Fuzzy Transfer Learning Using an Infinite Gaussian Mixture Model and Active Learning

Hua Zuo, Jie Lu, *Fellow, IEEE*, Guangquan Zhang, and Feng Liu, *Student Member, IEEE*

***Abstract*— Transfer learning is gaining increasing attention due to its ability to leverage previously acquired knowledge to assist in completing a prediction task in a related domain. Fuzzy transfer learning, which is based on fuzzy system, especially fuzzy rule-based models, is developed because of its capability to deal with the uncertainty in transfer learning. However, two issues with fuzzy transfer learning have not yet been resolved: choosing an appropriate source domain, and efficiently selecting labeled data for the target domain. This study presents an innovative method based on fuzzy rules that combines an infinite Gaussian mixture model (IGMM) with active learning to enhance the performance and generalizability of the constructed model. An IGMM is used to identify the data structures in the source and target domains, providing a promising solution to the domain selection dilemma. Further, we exploit the interactive query strategy in active learning to correct imbalances in the knowledge to improve the generalizability of fuzzy learning models. Through experiments on synthetic datasets, we demonstrate the rationality of employing an IGMM and the effectiveness of applying an active learning technique. Additional experiments on real-world datasets further support the capabilities of the proposed method in practical situations.**

***Index Terms*—Transfer learning, fuzzy rules, domain adaptation, machine learning, regression**

I. Introduction

THE achievements of machine learning [1] in applications such as prediction, computer vision, biology, and business management have deeply affected all walks of life. And the rise of deep learning [2] is further advancing the development of machine learning in many areas of industry. However, many of the well-known algorithms are fundamentally supervised learning [3] processes, and the performance and generalizability of the resulting models tends to rely on massive amounts of labeled data. Unfortunately, in some fields, especially in new and emerging areas of business, gathering enough labeled data to properly train a model is difficult, even impossible. Without enough labeled data, the accuracy of a model suffers. Thus, transfer learning [4] has emerged as a potential solution.

Transfer learning aims to handle tasks in one domain (the target domain) much more quickly and effectively using knowledge from a related domain (the source domain). Barring cold start problems, transfer learning is well-suited to situations where a model performs poorly due to outdated or scant data. And the idea of transfer learning is related to data stream mining [5, 6], which is also apply the changing data distribution.

Some examples of successful transfer learning applications include: using already-categorized French documents to help classify English documents [7]; detecting a user's current location given previously collected WiFi data [8]; and predicting the failure of banks in Australia based on the data of banks in America [9].

To date, transfer learning methods can be categorized into two forms of training: semi-supervised learning [10, 11] and unsupervised learning [12, 13]. In semi-supervised settings, the source domain contains abundant labeled data, while the target domain has very little labeled data but an adequate amount of unlabeled data. In unsupervised settings, the target domain only contains unlabeled data. In addition to the learning approach, knowledge transfer problems can be categorized by whether the feature spaces in the two domains are homogeneous [14, 15] or heterogeneous [16, 17]. In homogeneous spaces, the source domain and target domain share the same feature space but the feature distributions are different. Heterogeneous spaces are more challenging. Here, the both the feature space and the data distributions differ across the two domains.

Transfer learning sits within the area of machine learning. Hence, its methods and basic training models rely on many notable machine learning techniques, such as SVM [7, 18], neural networks [19], and Bayesian models [20]. Some researchers have also explored deep learning for its ability to transfer knowledge between deep models [21]. In practice, it is common to pre-train a ConvNet [22] on a very large dataset, then remove the last fully-connected layer, treating the rest of the ConvNet as a fixed feature extractor for the new dataset. This idea was motivated by the observation that the earlier features of a ConvNet contain many generic characteristics that could be useful in other tasks, such as edge detectors or color blob detectors, but the later layers of the ConvNet progressively become more specific to the details of the classes in the original dataset.

Although machine learning algorithms are further advancing the development of AI, their lack of interpretability is very

Manuscript received February 08, 2018; revised May 24, 2018; accepted July 16, 2018. This work was supported by the Australian Research Council under DP 170101632.

H. Zuo, J. Lu, G. Zhang and F. Liu are with the Decision Systems & e-Service Intelligence Lab, Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia (e-mail: Hua.Zuo@uts.edu.au; Jie.Lu@uts.edu.au; Guangquan.Zhang@uts.edu.au, Feng.Liu-2@student.uts.edu.au).

controversial [23]. Fuzzy systems have gained popularity because they are easy to understand and allow for clear explanations. Further, their ability to deal with imprecision, vagueness, and ambiguity makes fuzzy models powerful in cases with uncertainty. Thus, many scholars have turned to fuzzy systems as a solution for transfer learning problems with promising results. Behbood et al. [24, 25] proposed a fuzzy-based transductive transfer learning approach to long-term bank failure prediction models with source and target domains that have different data distributions. They first applied a fuzzy neural network to predict the initial labels for data in the target domain, then used fuzzy similarity measures to refine the labels. To improve performance, they simultaneously took similarity and dissimilarity into account during the refinement process. By using fuzzy techniques for the similarity measurement, the authors revealed the advantage of fuzzy logic in knowledge transfers when the target domain lacks critical information, is vague, or involves uncertainty. Deng et al. [26, 27] proposed a series of transfer learning methods based on a Mamdani-Larsen-type fuzzy system and a Takagi-Sugeno-Kang (TSK) fuzzy model to deal with insufficient data scenarios through the integration of a corresponding knowledge-leverage mechanism. Further, their methods were applied to recognizing electroencephalogram signals in environments with a data shortage.

All the above fuzzy transfer learning methods incorporate fuzzy rule-based systems. Determining the number of fuzzy rules is a critical factor that affects the performance of the constructed fuzzy models. Some researchers have studied the number of fuzzy rules in fuzzy systems [28]. H. Rong et al. [29] introduced the concept of "influence" in a fuzzy rule and using this, fuzzy rules are added or removed dynamically based on the input data received. M. Pratama et al. [30] proposed a parsimonious network based on a fuzzy inference system in which the fuzzy rules can be stitched up and expelled by virtue of the statistical contributions of the fuzzy sets and injected datum afterwards.

