2^S),\cdots,(\boldsymbol{x}_{N_S}^S,y_{N_S}^S)\}$, where $\boldsymbol{x}_i^S$ is the *i*-th feature vector in the source domain, y_i^S is the corresponding label. N_S is the number of source domain data. $\boldsymbol{X}_T=\{\boldsymbol{x}_1^T,\boldsymbol{x}_2^T,\cdots,\boldsymbol{x}_{N_T}^T\}$ denotes target domain dataset, where $\boldsymbol{x}_i^T$ is the *i*-th feature vector in the target domain, N_T is the number of the target domain data. The whole dataset is $\boldsymbol{X}=\boldsymbol{X}_S\cup\boldsymbol{X}_T$, and the sample number is $N=N_S+N_T$. In the problem setting, a lot of labeled data in the source domain is available, but there are only unlabeled data in the target domain. Formally, the joint distribution $P(\boldsymbol{x},y)$ of feature vector $\boldsymbol{x}$ with its label y is different, i.e., $P_S(\boldsymbol{x},y)\neq P_T(\boldsymbol{x},y)$.

Since the training data is hardly consistent with the distribution of the test data in the practical application, the model learned on the biased training set may be not reliable in the target domain. The transfer learning algorithm based on clustering analysis and re-sampling can generate an unbiased training dataset for target learning [20]. Without loss of generality, binary classification is taken into account. In Fig. 1, the solid points denote labeled training data and the blank points denote unlabeled test data, and the squares and the circles represent one class respectively. Fig. 1 (a) shows that the classification model learned on the initial biased training data is not reliable in the target domain (test dataset). In Fig. 1 (b), the data structure is explored by clustering analysis. Then, a new training dataset is generated based on these clusters under a certain strategy shown in Fig. 1 (c). In the end, a much more reliable model is learned, as shown in Fig. 1 (d).

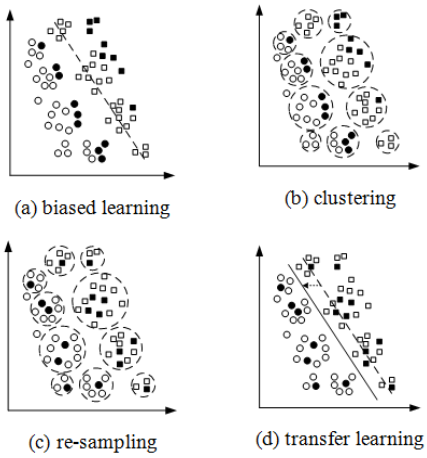


Fig. 1. Intuition of transfer learning based on clustering and re-sampling

In Fig. 1(c), the samples inside each cluster are very similar in their distribution if there are a few samples in each cluster. Here, we assume a selection variable t and the event $t=1$ denotes that a labeled sample $(\boldsymbol{x},y)$ is selected into the training set. Then, we have two probability variables: $P(t=1|\boldsymbol{x},y)$ is the probability of the event $t=1$ happening on $(\boldsymbol{x},y)$, $P(t=1)$ is the probability of the event $t=1$ on the whole data set. After clustering on $\boldsymbol{X}$, N_C clusters are obtained, i.e., $\bigcup_{i=1}^{N_C} C_i=\boldsymbol{X}$ and $C_i\bigcap_{i\neq j}C_j=\varnothing$. If we select samples evenly from each cluster under the same percentage P_1, then we have $P(t=1|C_i,y)=P(t=1)=P_1$. Then,

$$\begin{aligned}P(t=1,\boldsymbol{X},y)&=\sum_{i=1}^{N_C}P(t=1,C_i,y)\\&=\sum_{i=1}^{N_C}P(t=1|C_i,y)P(C_i,y)\\&=\sum_{i=1}^{N_C}P(t=1)\times P(C_i,y)\\&=P(t=1)\times P(\boldsymbol{X},y)\end{aligned}\tag{1}$$

Consequently, $P(t=1|\boldsymbol{X},y)=\dfrac{P(t=1,\boldsymbol{X},y)}{P(\boldsymbol{X},y)}=P(t=1)=P_1$. In other words, the selection variable t is a random variable completely independent from both the feature vectors $\boldsymbol{X}$ and the true class label y. Thus, intuitively, we can conclude that the data sets sampled evenly from each of these clusters should not have any samples of selection bias. Then, the obtained new training dataset is consistent with the distribution of the whole dataset.

III. METHODOLOGY

A. Data Structure Discovery by Fuzzy Neighborhood Density based Clustering

In the framework of transfer learning based on clustering and re-sampling as introduced in section II, a clustering algorithm is run on the total dataset $\boldsymbol{X}$ to explore the intrinsic structure of the data. Any clustering method could be used as long as it is appropriate for exploring the intrinsic structure of the data. The density based clustering algorithm called DBSCAN (Density Based Spatial Clustering of Application with Noise) is widely used as it can find the oddly-shaped cluster without knowing the number of clusters. The main idea of DBSCAN is: starting from a certain core point and expanding the density-reachable areas until maximization, the resulting region is a new cluster where there are only core points and border points and these points are density-connected with each other. How to find the core points in the whole dataset is crucial to the performance of DBSCAN. In DBSCAN, a data $\boldsymbol{x}_i$ is called a core point if

$$Card(\boldsymbol{x}_i) = \sum_j M(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq MinPts \tag{2}$$

is satisfied. $Card(\boldsymbol{x}_i)$ is the cardinality of the neighborhood set of $\boldsymbol{x}_i$ and *MinPts* is a threshold. *MinPts* can be evaluated based on the mean of $Card(\boldsymbol{x})$ in the whole dataset. Significantly, $M(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the membership degree of $\boldsymbol{x}_j$ to the neighborhood set of $\boldsymbol{x}_i$.

$$M(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} 1 & d(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq Eps \\ 0 & otherwise \end{cases} \tag{3}$$

where $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ represents the distance between point $\boldsymbol{x}_i$ and point $\boldsymbol{x}_j$, *Eps* is the neighborhood radius which can be estimated through labeled samples [20]. If $\boldsymbol{x}_j$ is in the neighborhood set of $\boldsymbol{x}_i$, $M(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1$; otherwise 0.