Some of our own previous research [31, 32] has focused on using fuzzy models to handle domain adaptation problems in regression tasks. We considered two main knowledge transfer scenarios depending on whether the domains are homogeneous or heterogeneous. In homogeneous settings, knowledge from source domain is transferred in the form of fuzzy rules. Three novel methods were developed to modify the existing fuzzy rules in the source domain to fit the target data: changing the input, changing the output, and changing both the input and output. In heterogeneous settings, CCA [33] is applied to map the source and target data to a latent feature space, which converts a heterogeneous problem into a homogeneous problem.

Despite these advancements in fuzzy rule-based transfer learning methods, two issues main issues are still outstanding. One critical factor that affects the performance of the constructed model is the similarity between the source domain and the target domain. Given multiple source domains, it is difficult to select an approximate source domain to fit the transfer learning task of the current target domain. The second issue is how to acquire enough labeled data of sufficiently high quality to build a model for the target domain. For example, even if some labeled data is available, if all or most of that data only pertain to one aspect of the domain, the constructed model will contain inherent bias. Hence, in this paper, we propose a method for dealing with these two issues to improve the accuracy of target models.

The main contributions of this work are twofold. First, the existing research on fuzzy transfer learning cannot identify the number of fuzzy rules effectively, and our method uses an IGMM to explore the structure of data in both domains, which provides a basis for the model's construction and the knowledge transfer procedure. Second, the imbalance information in the target domain leads to the bad generalizability of the model, and the proposed method applies active learning to assist with the selection of labeled data for the target model and improve its generalizability.

The remainder of this paper is structured as follows. Section II presents the preliminaries of this paper, including some important definitions in transfer learning, the Takagi-Sugeno fuzzy model, the IGMM, and active learning. Section III details the fuzzy rule-based method and how an IGMM and active learning are used to improve the performance of the target model. Sections IV and V present the validation tests using synthetic and real-world datasets. The final section concludes the paper and outlines future work.

II. Preliminaries

This section begins with some basic definitions of transfer learning, followed by an introduction to the Takagi-Sugeno fuzzy model, which is the basic learning model used in our method. An overview of the IGMM and active learning conclude the section.

A. Definition

Definition 1 (Domain) [4]: A domain is denoted by $D = \{F, P(X)\}$, where F is a feature space, and $P(X)$, $X = \{x_1, \cdots, x_n\}$, are the probability distributions of the instances.

Definition 2 (Task) [4]: A task is denoted by $T = \{Y, f(\cdot)\}$, where $Y \in R$ is the response, and $f(\cdot)$ is an objective predictive function.

Definition 3 (Transfer Learning) [4]: Given a source domain D_s, a learning task T_s, a target domain D_t, and a learning task T_t, transfer learning aims to improve learning of the target predictive function $f_t(\cdot)$ in D_t using the knowledge in D_s and T_s where $D_s \neq D_t$ or $T_s \neq T_t$.

In short, transfer learning aims to use previously acquired knowledge (from a source domain) to assist prediction tasks in a new, but related, domain (the target domain).

B. Takagi-Sugeno Fuzzy Models

A Takagi-Sugeno fuzzy model [34] is used to construct the fuzzy rules and transfer knowledge between the source and target domains. The model consists of c rules with the following representation:

$$\text{If } \boldsymbol{x} \text{ is } A_i(\boldsymbol{x}, \boldsymbol{v}_i), \text{ then } y \text{ is } L_i(\boldsymbol{x}, \boldsymbol{a}_i) \qquad i = 1, \dots, c \tag{1}$$

where $\boldsymbol{v}_i$ are the centers of the clusters that determine the layout of the fuzzy rules, and $\boldsymbol{a}_i$ defines the action of each rule on the input variables.

The set of fuzzy rules in the Takagi-Sugeno fuzzy model is constructed given a labeled dataset $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$ using two procedures in sequence. The first procedure is an unsupervised learning process that uses a fuzzy C-means algorithm to divide the input data into c clusters and identify the centers of the clusters $\boldsymbol{v}_i$. Each cluster represents a fuzzy rule. The second procedure computes the coefficients of the linear functions with a proven linear relationship to the output of the model [26].

C. The Infinite Gaussian Mixture Model (IGMM)

The finite Gaussian mixture model with k components (k Gaussian distributions) is written as

$$p(y|\mu_1, \mu_2, \dots, \mu_k, s_1, s_2, \dots, s_k, \pi_1, \pi_2, \dots, \pi_k) = \sum_{j=1}^{k} \pi_j \mathcal{N}\left(\mu_j, s_j^{-1}\right) \tag{2}$$

where μ_j are the means, s_j are the precisions (inverse variances), and π_j are the mixing proportions (which must be positive and sum to one). $\mathcal{N}$ is a (normalized) Gaussian distribution with a specified mean and variance, and $\mathbf{y} = \{y_1, \dots, y_n\}$ are the observations. We wish to find the best solution (π_j, μ_j, s_j) with respect to $\mathbf{y}$. However, k needs to be selected by a user and is sometimes sensitive to the training process. Thus, researchers find selecting k automatically a more desirable approach, which means that we need to find the best k even when $k \rightarrow \infty$. In [30], Rasmussen proposed an IGMM to explore the properties of (1) when $k \rightarrow \infty$. If we assume μ_j has Gaussian priors $p\left(\mu_j|\lambda, r\right) \sim \mathcal{N}(\lambda, r^{-1})$, s_j has Gamma priors $p\left(s_j|\beta, \omega\right) \sim \mathcal{G}(\beta, \omega^{-1})$, and π_j is given a symmetric Dirichlet prior (also known as multivariate beta) with a concentration parameter α/k, the limitation of the conditional posterior, when $k \rightarrow \infty$, is calculated by (3) and (4)

$$p\left(c_i = j|\mathbf{c}_{-i}, \mu_j, s_j, \alpha\right) \propto \frac{n_{-i,j}}{n-1+\alpha} s_j^{\frac{1}{2}} e^{-\frac{s_j\left(y_i-\mu_j\right)^2}{2}},$$
$$\text{for component } j \text{ where } n_{-i,j} > 0 \tag{3}$$

$$p(c_i \neq c_{i'} \text{ for all } i' \neq i|\mathbf{c}_{-i}, \lambda, r, \beta, \omega, \alpha) \propto \frac{\alpha}{n-1+\alpha} \int p\left(y_i|\mu_j, s_j\right) p\left(\mu_j, s_j\right| \lambda, r, \beta, \omega\left) d\mu_j ds_j \tag{4}$$

where c_i is a stochastic indicator variable taking its values from 1 . . . k, $\mathbf{c}_{-i} = (c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_n)$, and $n_{-i,j}$ is the number of observations, excluding y_i , that are associated with component j. Thus, we have conditional posterior for a single indicator given all the other indicators μ_j and s_j. Using (2) and the Gibbs sampling method, we can determine the value of k (i.e., by finding a new class or removing an existing class) based on the posterior probability for a single indicator, which means k can be selected automatically in a one-time sampling process [35]. After completing the sampling process T times, the k with the highest frequency is chosen as the final selection.

D. Active Learning

Active learning is a subfield of machine learning. The key hypothesis behind active learning is that the performance of a learning algorithm can be boosted if it is allowed to choose the data from which it learns [36]. In supervised machine learning systems, a large number of labeled instances are required to build a model. Sometimes these labels come at little or no cost, but in other cases, obtaining labels can be very difficult, time-consuming, or expensive. Active learning is well-suited to such scenarios, where labeled data is hard to obtain. A variety of different active learning strategies have been used to select unlabeled data for a human annotator to label. However, all these strategies require the “informativeness” of the unlabeled instances to be evaluated through a query strategy, such as uncertainty sampling [37], query-by-committee [38], expected model change [39], expected error reduction [40], or variance reduction [41].

Some research has solely focused on applying the active learning techniques in fuzzy models [42]. Lughofer et al. [43] proposed three criteria based on active learning for efficient sample selection in cases of data stream regression problems within an online active learning context. The selection criteria was developed in combination with an evolving generalized Takagi-Sugeno fuzzy model, which outperformed the conventional evolving TS models.

III. Domain Adaptation Using IGMM and Active Learning

This section presents the framework of our method and the motivation behind each procedure in overview and then in more detail. A theoretical analysis of the method’s performance index is also included.

A. The Framework

The proposed method consists of four main procedures: using an IGMM to reveal the structure of the data; applying active learning techniques to augment the information in the target domain; training the model in the source domain; and, finally, executing knowledge transfer in the form of fuzzy rules from the source domain to the target domain.

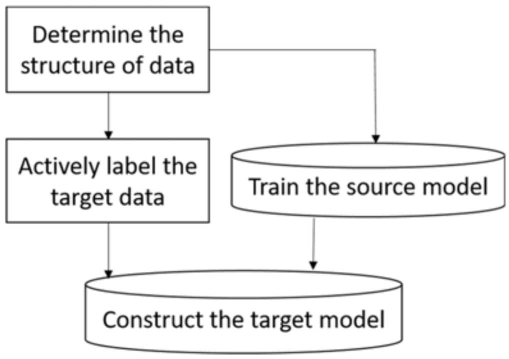

Determine the structure of data
Actively label the target data
Train the source model
Construct the target model

Fig. 1. The framework of the fuzzy transfer learning method

When constructing a Takagi-Sugeno fuzzy model, the number of clusters (fuzzy rules) should be known beforehand. However, it is often hard to determine the optimal number of clusters for a specific dataset without additional information. The most recent methods traverse all the numbers in a range and select the one with the best performance – a brute-force approach. However, this approach is costly and, sometimes, finding the optimal range is not easy.

IGMM provides a solution for clustering the data with no necessary prior knowledge to limit the number of components to be found. The idea behind IGMM is to fit the data distribution by mixing Gaussian distributions – a process that can be treated as a data structure detection procedure, which is of benefit to all cluster-based systems.

IGMM's role in the first procedure is to describe the structure of the data, which is beneficial for determining the number of fuzzy rules to use when constructing the subsequent models for both domains. Knowing the data structures in each domain is very important, as it greatly influences the effectiveness of the knowledge transfer process. For example, if the data structures in both domains are very similar, i.e., they have the same number of clusters and the means of the combined Gaussian distributions are close, then we can assume that the source and target domains have a strong corresponding relationship. Conversely, if the source and target domains have different numbers of clusters and the means of the distributions are quite far, it is reasonable to assume they have a weak relationship. To extract useful information, different transfer strategies should be adopted for different domain relationships. Thus, IGMM also provides a basis for guiding the transfer procedure.

After the correlation between two domains has been evaluated, the information in the target domain is assessed to determine the potential knowledge that can be transferred. The data in the target domain are divided into two groups: the instances with labels and the instances without labels. Compared to unlabeled data, labeled data contain more information and have a greater influence on the outcomes of prediction problems, especially regression tasks. Unlike classification tasks, where the results largely depend on the distribution and structure of the data, regression prediction tasks rely on more complicated factors. For instance, in the Takagi-Sugeno fuzzy regression model, the data distributions only determine the conditions of the fuzzy rules, i.e., whether or not each instance adheres to a particular fuzzy rule. The conclusions and the linear functions are governed by other factors that have a more critical impact on the final output. This is also the main reason that unsupervised domain adaptation is infeasible for regression tasks where only unlabeled data are available.

Therefore, labeled target data is necessary for the learning process in domain adaptation problems, and the quality of the labeled data greatly determines the quality of the transfer learning results. Thus, evaluating the quality of the labeled target data is an important part of the process if the results are to be used as a basis for subsequent procedures. Based on the characteristics of Takagi-Sugeno fuzzy model, the aim is to try and ensure the labeled data is a member of as many clusters as possible. If the labeled data in the target domain are spread among all clusters, we can assume the information is sufficient to train a well-performing model. Otherwise, if all the labeled data fall into one cluster or do not overlap with all the clusters, the information in the target domain is considered to be of low quality and needs to be augmented. Employing the IGMM and an active learning technique makes augmenting the information in target domain a reality.