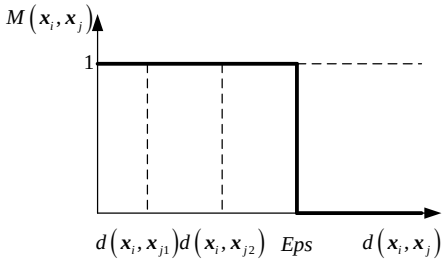
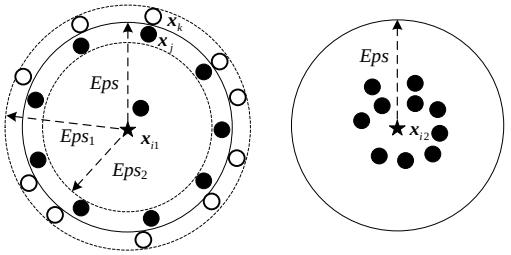


Fig. 2. Crisp neighborhood membership degree

The neighborhood relation determined above is called crisp neighborhood membership degree, which means that $M(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is equal to 1 or 0 only depending on whether $\boldsymbol{x}_j$ is in the *Eps*-neighborhood of $\boldsymbol{x}_i$ or not (Fig. 2). According to the above definitions, we can see the drawbacks of the crisp neighborhood membership degree in (3):

1) It cannot distinguish the difference between the samples in a certain neighborhood set. If regarding $M(\boldsymbol{x}_i, \boldsymbol{x}_j)$ as $\boldsymbol{x}_j$'s contribution to prompting $\boldsymbol{x}_i$ being a core point, we can see that all the points in the *Eps*-neighborhood make the same contribution without considering the difference of their distances to $\boldsymbol{x}_i$. However, this setting may not be appropriate. Intuitively, the closer it comes between $\boldsymbol{x}_j$ and $\boldsymbol{x}_i$, the more contribution of $\boldsymbol{x}_j$ to $\boldsymbol{x}_i$ it should be.

2) It cannot show the detail information of the neighborhood structure. Fig. 3 shows two central points $\boldsymbol{x}_{i1}$ and $\boldsymbol{x}_{i2}$ which have the same number of neighbors within *Eps* radius but different neighborhood distribution. The solid line circle represents the border of *Eps*-neighborhood. The cardinalities of the neighborhood set of $\boldsymbol{x}_{i1}$ and $\boldsymbol{x}_{i2}$ are the same according to (2)-(3). However, it is obvious that their neighborhood structures are quite different. The crisp neighborhood membership degree cannot reflect this difference which means that some useful data structure information is neglected. And this information may be useful for learning.



(a) point $\boldsymbol{x}_{i1}$ (b) point $\boldsymbol{x}_{i2}$

Fig. 3. Neighborhood membership degree of different data distribution

3) It has poor robustness to the parameter Eps and bad adaptability to the data with various shapes and densities. Additionally, let us investigate points $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ in Fig. 3 (a) where $\boldsymbol{x}_j$ is in the neighborhood of $\boldsymbol{x}_{i1}$ but $\boldsymbol{x}_k$ is outside. Even though the two points are both close to the border, their neighborhood membership degree to $\boldsymbol{x}_{i1}$ is completely different according to (3), i.e., $M(\boldsymbol{x}_{i1}, \boldsymbol{x}_j) = 1$ but $M(\boldsymbol{x}_{i1}, \boldsymbol{x}_k) = 0$. A little change in *Eps* may lead to very different consequences. For example, taking Eps_1 instead of *Eps*, the $Card(\boldsymbol{x}_{i1})$ will increase dramatically from 10 to 20. On the contrary, if *Eps* is replaced by Eps_2, the $Card(\boldsymbol{x}_{i1})$ will decrease from 10 to 1, which may make $\boldsymbol{x}_{i1}$ from a core point to a non-core point. In other words, the crisp neighborhood membership degree is over sensitive to the parameter *Eps*, which makes the whole learning process badly adapt to the data with various shapes and densities. As the optimal *Eps* is difficult to obtain in the practical application, we can not guarantee the good learning performance.

As the performance of the clustering algorithm is significant for data structure exploration, the proposed

TL-FNDCR can overcome the drawbacks of the crisp neighborhood membership degree and explore more useful information for target learning by using a fuzzy neighborhood membership degree [21]. Equation (4) gives a linear type of fuzzy neighborhood membership degree.

$$M(\boldsymbol{x}_i,\boldsymbol{x}_j)=\begin{cases}1-\beta\dfrac{d(\boldsymbol{x}_i,\boldsymbol{x}_j)}{Eps} & d(\boldsymbol{x}_i,\boldsymbol{x}_j)\le Eps\\ 0 & otherwise\end{cases} \tag{4}$$

where the parameter β ($\beta>0$) is used to control the sensitivity of membership degree to distance. In order to set the membership degree value $M(\boldsymbol{x}_i,\boldsymbol{x}_j)$ in the range [0, 1], β is given as follows.

$$\beta=(1-m_0) \tag{5}$$

where $m_0\in[0,1)$ is a certain value, representing the neighborhood membership degree of the points just on the *Eps*-neighborhood border.

In (4) and (5), we can observe the following properties of the linear fuzzy neighborhood membership degree:

1) In the neighborhood, the neighborhood membership degree of points with different distance would be extremely different. If the point is closer to the center point, its membership degree is larger. The fuzzy neighborhood can accurately distinguish the difference of data in the neighborhood set.

2) According to (4), $Card(\boldsymbol{x}_{i2})$ becomes larger than $Card(\boldsymbol{x}_{i1})$, rather than equal (Fig. 3). In other words, the linear fuzzy membership degree could carry more information about the data neighborhood structure.

3) If m_0 approaches 0, the membership degree of the points close to the neighborhood border all tend to 0 no matter whether the points are in the neighborhood set or not (Fig. 4). Then, the density of the central point will not change dramatically along with the small change of *Eps* (Fig. 3(a)). Fuzzy neighborhood membership degree could improve the poor robustness of the crisp case.

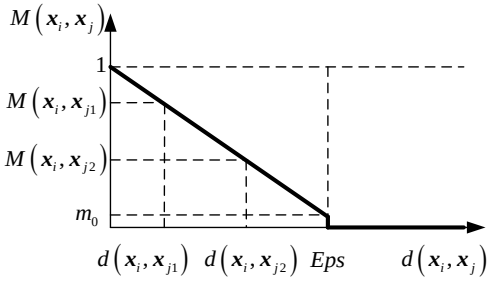


Fig. 4. Linear fuzzy neighborhood membership degree

Considering the possible nonlinear relationship in the feature space, the linear fuzzy neighborhood membership degree will not be very appropriate. Thus, another important fuzzy neighborhood membership degree called exponential fuzzy neighborhood membership degree is given below.

$$M(\boldsymbol{x}_i,\boldsymbol{x}_j)=\begin{cases}\exp\left\{-\left(\beta\dfrac{d(\boldsymbol{x}_i,\boldsymbol{x}_j)}{Eps}\right)^2\right\} & d(\boldsymbol{x}_i\ \boldsymbol{x}_j)\le Eps\\ 0 & otherwise\end{cases} \tag{6}$$

where β is given as follows

$$\beta=\sqrt{-\ln(m_0)} \tag{7}$$

where $m_0\in(0,1)$ is also the neighborhood membership degree of the points just on the *Eps*-neighborhood border.