The IGMM's "detection procedure" reveals the cluster structure of the unlabeled data in the target domain. Additionally, the IGMM evaluates the quality of the existing labeled target data, if lacking, the active learning technique adds information by selecting some instances from the unlabeled group for annotation by an expert. The process of selecting the unlabeled target data obeys one of the core principles of active learning: the instances that contain the most information are chosen. The informativeness of each instance is defined, which then supports the following data selection process. Thus, with the help of active learning, the number of the labeled target data is increased, and the information in the target domain is expanded.

The first two procedures can be thought of as pre-processing and preparation steps for constructing the models for each domain in the following step. A Takagi-Sugeno fuzzy model for the source domain is trained, and a set of fuzzy rules is generated. However, due to the discrepancies between the source and target domains, the fuzzy rules in the source domain cannot be directly used with the target data; they need to be modified.

An approach of changing the input variables is used to modify the existing fuzzy rules. Each input variable is assumed to be determined by some hidden comparing features, so the different distributions of input variables in the two domains are due to the different hidden features or different weights of these features. The idea behind changing the input variables is to adjust the number and weights of the hidden features so that the changed input distribution is more compatible with the target data. These modifications are made through an optimization process.

B. A Transfer Learning Method Based on IGMM and Active Learning

Consider two domains; a source domain with a large amount of labeled data, and a target domain with very little labeled data. The dataset in the source domain is denoted as $\boldsymbol{D} = \{(\boldsymbol{x}_1^s,\ y_1^s), \cdots, (\boldsymbol{x}_{N_s}^s,\ y_{N_s}^s)\}$, where $\boldsymbol{x}_k^s \in R^n$ $(k = 1, \cdots, N_s)$ is an n-dimensional input variable, the label $y_k^s \in R$ is a continuous variable, and N_s indicates the number of data pairs. The dataset in the target domain $\boldsymbol{H}$ consists of two subsets: one with labels and one without. $\boldsymbol{H} = \{\boldsymbol{H}_L, \boldsymbol{H}_U\} = \{\{(\boldsymbol{x}_1^t, y_1^t), \cdots, (\boldsymbol{x}_{N_{t1}}^t, y_{N_{t1}}^t)\},\ \{\boldsymbol{x}_{N_{t1+1}}^t, \cdots, \boldsymbol{x}_{N_t}^t\}\}$, where $\boldsymbol{x}_k^t \in R^n$ $(k = 1, \cdots, N_t)$ is the n-dimensional input variable, $y_k^t \in R$ is the label only accessible for the first N_{t1} data. $\boldsymbol{H}_L$ includes the instances with labels, and $\boldsymbol{H}_U$ contains the data without labels. The numbers of instances in $\boldsymbol{H}_L$ and $\boldsymbol{H}_U$ are N_{t1} and $N_t - N_{t1}$ respectively, and satisfy $N_{t1} \ll N_t$, $N_{t1} \ll N_s$.

In this problem setting, a well-performing model can be built for the source domain because there are sufficient labeled data. However, that model cannot be used directly to solve regression tasks in the target domain because the rules needs to be modified to fit the target data first. The following steps outline a fuzzy rule-based transfer learning method, based on IGMM and active learning, to modify the source model for use with the target data.

Step 1. Applying IGMM to discover the structure of data

This procedure reveals the data structure of both domains, with two benefits. First, identifying the data structures of each domain provides insights into the relationships between the data, which can be used to guide the transfer learning procedure. Second, understanding the data structures is conducive to selecting the most informative labeled data for the target domain.

Through two separate parses of an unsupervised learning process, IGMM simulates the distributions of the data in the source and target domains. The input data are $\{\boldsymbol{x}_1^s, \cdots, \boldsymbol{x}_{N_s}^s\}$ and $\{\boldsymbol{x}_1^t, \cdots, \boldsymbol{x}_{N_t}^t\}$. IGMM's exploration process is illustrated in the following example.

Fig. 2 shows the probabilities of various data structures in a dataset in histogram form. The x-coordinate represents the number of Gaussian distributions, i.e., the number of clusters, and the y-coordinate represents the number of times that the dataset is divided into the corresponding clusters. As the figure shows, in 2000 iterations of IGMM, the dataset was divided into three clusters more than 1000 times, into four clusters about 500 times, into one cluster about 250 times, and into two or five clusters less than 100 times. Therefore, we can conclude, with high probability, that the dataset is composed of three Gaussian distributions (clusters).

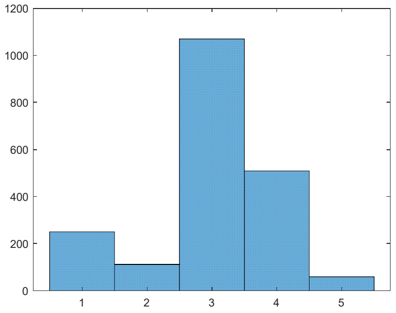


Fig. 2. Examples of the results for IGMM

Suppose the cluster range for the source domain is $[csmin, csmax]$, where csP has the highest probability. Similarly, the range for the target domain is $[ctmin, ctmax]$, with ctP as the highest probability. Comparing the histograms of the data reveals insights into the relationship between the two domains that can be used to select an appropriate transfer strategy. For instance, if csP is equal to ctP, then the corresponding parameters of the Gaussian distributions are also similar, which means the source and target domains are close, and the knowledge of source domain is likely to be highly beneficial in constructing the model for the target domain. Additionally, the number csP (=ctP) provides a good guide as to the number of fuzzy rules to use to build the prediction models for each domain. Further, this type of analysis can also be used to inform domain selection. For example, given multiple source domains to choose from, comparing the data structure of each candidate may reveal the most suitable match for the target domain. And, if none of the data structures match, i.e., csP is not identical to ctP, a different transfer strategy can be explored. However, even with the results of the structural analysis, determining the optimal number of fuzzy rules to use when constructing the models may still be difficult. Therefore, a brute-force approach across a reduced range of cluster numbers may be required to select the one that delivers the best performance. The range for the number of clusters is:

$$[cmin, cmax] \tag{5}$$

where $cmin = \max\{2, \min\{csmin, ctmin\}\}$, and $cmax = \min\{csmax, ctmax\}$. Restricting $cmin$ to not less than 2 ensures the nonlinearity of the model.

Step 2. Using active learning to augment the labeled target data

The purpose of this procedure is to increase the amount information in the target domain by actively selecting and labeling some of the data.

The procedure begins with an evaluation of the existing labeled target data. Given the analysis in Step 1, suppose we choose to construct the model with three fuzzy rules $c(= 3)$. FCM clusters the input data for the target domain into the membership matrix U. The index of H_L in H determines the membership of all the labeled target data to all the clusters. The membership matrix for the labeled target data is denoted as U_L:

$$U_L = \begin{bmatrix} 0.2 & 0.7 & 0.1 \\ 0.7 & 0.1 & 0.2 \\ 0.5 & 0.2 & 0.3 \end{bmatrix} \tag{6}$$

The number of labeled data in each cluster is counted, with each instance counted in the cluster with the highest membership. The statistic result is as follows:

$$SU_L = [2 \quad 1 \quad 0] \tag{7}$$

The first two clusters contain labeled data, but the third cluster does not, so active learning is used to augment the information in the target domain and, hopefully, populate this cluster.

"Informativeness" is the key criteria used for selecting which data is labeled in each cluster. Essentially, informativeness is a measure of the information contained in the data, and the level of informativeness is highly dependent on the cluster it is grouped with, i.e., an instance will have different levels of informativeness for different clusters. A concrete instance $\boldsymbol{x}_k^t$ in the ith cluster is highly informative when $\boldsymbol{x}_k^t$'s membership to the ith cluster is high. Thus, the informativeness of $\boldsymbol{x}_k^t$ in cluster i is defined as

$$I_i(\boldsymbol{x}_k^t) = U_{ki} \tag{8}$$

Further, a threshold d determines the minimum number of labeled target data needed for each cluster. Unlabeled data assessed as being highly informative according to the above definition are then selected and sent to experts to be annotated. Taking the example above, the $d-2$ unlabeled data with the highest informativeness in the first cluster are selected for labeling. Similarly, $d-1$ and d unlabeled data in the remaining two clusters are selected for labeling.

At the end of Step 2, the number of labeled target data has increased from N_{t1} to $3d$.

Step 3. Constructing a prediction model for the source domain

This procedure governs the construction of the source model M^s based on the source dataset $\boldsymbol{D}$. The formulation for model M^s follows

$$\text{if } \boldsymbol{x}_k^s \text{ is } A_i(\boldsymbol{x}_k^s, \boldsymbol{v}_i^s)\text{, then } y_k^s \text{ is } L_i(\boldsymbol{x}_k^s, \boldsymbol{a}_i^s) \qquad i = 1, \cdots, c \tag{9}$$

where $\boldsymbol{v}_i^s$ are the centers of the clusters that determine the conditions of the fuzzy rules, and $\boldsymbol{a}_i^s$ are the coefficients of the linear functions of the input variables that govern the conclusions of the fuzzy rules.

The number of fuzzy rules c depends on the results of the analysis in Step 1. If the data structure of the (chosen) source domain is similar to the data structure of the target domain, c is set to csP (=ctP). If the data structures are divergent, c is taken from the range $[cmin, cmax]$.

Given a sufficient amount of labeled source data, a well-performing prediction model M^s for the source domain can be built. However, the target domain will invariably contain different data distributions to the source data, and the model M^s would perform poorly if trained on the target data without prior modifications, which leads to Step 4.

Step 4. Modifying the existing fuzzy rules to fit the target data

Through this procedure, the hidden features in model M^s are adjusted, including the amounts and weights, to modify the input space so that the distributions of the input variables are more compatible with target data. This approach is based on the assumption that the input variables in both domains have similar, or even the same, hidden features.

Fig. 3 shows the mapping structure used to change the input space. The neurons in the hidden layers represent the connotative features to be used as input variables. The transformation of these neurons through the layers modifies their weights to ultimately construct input variables with new meanings and distributions that can be used to build a new target model $\bar{M}^t$.

After the mapping procedure, the fuzzy rules are transformed into

$$\text{if } \boldsymbol{x}_k^t \text{ is } A_i(\boldsymbol{\Phi}(\boldsymbol{x}_k^t), \boldsymbol{\Phi}(\boldsymbol{v}_i^s))\text{, then } y_k^t \text{ is } L_i((\boldsymbol{\Phi}(\boldsymbol{x}_k^t), \boldsymbol{a}_i^s) \quad i = 1, \cdots, c \tag{10}$$

With this modified input space, the new fuzzy rules, including the cluster centers and the linear coefficients can be used to predict outputs with the target data.

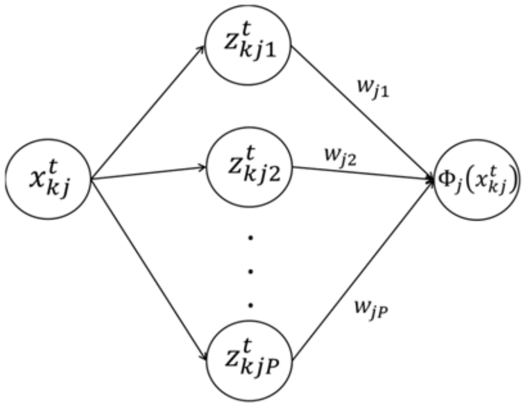


Fig. 3. Nonlinear mapping structure

$\boldsymbol{\Phi}$ is mapped through an optimization procedure using the target data, which was augmented in Step 2 to ensure a sufficient level of labeled data to construct a well-performing model. Therefore, once optimized, the cost function in (11) minimizes the distance between the outputs of model $\bar{M}^t$ and the real values in the target data. The cost function follows:

$$S = \sqrt{\frac{1}{N_{t1}} \sum_{k=1}^{N_{t1}} \left(\sum_{i=1}^{c} A_i(\Phi(\boldsymbol{x}_k^t), \Phi(\boldsymbol{v}_i^s))L_i(\Phi(\boldsymbol{x}_k^t), \boldsymbol{a}_i^s) - y_k^t\right)^2} + \frac{\lambda_2}{2} w^T w \tag{11}$$

The overall algorithm for the proposed method described above is provided in Algorithm 1.