Similar to the radial basis function, the exponential fuzzy neighborhood membership degree can effectively deal with high-dimensional nonlinear data owing to its ability of mapping the data into infinite dimensional space. Fig. 5 shows the exponential fuzzy neighborhood membership degree.

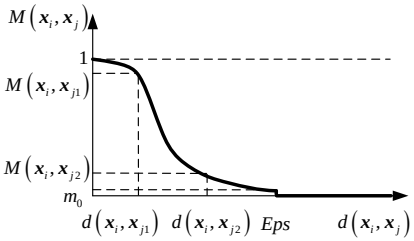


Fig. 5. Exponential fuzzy neighborhood membership degree

When m_0=1 , the fuzzy neighborhood membership degrees defined in (4) and (6) are degenerated to (3). The fuzzy neighborhood membership degree is the generalization of the crisp case and the latter is just a special case of the former.

B. Re-Sampling by Considering Label Reliability and Data Representativeness

After the clustering analysis on the whole dataset $\boldsymbol{X}$, we obtain N_C clusters: $\{C_1,C_2,\cdots,C_{N_C}\}$. The number of samples in C_k is N_{C_k} . In the step of re-sampling, a new training dataset for the target learning is evenly sampled from every cluster under the same proportion. The proportion is usually taken as the ratio of the number of samples in the source domain to that in the whole dataset, i.e., N_S/N . The re-sampling strategy considers the label reliability and representativeness comprehensively.

Firstly, the label reliability of the sample $\boldsymbol{x}_i$ is defined as

$$Rl(\boldsymbol{x}_i)=\arg\max_{\boldsymbol{x}_j^S\in\boldsymbol{X}_S}\exp\left(-\frac{1}{4\sigma}\left\|\boldsymbol{x}_i-\boldsymbol{x}_j^S\right\|^2\right) \tag{8}$$

where parameter σ controls the radial range of the Gaussian kernel function and is generally fixed as the mean distance between samples in the whole data set. If $\boldsymbol{x}_i$ is labeled, it has the highest label reliability (i.e., $Rl(\boldsymbol{x}_i)=1$) as its nearest labeled neighbor is itself, e.g., the labeled source domain samples $\boldsymbol{x}_1^S$, $\boldsymbol{x}_2^S$, $\boldsymbol{x}_3^S$ and $\boldsymbol{x}_4^S$ in Fig. 6. This is reasonable because they already have

the correct class labels. If x_i is unlabeled, it is given the same label as its nearest labeled neighbor's, e.g., the unlabeled target domain samples x_1^T and x_2^T have the same label with x_1^S 's and x_4^S 's respectively in Fig. 6.

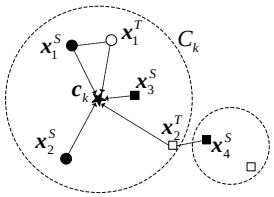


Fig. 6. Re-sampling strategy for selecting new training set

In the *k*th cluster C_k, a certain number of samples (more than $N_{C_k} N_S / N$) with highest label reliability are selected. Then, the $N_{C_k} N_S / N$ samples with highest data representativeness are chosen from them. The data representativeness $Rp(x_i)$ is defined as

$$Rp(x_i) = \exp\left(-\frac{1}{4\sigma}\|x_i - c_k\|^2\right) \quad (9)$$

where $c_k = \frac{1}{N_{C_k}} \sum_{l=1}^{N_{C_k}} x_l$ is the center of the cluster C_k. The parameter σ is also fixed as the mean distance between samples in the whole dataset. As shown in Fig. 6, x_3^S has the highest data representativeness in C_k because it is most close to the cluster center and can best represent the cluster.

Finally, the $N_{C_k} N_S / N$ selected samples are added into the new training set X_A. After the re-sampling processing is conducted in every cluster, the new training set X_A is formed.

In (8)-(9), $\exp\left(-\frac{1}{4\sigma}\|x_i - c_k\|^2\right)$ is essentially a kind of similarity metric, which is also similar to the Gaussian kernel function. It is well known that Gaussian kernel function has large coverage and can deal with the nonlinear high-dimensional data. Thus, we can expect a high reliability of the new training samples according $Rl(x_i)$ and $Rp(x_i)$ even when the feature space is nonlinear and high dimensional.

The flow chart of TL-FNDCR is summarized in the Algorithm 1.

Algorithm 1 A summary of TL-FNDCR algorithm

Input: The labeled source domain data set X_S and unlabeled target domain data set X_T.

Initial: Initialize the new training data set $X_A = \varnothing$.

Structure discovery: Clustering analysis on $X = X_S \cup X_T$ by the fuzzy neighborhood density clustering with (4) or (6), then N_C clusters $\{C_1, C_2, \cdots, C_{N_C}\}$ are obtained.

Re-sampling: For each $C_i \in \{C_1, C_2, \cdots, C_{N_C}\}$, select a subset X_{A_i} from C_i under the re-sampling strategy. Then, add X_{A_i} into X_A.

Output: The new training dataset X_A for target domain learning.

IV. Experiments and Discussions

A. Performance Comparisons on UCI Data Sets

In this section, a set of experiments on UCI datasets is provided to evaluate the performance of the proposed algorithm. Fig. 7 shows the flow chart of data processing with TL-FNDCR. In order to demonstrate the "goodness" of the proposed method with different base classifiers, five different base classifiers are used respectively: C4.5, Naive Bayes, NNge, Logistic Regression, and SVM. For short, we call TL-FNDCR with linear fuzzy neighborhood membership degree and exponential fuzzy neighborhood membership degree as TL-Linear and TL-Exp respectively.

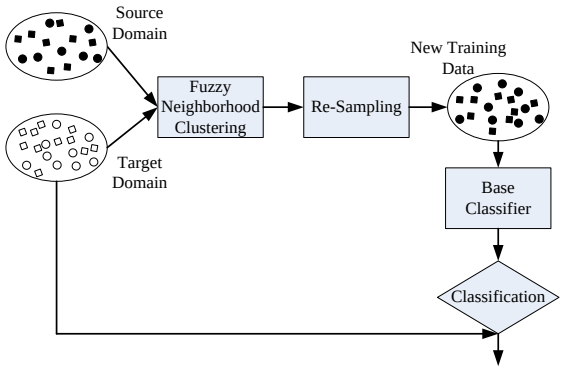


Fig. 7. The flow chart of data processing with TL-FNDCR

Fourteen UCI datasets are used in the experiments. To use a dataset for the purpose of transfer learning, we need to create two sets, used as the source and target domains respectively. Here, bootstrap sampling [22] is applied to randomly select the two datasets from the whole dataset. Information on these data sets is tabulated in Table I, where the Source/Target represents the number of labeled source domain data against that of unlabeled target domain data.