Algorithm 1. Domain Adaptation Procedure based on IGMM and Active Learning

```
Input: $\boldsymbol{D}$, $\boldsymbol{H}$,
Output: $\boldsymbol{Y}_U$ for $\boldsymbol{H}_U$

1. Apply IGMM to $\{\boldsymbol{x}_1^s, \cdots, \boldsymbol{x}_{N_s}^s\}$ and $\{\boldsymbol{x}_1^t, \cdots, \boldsymbol{x}_{N_t}^t\}$
2. Decide the number of clusters $c$
3. Apply active learning to augment target information
   3.1 give the threshold $d$
   3.2 validate the current labeled target data
   3.3 find out the data with most information in each cluster
   3.4 select the target data and label them
4. Train source model $M^s$
5. Modifying the existing model to get $\bar{M}^t$
6. Use $\bar{M}^t$ to predict the response $\boldsymbol{Y}_U$ for $\boldsymbol{H}_U$
```

C. Performance Index

This section provides the formulations for the performance indexes of the constructed models used to evaluate the proposed method.

Each model is constructed through five-fold cross-validation. The source model M^s is trained on the dataset $\boldsymbol{D}$, and the root mean square error (RMSE) of M^s on training set is calculated by

$$Q = \sqrt{\frac{1}{N_{s1}} \sum_{k=1}^{N_{s1}} (\sum_{i=1}^{c} A_i(\boldsymbol{x}_k^s, \boldsymbol{v}_i^s\ ) L_i(\boldsymbol{x}_k^s, \boldsymbol{a}_i^s) - y_k^s)^2} \qquad (12)$$

where $\boldsymbol{v}_i^s$ are the centers of the clusters and $\boldsymbol{a}_i^s$ are the linear functions of the fuzzy rules . $\boldsymbol{x}_k^s$ is the input variable for the source data, and $\sum_{i=1}^{c} A_i(\boldsymbol{x}_k^s, \boldsymbol{v}_i^s\ ) L_i(\boldsymbol{x}_k^s, \boldsymbol{a}_i^s)$ is the corresponding output of the model M^s. y_k^s is the real output for $\boldsymbol{x}_k^s$.

The probability of model M^s is tested on the target data with

$$Q1 = \sqrt{\frac{1}{N_t - N_{t1}} \sum_{k=1}^{N_t - N_{t1}} (\sum_{i=1}^{c} A_i(\boldsymbol{x}_k^t, \boldsymbol{v}_i^s\ ) L_i(\boldsymbol{x}_k^t, \boldsymbol{a}_i^s) - y_k^t)^2} \qquad (13)$$

where $\boldsymbol{v}_i^s$ and $\boldsymbol{a}_i^s$ are the parameters of source model. $\boldsymbol{x}_k^t$ is the input variable of target data, and $\sum_{i=1}^{c} A_i(\boldsymbol{x}_k^t, \boldsymbol{v}_i^s\ ) L_i(\boldsymbol{x}_k^t, \boldsymbol{a}_i^s)$ is the corresponding output of the model M^s. y_k^t is the real output for $\boldsymbol{x}_k^t$.

The target model $\bar{M}^t$ is tested on the unlabeled dataset to verify the generalizability of the constructed model with

$$Q2 = \sqrt{\frac{1}{N_t - N_{t1}} \sum_{k=1}^{N_t - N_{t1}} (\sum_{i=1}^{c} A_i(\Phi(\boldsymbol{x}_k^t), \Phi(\boldsymbol{v}_i^s)\ ) L_i(\Phi(\boldsymbol{x}_k^t), \boldsymbol{a}_i^s) - y_k^t)^2} \qquad (14)$$

where $\boldsymbol{x}_k^t \in H_U$ are the data without labels. The real labels y_k^t are only available in the testing procedure.

IV. Experiments on Synthetic Datasets

In this section, we present the experiments conducted with synthetic datasets to validate the proposed method. We specifically evaluated the impact of the IGMM and the active learning technique on the performance of the constructed models. Firstly, the IGMM's ability is tested to explore the structure of data as a basis for the knowledge transfer procedure. Secondly, the ability of the active learning technique is tested to optimally augment information in the target domain. Thirdly, we compared our methods with some state-of-the-art transfer learning methods.

A. Exploring Data's Structure Using IGMM

The design of a well-performing transfer learning algorithm depends on the relationship between the source and target data. Using IGMM to explore the structure of the data provides guidance when selecting a domain in situations where multiple source domains are available since similar data structures in both domains can benefit the transfer of fuzzy rules from the source domain to the target domain. In this experiment, we considered two cases to simulate different relationships between the domains. The first case assumes that the source and target data stem from identical domains and, therefore, their distributions are quite similar. The second case considers two quite different source and target domains, where there is a great divergence in the data structures between each domain. A different transfer strategy was applied in each case, and IGMM plays a different role in each scenario.

1) Similar source and target data

Here, the target data was derived from the source domain, so the distributions and structures of data in both domains are very similar. Thus, IGMM's role is to provide guidance with domain selection – a challenging issue in transfer learning. For instance, suppose there are multiple source domains, but only one domain is the most suitable for the target domain. If IGMM is able to easily find the most suitable source domain, it would effectively improve the transfer learning performance.

Three groups of experiments were executed to illustrate the role of IGMM in selecting a source domain. In each group of experiments, the data in the target domain was generated with a different number of clusters, and three source domains with various numbers of clusters were prepared. The results of IGMM to the datasets in three groups of experiments are shown in Figs. 4-6.

From Fig. 4, we can see that the three source domains are divided into two, three, and four clusters, respectively, with the highest probabilities. Moreover, the target data consists of two clusters with the highest probability. In Fig. 5, the target data was generated with three clusters, and three source domains were generated with two, three and four clusters. In Fig. 6, the target data was generated with four clusters, and three source domains were generated with two, three and four clusters.

In analyzing the results of IGMM, in the first experiment, the first source domain has the same number of clusters as the target domain and should be the best choice for the target domain. In experiment 2, although the datasets in the source domain 1 and source domain 2 both have three clusters with highest probabilities, the source domain 2 has the similar data structure with target domain. Therefore, source domain 2 is the best choice for the target domain in experiment 2. Similarly, source domain 3 is the best choice for the target domain. To verify this conclusion, knowledge from the three source domains was transferred, in turn, to the target domain, and we compared the results to assess the transfer performance. These results are shown in Tables I-III.

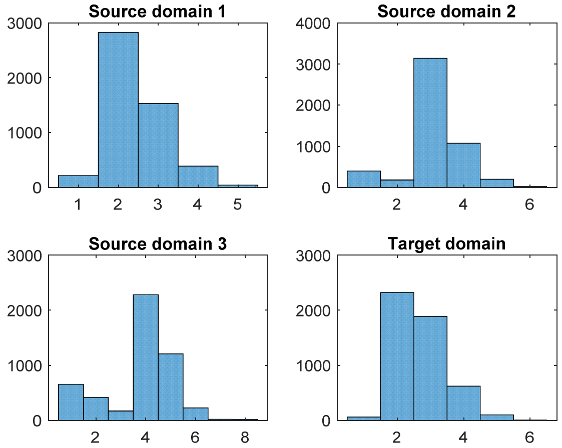


Fig. 4. The data structure in experiment 1

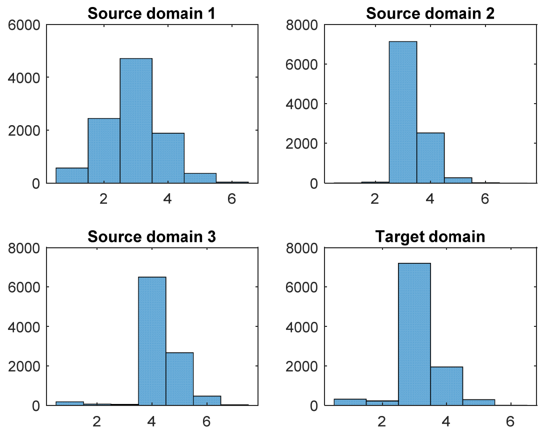


Fig. 5. The data structure in experiment 2

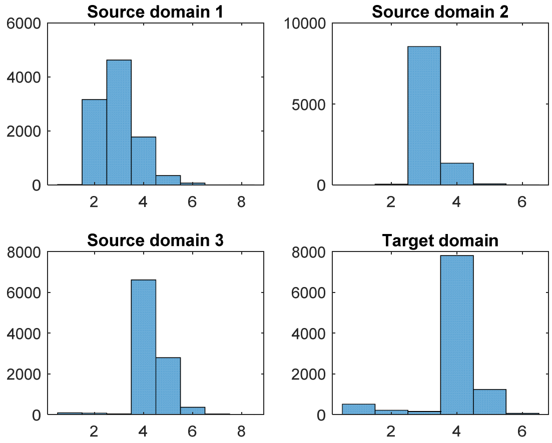


Fig. 6. The data structure in experiment 3

Table I: Various source domains and a two-cluster target domain

Datasets		Q	Q1	Q2
Source 1 (2)	Target (2)	0.0441± 0.0006	2.2774± 0.0000	1.8487± 0.0350
Source 2 (3)	Target (2)	0.0250± 0.0002	2.2743± 0.0000	1.8549± 0.0397
Source 3 (4)	Target (2)	0.1023± 0.0009	2.5293± 0.0003	1.9929± 0.0481

Table II: Various source domains and a three-cluster target domain

Datasets		Q	Q1	Q2
Source 1 (2)	Target (3)	0.0301± 0.0005	1.0320± 0.0001	0.8560± 0.0108
Source 2 (3)	Target (3)	0.0196± 0.0000	0.4586± 0.0000	0.7997± 0.0482
Source 3 (4)	Target (3)	0.0505± 0.0086	2.0170± 0.0067	0.8678± 0.0024

Table III: Various source domains and a four-cluster target domain

Datasets		Q	Q1	Q2
Source 1 (2)	Target (4)	0.0153± 0.0001	3.0610± 0.0001	1.6237± 0.0043
Source 2 (3)	Target (4)	0.0930± 0.0384	3.6952± 0.0586	1.4924± 0.0415
Source 3 (4)	Target (4)	0.1104± 0.0142	1.9950± 0.0069	1.1537± 0.0066

The first two columns in Table I represent the datasets for the source and target domains. The number in the brackets indicates the number of clusters in that dataset. All models were constructed through five-fold cross-validation; therefore, the results in the last four columns are displayed in the form of "mean±variance". The third column is the RMSE of the source model on the source data, and a low error means a well-performing regression prediction model was produced for the source domain. The fourth column is the RMSE of source model on the target data, which indicates that the source model is not compatible with target data. The results in column five show the performance of the target models constructed using the proposed methods. The mean values in the fifth column are less than in the fourth column, which means that the proposed method has greatly improved the prediction accuracy of the existing model in the target domain. In particular, comparing the results in the fifth column, we can conclude that the first source domain showed the best performance and has the same number of clusters as the target domain

We reached the same conclusion from the second and third experiments: that the results from the IGMM both improve the performance of the constructed target model and provide useful clues for the domain selection process. This conclusion, therefore, validates the role of IGMM in selecting a suitable domain in cases where the source and target domain have very similar structures.

The above three experiments tell us that the transfer learning has an obvious effect when the number of clusters (fuzzy rules) in the source and target domains is identical However, the distance of the data can also affect transfer learning. Hence, the next experiment was designed to explore the impact of data distance on the model's performance.

The target dataset was generated first, and the source datasets were generated based on the target data by gradually increasing the gap between the centers of the clusters and the linear functions. The center of the clusters and the linear functions in both domains follow the relation below:

$$\boldsymbol{v}_i^s = \boldsymbol{v}_i^t(\mathbf{1}+\varepsilon)\,, \qquad \boldsymbol{a}_i^s = \boldsymbol{a}_i^t(\mathbf{1}+\varepsilon) \tag{15}$$

where ε is an constant that controls the increment, and ε is a crucial parameter that controls the degree of difference between the source and target data.