TABEL I: EXPERIMENTAL UCI DATA SETS

Dataset	Attribute	Class	Source/Target
SatImage	37	6	180/6255
Iris	4	3	20/130
Letter	17	26	300/19700
Wine	13	3	15/163
ColonTumor	2000	2	15/47
Diabetes	8	2	80/688
Glass	9	7	20/194
Haberman	3	2	30/276
Mfeat	649	10	60/1940
Wdbc	30	2	30/539
Sonar	60	2	20/188
Spambase	57	2	25/4580
Vehicle	18	4	84/762
WaveForm	40	3	75/4925

The classification accuracy of the proposed methods compared with the Baseline and BRSD under five different base classifiers is recorded in Table II to Table VI respectively. The Baseline means that using labeled source domain samples as training data for target domain learning directly without any knowledge transfer processing. BRSD [20] is a learning method under the

same framework in section II, which uses DBSCAN for clustering analysis and Manhattan distance to measure the similarities between samples. Regardless of the base classifiers, there is only one parameter need to be fixed empirically in our method (i.e., m_0) and no parameter in Baseline and BRSD. The m_0 parameter is empirically fixed as 0 and 0.01 for TL-Linear and TL-Exp respectively. The parameters in every base classifier are set the default values as in Weka [23]. For each dataset, the highest accuracy is highlighted in bold.

In Table II to Table VI, TL-FNDCR (both TL-Linear and TL-Exp) can effectively improve classification accuracy in most cases. The performance improvement owes the better data structure exploration of TL-FNDCR. As the Baseline method uses source domain data directly without considering the difference between domains, its classification results are always not good. Moreover, the accuracy of TL-Exp appears higher than that of TL-Linear. If the classification results of each algorithm are averaged across all the data sets and base classifiers, it can be found that the average improvement of TL-Exp is as high as 9.48% compared with BRSD, while TL-Linear also has an improvement of 6.43%. It proves that TL-FNDCR can effectively improve the learning performance.

TABLE II: CLASSIFICATION ACCURACY WITH C4.5

Dataset	Baseline	BRSD	TL-Linear	TL-Exp
SatImage	0.44	0.67	**0.79**	**0.79**
iris	0.61	0.83	0.94	**0.95**
Letter	0.08	0.50	0.51	**0.54**
Wine	0.30	0.81	**0.88**	**0.88**
ColonTumor	0.40	0.57	0.57	**0.72**
Diabetes	0.66	0.71	0.76	**0.77**
Glass	0.47	0.58	**0.60**	**0.60**
Haberman	0.20	0.77	**0.80**	0.79
Mfeat	0.17	0.68	**0.82**	**0.82**
Wdbc	0.90	0.82	**0.91**	**0.91**
Sonar	0.46	0.60	0.69	**0.70**
Spambase	0.78	0.80	**0.83**	**0.85**
Vehicle	0.31	0.48	0.60	**0.65**
WaveForm	0.55	0.69	**0.74**	0.73

TABLE III: CLASSIFICATION ACCURACY WITH NAIVE BAYES

Dataset	Baseline	BRSD	TL-Linear	TL-Exp
SatImage	0.33	0.71	**0.8**	**0.8**
iris	0.61	**0.95**	0.92	0.92
Letter	0.06	0.54	**0.55**	**0.55**
Wine	0.22	**0.95**	0.92	0.91
ColonTumor	0.57	0.57	0.57	**0.74**
Diabetes	0.70	0.69	0.74	**0.75**
Glass	0.10	0.45	**0.52**	**0.52**
Haberman	0.20	0.77	**0.81**	**0.81**
Mfeat	0.08	**0.69**	0.67	0.67
Wdbc	0.91	0.91	0.91	**0.92**
Sonar	0.50	0.69	0.72	**0.75**
Spambase	0.85	0.84	**0.87**	**0.87**
Vehicle	0.29	**0.51**	0.50	0.49
WaveForm	0.59	0.80	**0.82**	**0.82**

TABLE IV: CLASSIFICATION ACCURACY WITH NNge

Dataset	Baseline	BRSD	TL-Linear	TL-Exp
SatImage	0.42	0.68	**0.81**	**0.81**
iris	0.62	**0.95**	0.94	0.94
Letter	0.09	0.55	0.59	**0.62**
Wine	0.53	**0.90**	0.87	0.87
ColonTumor	0.64	0.62	0.62	**0.83**
Diabetes	0.68	0.74	**0.77**	**0.77**
Glass	0.08	0.41	**0.51**	**0.51**
Haberman	0.20	0.64	**0.77**	**0.77**
Mfeat	0.12	**0.70**	**0.70**	**0.70**
Wdbc	0.91	0.91	0.92	**0.93**
Sonar	0.50	0.54	0.60	**0.69**
Spambase	0.78	0.72	**0.85**	0.83
Vehicle	0.30	0.44	0.53	**0.55**
WaveForm	0.50	0.76	**0.80**	**0.80**

TABLE V: CLASSIFICATION ACCURACY WITH LOGISTIC REGRESSION

Dataset	Baseline	BRSD	TL-Linear	TL-Exp
SatImage	0.45	0.72	**0.73**	**0.73**
iris	0.62	**0.96**	0.95	0.95
Letter	0.12	**0.61**	0.54	0.58
Wine	0.38	0.83	**0.96**	**0.96**
ColonTumor	0.55	0.57	0.60	**0.83**
Diabetes	0.7	0.73	0.76	**0.77**
Glass	0.14	0.47	**0.60**	**0.60**
Haberman	0.2	0.77	**0.80**	**0.80**
Mfeat	0.33	0.87	**0.89**	**0.89**
Wdbc	0.87	0.88	0.93	**0.95**
Sonar	0.46	0.65	**0.75**	**0.75**
Spambase	0.80	**0.85**	0.80	0.81
Vehicle	0.42	0.63	**0.69**	**0.69**
WaveForm	0.59	**0.85**	0.75	0.79

TABLE VI: CLASSIFICATION ACCURACY WITH SVM

Dataset	Baseline	BRSD	TL-Linear	TL-Exp
SatImage	0.22	0.25	**0.26**	**0.26**
iris	0.62	0.97	**0.98**	**0.98**
Letter	0.05	0.40	0.49	**0.51**
Wine	0.22	**0.50**	0.41	0.45
ColonTumor	**0.57**	**0.57**	**0.57**	**0.57**
Diabetes	0.63	0.66	**0.67**	**0.67**
Glass	0.07	**0.43**	**0.43**	**0.43**
Haberman	0.20	0.73	**0.81**	**0.81**
Mfeat	0.08	0.13	**0.17**	0.16
Wdbc	0.62	0.62	**0.64**	**0.64**
Sonar	0.50	0.50	0.53	**0.57**
Spambase	0.60	0.60	0.65	**0.69**
Vehicle	0.19	0.22	**0.23**	**0.23**
WaveForm	0.51	0.84	**0.85**	**0.85**

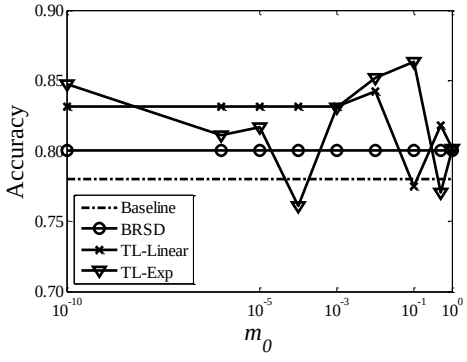


Fig. 8. Impact of m_0 on the algorithm performance

B. Analysis for the Algorithm

In the clustering analysis processing of TL-FNDCR, the parameter m_0 is crucial to the property of the fuzzy neighborhood membership degree ((4) to (7)). For further studying the performance of the proposed algorithm, the influence of m_0 on the algorithm performance is considered. Here, the classification results of base classifier C4.5 on *Spambase* dataset with different values of m_0 are shown in Fig. 8. m_0 is tuned in the range [10^{-10} 10^{-6} 10^{-5} 10^{-4} 10^{-3} 10^{-2} 0.1 0.5 1].

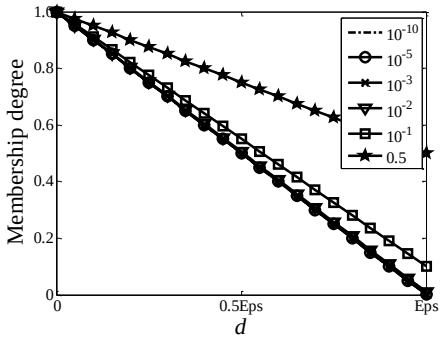


Fig. 9. Linear fuzzy neighborhood membership degree of different m_0

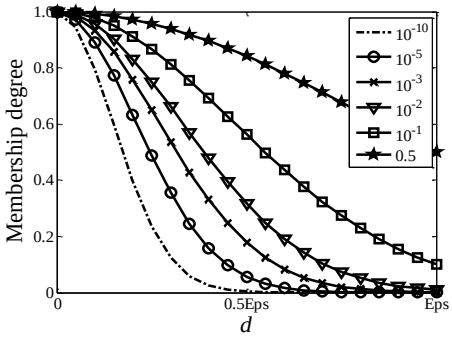
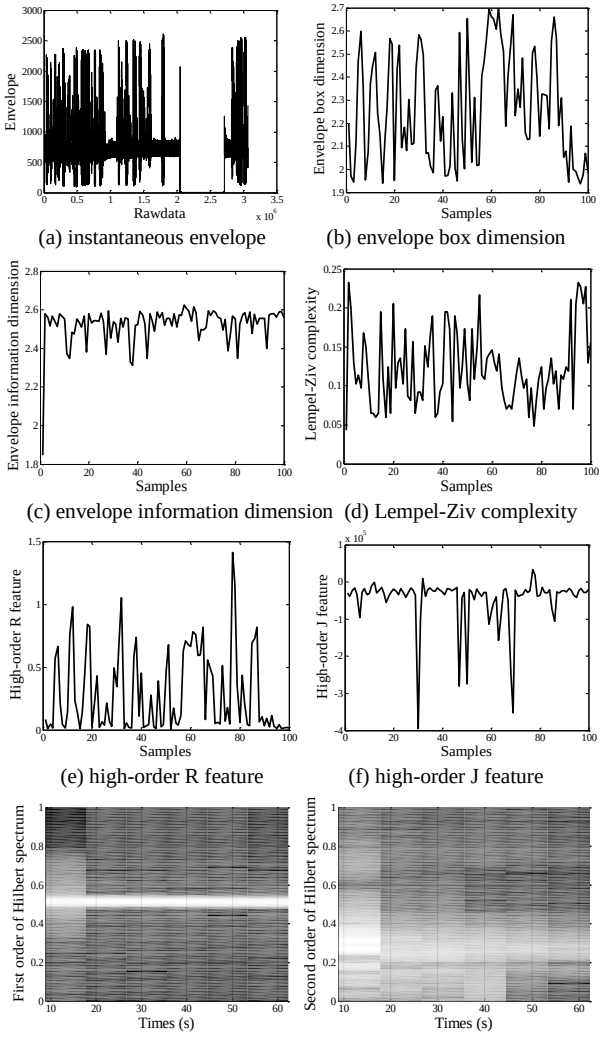


Fig. 10. Exponential fuzzy neighborhood membership degree of different m_0

Fig. 8 clearly shows that the performance of TL-FNDCR is affected by the setting of the parameter m_0. Since Baseline and BRSD do not have a m_0 parameter, their performance does not depend on it. The classification accuracy of TL-Linear becomes stable as m_0 approaches 0 while that of TL-Exp fluctuates throughout. It is can be explained by the different properties of their fuzzy neighborhood membership degree. In the linear case (Fig. 9), the neighborhood membership degree tends to be the same when m_0 approaches 0. However, in the exponential setting (Fig. 10), the neighborhood membership degree is obviously different from each other no matter the variations of m_0. When m_0 takes a very small value (e.g., m_0=10^{-10}), there is a large region in the *Eps*-neighborhood where the neighborhood membership degree approximates to 0 as well. That is, a much smaller value for m_0 does not only change the properties of the fuzzy neighborhood membership degree but also reduces the actual value of *Eps*. According to the above analysis, in the TL-Exp setting, the value of m_0 should not be too small and can be fixed in the range [10^{-3}, 10^{-1}]. On the contrary, for TL-Linear, the value of m_0 can be simply taken as 0.

C. Application to Communication Specific Emitter Identification

Communication specific emitter identification is widely used in applications such as spectrum management, cognitive radio, network intrusion detection, intelligence gathering, etc. This system discerns wire-less radio emitters of interest only based on the external signal feature measurements. However, in the practical application, the feature of the emitter signal is always changing along with different operation modes, different times and other conditions. It is difficult to make sure that the training data collected previously are suitable for the current target task. Here, the proposed transfer learning algorithm based on TL-FNDCR is applied to this application.



(a) instantaneous envelope (b) envelope box dimension

(c) envelope information dimension (d) Lempel-Ziv complexity

(e) high-order R feature (f) high-order J feature

(g) first order of Hilbert spectrum (h) second order of Hilbert spectrum

Fig. 11. Radio emitter signal and extracted features

Digital radios with the same type and same modulation mode are selected as the specific emitters, whose transmitting signal bandwidth is 25 KHz. The signal is

sampled at the sampling frequency of 204.8 KHz under different conditions, e.g., different work frequencies (160MHz or 410MHz), different speakers (speaker 1, speaker 2 or speaker 3), and different receive distances (short distance with direct wave or long distance without direct wave). After the raw data of emitter signal are obtained, feature extraction is conducted. The extracted features include the widely adopted emitter feature such as envelope box dimension, envelope information dimension, Lempel-Ziv complexity, high-order spectrum, and Hilbert spectrum. Fig. 11 shows the instantaneous envelope and extracted features of one radio emitter's signal.

To validate the performance of transfer learning, we select data sets under various conditions as source domain and target domain. Information on these data sets are tabulated in Table VII.

100 samples of each class are randomly chosen from 'Target' in Table VII as test set to evaluate the performance of learned hypothesis. One source domain is firstly selected from Table VII, then $100\times r$ samples of each class are randomly selected from the determined source domain for the experiment. The proportion parameter r is tuned in the range [0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]. C4.5 and Naïve Bayes are used as base classifier respectively. The experiments are repeated for 20 times with different source domain data and target domain data. The average accurate rates are recorded in Table VIII to Table XI.

TABLE VII: EXPERIMENTAL RADIO EMITTER DATA

Domains	Work frequency	Speaker	Receive distance
Target	160MHz	Speaker 1	long distance
Source 1	410MHz	Speaker 1	long distance
Source 2	160MHz	Speaker 2	long distance
Source 3	160MHz	Speaker 3	long distance
Source 4	160MHz	Speaker 1	short distance

TABLE VIII: CLASSIFICATION ACCURACY WHEN SOURCE DATA COMES FROM SOURCE 1

r	C4.5				Naive Bayes			
	Baseline	BRSD	TL-Linear	TL-Exp	Baseline	BRSD	TL-Linear	TL-Exp
0.05	0.4425	0.6195	0.7100	**0.7225**	0.4680	0.6505	**0.7905**	**0.7905**
0.1	0.4355	0.5740	**0.6860**	**0.6860**	0.4680	0.6515	0.7480	**0.7715**
0.2	0.3820	0.6905	**0.8560**	0.8510	0.4050	0.7230	**0.7865**	0.7855
0.3	0.4250	0.7980	**0.8795**	0.8405	0.4730	0.8050	**0.8245**	**0.8245**
0.4	0.4590	0.7280	**0.8825**	0.8700	0.4700	0.6925	0.8620	**0.8660**
0.5	0.4160	0.6685	**0.8725**	**0.8725**	0.4185	0.6270	0.7775	**0.8195**
0.6	0.4525	0.5470	0.8175	**0.8205**	0.4735	0.5630	**0.7955**	**0.7955**
0.7	0.3035	0.5990	**0.8475**	**0.8475**	0.4515	0.6920	0.7965	**0.8040**
0.8	0.4830	0.5305	**0.8070**	0.8050	0.4985	0.5600	0.7185	**0.7345**
0.9	0.4615	0.5295	0.8870	**0.8930**	0.4615	0.5225	**0.8260**	**0.8260**
Ave.	0.4261	0.6285	**0.8246**	0.8209	0.4588	0.6487	0.7926	**0.8018**

TABLE IX: CLASSIFICATION ACCURACY WHEN SOURCE DATA COMES FROM SOURCE 2

r	C4.5				Naive Bayes			
	Baseline	BRSD	TL-Linear	TL-Exp	Baseline	BRSD	TL-Linear	TL-Exp
0.05	0.5373	0.6700	0.7235	**0.7530**	0.5310	0.6913	0.7560	**0.7595**
0.1	0.6048	0.7113	0.8115	**0.8415**	0.5468	0.7310	**0.8740**	0.8730
0.2	0.6050	0.7013	0.7985	**0.8235**	0.6030	0.6983	0.7635	**0.7830**
0.3	0.6263	0.700	0.7885	**0.8005**	0.5760	0.6703	0.7650	**0.7700**
0.4	0.6400	0.7840	0.8800	**0.8850**	0.6095	0.7738	**0.9450**	**0.9450**
0.5	0.5523	0.6188	0.7430	**0.7610**	0.5733	0.5955	0.7180	**0.7335**
0.6	0.5878	0.6585	0.7140	**0.7225**	0.5838	0.6393	**0.7730**	**0.7730**
0.7	0.6053	0.6063	**0.7640**	0.7435	0.5753	0.6355	**0.7200**	**0.7200**
0.8	0.5948	0.6620	**0.7960**	**0.7960**	0.6365	0.7000	0.7515	**0.7565**
0.9	0.5470	0.6153	0.7555	**0.7600**	0.5720	0.6128	**0.7610**	0.7605
Ave.	0.5901	0.6728	0.7775	**0.7887**	0.5807	0.6748	0.7827	**0.7874**

TABLE X: CLASSIFICATION ACCURACY WHEN SOURCE DATA COMES FROM SOURCE 3

r	C4.5				Naive Bayes			
	Baseline	BRSD	TL-Linear	TL-Exp	Baseline	BRSD	TL-Linear	TL-Exp
0.05	0.5990	0.6743	0.7840	**0.7865**	0.6085	0.7340	0.8595	**0.8625**
0.1	0.6600	0.6895	0.7245	**0.7550**	0.6240	0.7565	**0.8090**	0.8055
0.2	0.6813	0.7625	**0.8985**	0.8870	0.6245	0.7668	**0.8065**	0.8055
0.3	0.7045	0.7788	**0.8340**	0.8315	0.6343	0.7055	0.8345	**0.8360**
0.4	0.6793	0.7158	**0.7595**	**0.7595**	0.6965	0.7223	0.7355	**0.7495**
0.5	0.6838	0.7780	0.8265	**0.8460**	0.6915	0.7755	**0.8545**	**0.8545**
0.6	0.6743	0.7048	**0.8105**	0.8095	0.6678	0.6848	0.7835	**0.7900**
0.7	0.6795	0.7485	0.8035	**0.8240**	0.6905	0.7250	0.7935	**0.8095**
0.8	0.6665	0.7765	0.8560	**0.8575**	0.7065	0.7165	0.8340	**0.8445**
0.9	0.7220	0.7785	0.8305	**0.8340**	0.7675	0.7430	0.8435	**0.8765**
Ave.	0.6750	0.7407	0.8128	**0.8191**	0.6712	0.733	0.8154	**0.8234**

TABLE XI: CLASSIFICATION ACCURACY WHEN SOURCE DATA COMES FROM SOURCE 4

r	C4.5				Naive Bayes			
	Baseline	BRSD	TL-Linear	TL-Exp	Baseline	BRSD	TL-Linear	TL-Exp
0.05	0.5000	0.5000	0.6400	**0.6475**	0.4400	0.5000	**0.7425**	**0.7425**
0.1	0.4753	0.5055	**0.6220**	0.6050	0.4605	0.5143	**0.6460**	0.6400
0.2	0.5000	0.5103	**0.7300**	**0.7300**	0.5060	0.533	**0.7095**	**0.7095**
0.3	0.5048	0.5283	**0.7730**	**0.7730**	0.4780	0.4795	0.6720	**0.6830**
0.4	0.5000	0.5013	**0.7225**	**0.7225**	0.4925	0.4890	**0.7315**	0.7265
0.5	0.5248	0.5250	**0.6930**	**0.6930**	0.4718	0.4555	**0.5775**	0.5770
0.6	0.5248	0.5000	**0.6770**	0.6765	0.4823	0.4840	**0.5765**	0.5725
0.7	0.5000	0.5000	0.5870	**0.6115**	0.4793	**0.4998**	0.4925	0.4985
0.8	0.5248	0.5250	**0.6240**	**0.6240**	0.4880	0.4850	**0.5945**	0.5910
0.9	0.5000	0.5003	**0.5615**	0.5545	0.4698	0.4730	**0.5805**	**0.5805**
Ave.	0.5055	0.5096	0.6630	**0.6638**	0.4768	0.4913	**0.6323**	0.6321

It can be seen that the TL-FNDCR has achieved higher accuracies compared with Baseline and BRSD. The highest classification accuracy of TL-FNDCR can be as high as 94.5%. All of these experimental results validate that our method is more suitable for communication specific emitter identification task, where the data is always heterogeneous under different conditions. Compared with Source 4, the classification results of the first three source domains are better. The reason can be concluded that the difference of data sets is bigger when receive distance is varying.

Fig. 12 gives the average accuracies over all base classifiers and source domains. Ignoring the difference of base classifiers and source domains, the performances of TL-Linear and TL-Exp still have evident improvements. Obviously, the performance of TL-Exp is still slightly better than that of TL-Exp. In addition, it is can be found that the accurate rates do not always increase and even decreases along with the increasing of the number of the source domain samples. Here, a probable interpretation is that the useful information from the source domain to target domain is finite due to the domain bias. When the amount of source domain data is large enough, the source domain could not offer more extra useful information further but gives rise to disturbance to the target learning conversely.

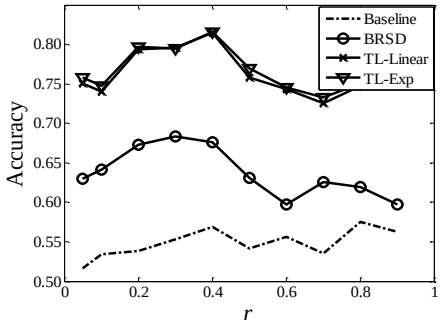


Fig. 12. Classification accuracy of different ratio of training samples

The experiment about the impact of parameter m_0 is also conducted here. As shown in Fig. 13, the accuracy of TL-Linear becomes stable as m_0 approaches 0 while that of TL-Exp fluctuates throughout. TL-Linear has the best results when m_0 approaches 0, while TL-Exp can get the best performance when m_0 in the range $[10^{-3}, 10^{-1}]$. The results are generally consistent with the conclusions in section IV.B.

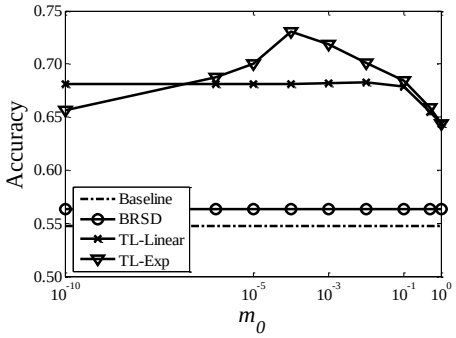


Fig. 13. Impact of m_0 on the algorithm performance

V. CONCLUSIONS

In this paper, a transfer learning algorithm called TL-FNDCR is proposed. TL-FNDCR can correct different types of domain differences and does not need to estimate the different distribution directly. With the introduction of the fuzzy neighborhood membership degree, TL-FNDCR is more robust to datasets with various shapes and densities and could explore more data structure information. Besides, the Gaussian kernel function is also applied to measure the similarities between samples to improve the reliability of the new training samples. Finally, the method can select much more suitable training data for the target domain learning. Extensive experiments on public data sets are carried out to evaluate the performance of the proposed method. For further showing the practicality, the proposed algorithm is applied to the task of communication specific emitter identification and the result is also satisfying.

TL-FNDCR can naturally explore the data structure for transfer learning, so it effectively reduces the bias across domains and significantly increases the learning performance. However, there is still much work to do in some aspects for further improvement of TL-FNDCR, e.g., the more effective estimation of *Eps* and the more reliable re-sampling strategy.

ACKNOWLEDGMENT

Both authors would like to acknowledge the support of Key Laboratory of Electronic Restriction of Anhui Province, National High-tech R&D Program (863 Program), and Anhui Provincial Natural Science Foundation (NO.1308085QF99, NO.1408085MKL46).

REFERENCES

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

[2] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowledge-Based Systems*, vol. 80, pp. 14-23, May. 2015.

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

[4] L. Shao, F. Zhu, and X. L. Li, "Transfer learning for visual categorization: A survey," *IEEE Trans. Neural Networks and Learning Systems*, vol.26, no. 5, pp. 1019-1034, June 2015.

[5] P. Yang and W. Gao, "Information-theoretic multi-view domain adaptation: A theoretical and empirical study," *Journal of Artificial Intelligence Research*, vol. 49, no. 1, pp. 501-525, Mar. 2014.

[6] F. M. Wei, J. P. Zhang, C. Yan, and J. Yang, "FSFP: Transfer learning from long texts to the short," *Appl. Math. Inf. Sci.*, vol. 8, no. 4, pp. 2033-2044, July 2014.

[7] Z. H. Deng, K. S. Choi, and Y. Z. Jiang, "Generalized hidden-mapping ridge regression, knowledge-leveraged inductive transfer learning for neural networks, fuzzy systems and kernel methods," *IEEE Trans. on Cybernetics*, vol. 44, no. 12, pp. 2585-2599, Mar. 2014.

[8] M. T. Bahadori, Y. Liu, and D. Zhang, "A general framework for scalable transductive transfer learning," *Knowl. Inf. Syst.*, vol. 38, no. 1, pp. 61-83, Jan. 2014.

[9] T. Martín-Wanton, J. Gonzalo, and E. Amigó, "An unsupervised transfer learning approach to discover topics for Online Reputation Management," in *Proc. 22nd ACM International Conf. Information & Knowledge Management*, San Francisco, 2013, pp. 1565-1568.

[10] B. Cheng, M. Liu, H. Suk, D. Shen, and D. Q. Zhang, "Multimodal manifold-regularized transfer learning for MCI conversion prediction," *Brain Imaging and Behavior*, vol. 9, no. 4, pp. 913-926, Dec. 2015.

[11] S. Z. Wang and J. Yang, "Learning to transfer privileged ranking attribute for object classification," *Journal of Information & Computational Science*, vol. 12, no. 1, pp. 367-380, Jan. 2015.

[12] G. Mesnil, S. Rifai, A. Bordes, X. Glorot, Y. Bengio, and P. Vincent, "Unsupervised and transfer learning under uncertainty: From object detections to scene categorization," in *Proc. 2nd International Conf. Pattern Recognition Applications and Methods*, Barcelona, 2013, pp. 345-354.

[13] B. Du, L. P. Zhang, D. C. Tao, and D. Y. Zhang, "Unsupervised transfer learning for target detection from hyperspectral images," *Neurocomputing*, vol. 120, no. 10, pp. 72-82, Mar. 2013.

[14] H. Y. Wang, V. W. Zheng, J. Zhao, and Q. Yang, "Indoor localization in multi-floor environments with reduced effort," in *Proc. IEEE International Conf. Pervasive Computing and Communications*, Mannheim, 2010, pp. 244-252.

[15] M. Fang, J. Yin, X. Q. Zhu, and C. Q. Zhang, "TrGraph: Cross-Network transfer learning via common signature subgraphs," *IEEE Trans. Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2536-2549, Mar. 2015.

[16] R. A. Shafik, A. Das, L. A. Maeda-Nunez, S. Yang, G. V. Merrett, and B. M. Al-Hashimi, "Learning transfer-based adaptive energy minimization in embedded systems," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, in Press, Oct. 2015.

[17] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *Proc. 21th International Conf. Machine Learning*, Banff, 2004, pp. 903-910.

[18] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in *Proc. 20th Annual Conf. Neural Information Processing Systems*, Vancouver, 2006, pp. 601-608.

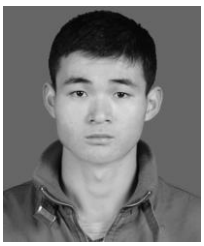
[19] M. Sugiyama, S. Nakajima, H. Kashima, P. Bünau, and M Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Proc. 22th Annual Conf. Neural Information Processing Systems*, Vancouver, 2008, pp. 1433-1440.

[20] J. Ren, X. Shi, W. Fan, and P. S. Yu, "Type independent correction of sample selection bias via structural discovery and re-balancing," in *Proc. SDM'08 Conf.*, Auckland, 2008, pp. 565-576.

[21] G. Ulutagaya and E. Nasibovb, "Fuzzy and crisp clustering methods based on the neighborhood concept: A comprehensive review," *Journal of Intelligent & Fuzzy Systems*, vol. 23, no. 6, pp. 271-281, Nov. 2012.

[22] W. Fan, and I. Davidson, "On sample selection bias and its efficient correction via model averaging and unlabeled examples," in *Proc. the SIAM International Conf. Data Mining*, Philadelphia, 2007, pp. 320-331.

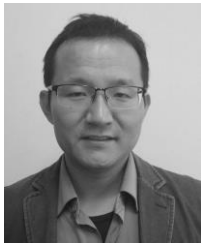
[23] E. Frank, M. Hall, P. Reutemann, and L. Trigg. Waikato environment for knowledge analysis. [Online]. Available: http://www.cs.waikato.ac.nz/~ml/weka/



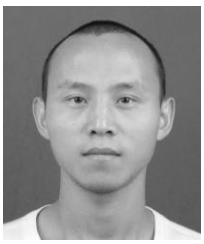
Zhen Liu was born in Anhui Province, China, in 1989. He received the B.S. degree in electrical engineering & automation from Anhui University, Hefei, China, in 2010 and the M.S. degree in circuits & systems from Electronic Engineering Institute, Hefei, China, in 2013. He is currently pursuing the Ph.D. degree with the Department of Communications, Electronic Engineering Institute, Hefei, China. His research interests include communication specific emitter identification, intelligent computing, data mining and machine learning.

Jun-An Yang was born in Anhui Province, China, in 1965. He received the B.S. degree in radio technology from Southeast University, Nanjing, China in 1986 and the M.S. degree in communication & information systems from Electronic Engineering Institute, Hefei, China, in 1991. He received the

Ph.D. degree in signal & information processing from University of Science and Technology of China (USTC), Hefei, China, in 2003. He is currently a professor in the Department of Communications at Electronic Engineering Institute, Hefei, China. His research interests include neural computing, large-scale machine and computer vision.



Hui Liu was born in Anhui Province, China, in 1983. He received the B.S. degree in communication engineering from Wuhan University, Wuhan, China, in 2005. He received the M.S. degree and the Ph.D. degree in communication & information systems from Electronic Engineering Institute, Hefei, China, in 2008 and 2011 respectively. He is currently a lecturer in the Department of Communications at Electronic Engineering Institute, Hefei, China. His research interests include intelligent information processing and cognitive communication.



Wei Wang was born in Anhui Province, China, in 1987. He received the B.S. degree in mechanism design, manufacturing & automatization from Northwestern Polytechnical University, Xi'an, China, in 2008. He received the M.S. degree and the Ph.D. degree in precision instrument and machinery from University of Science and Technology of China (USTC), Hefei, China, in 2011 and 2014 respectively. He is currently a lecturer in the Department of Communications at Electronic Engineering Institute, Hefei, China. His research interests include intelligent information processing and computer vision.