As ε increases, the divergence between the source and target data becomes greater. The value of ε was set to increase from 0.05 to 0.5 in steps of 0.05. Thus, ten source datasets were generated and knowledge was transferred from each source dataset to the same target dataset. The results are shown in Table IV. Fig. 7 clearly displays the changing tendency in model performance when as the value of ε changes.

The red circles indicate the RMSE of the source model on the target data. The increasing trend shows that with an increase of ε, the discrepancy between the source data and target data becomes greater, and the source model becomes more mismatched to target data. The blue circles represent the performance of the target model using the proposed method. There are no obvious changes, but effectiveness of the proposed method is still verified. These results also indicate good transfer learning performance if the data structures of the two domains are similar.

Table IV: Performance of the constructed target model with a varying ε

ε	Q1	Q2
0.05	0.2851 ± 0.0000	0.2299±0.0055
0.1	0.5788 ± 0.0000	0.1931±0.0037
0.15	0.8833 ± 0.0000	0.2168±0.0024
0.2	1.1956 ± 0.0000	0.2602±0.0007
0.25	1.5246 ± 0.0000	0.1903±0.0002
0.3	1.8561 ± 0.0000	0.2517±0.0013
0.35	2.1902 ± 0.0000	0.2108±0.0004
0.4	2.5384 ± 0.0000	0.2607±0.0057
0.45	2.8882 ± 0.0000	0.3365±0.0173
0.5	3.2610 ± 0.0000	0.3403±0.0003

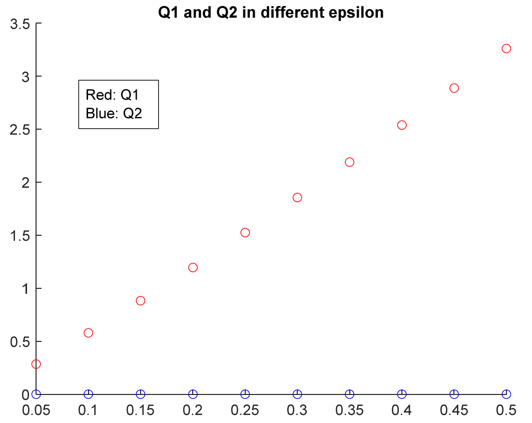


Fig. 7. The model’s performance with varying values for ε

2) *Different source and target*

With this set of experiments, we evaluated IGMM’s performance in cases where the source and target domains have very different data structures, i.e., where each domain has a different number of clusters (fuzzy rules). Three separate domain adaptation experiments were conducted. Three datasets were generated. Each time, one of the datasets was selected as the target domain, and the remaining two datasets were treated as the source domains. We used the approach of traversing all the clusters in source and target domains to determine the optimal number of clusters.

The analytical results from IGMM for the three datasets are shown in Fig. 8. The results of these experiments are shown in Tables V-VII.

The results in Tables V-VIII do not reveal an obvious rule for determining the optimal number of clusters in the domain adaptation process. Thus, in cases where the source and target domains have very different structures, the brute-force approach of trying all the numbers and selecting the one with the best performance remains the best option.

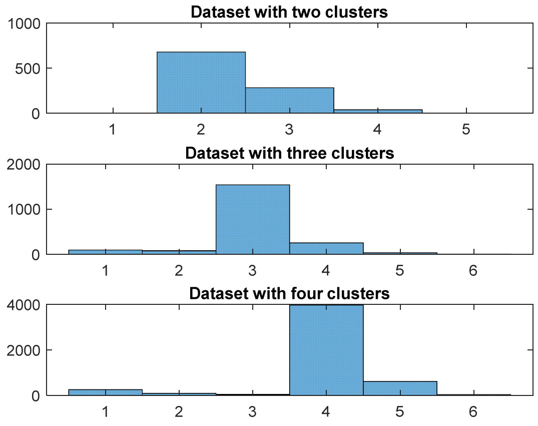


Fig. 8. The structure of the three datasets

Table V: The values of Q2 with varying clusters (target domain: two rules)

clusters	4r to 2r	3r to 2r
2	1.5003±0.4920	0.6256±0.0031
3	2.0345±0.2321	0.6150±0.1162
4	**0.5992±0.0089**	0.8732±0.5013
5	0.7783±0.0353	**0.4578±0.0245**
6	0.9645±0.1183	0.8543±0.3426

Table VI: The values of Q2 with varying clusters (target domain: three rules)

clusters	2r to 3r	4r to 3r
2	1.5431±0.0213	3.0206±0.0375
3	1.5438±0.0210	2.8064±0.4057
4	**1.4310±0.0172**	4.7779±0.2630
5	1.4653±0.0024	**1.7450±0.0041**
6	1.5910±0.0024	2.3647±0.0507

Table VII: The values of Q2 with varying clusters (target domain: four rules)

clusters	2r to 4r	3r to 4r
2	**1.0372±0.0072**	1.4666±0.0163
3	1.6847±0.0079	1.4228±0.0025
4	1.2896±0.0143	1.3803±0.0123
5	1.6691±0.0064	**1.3791±0.0077**
6	1.5846±0.0237	1.4657±0.0768

B. Augmenting the Information in the Target Domain with an Active Learning

The three experiments in this section were designed to verify the use of active learning technique in improving the performance of the built target model. In each experiment, the source and target datasets were generated with the same number of fuzzy rules (two, three, and four respectively), and all the labeled target data were selected from one cluster. The experimental results are shown in Table VIII. ‘Q2 (no active learning)’ represents the performance of the model without the benefit of the active learning technique. ‘Q2 (active learning)’ indicates the performance of the model with active learning.

Comparing values ‘Q2 (no active learning)’ and ‘Q2 (active learning)’ in the three experiments, we found that using the active learning technique significantly enhances the accuracy of the target model constructed using the proposed method.

Table VIII: Exploring the effect of the active learning technique

Clusters	Q1	Q2 (no active learning)	Q2 (active learning)
2	1.3676± 0.0001	1.4596± 0.0202	1.0731± 0.0075
3	0.3534± 0.0001	0.9250± 0.0320	0.8760± 0.0157
4	2.0330± 0.0026	2.1865± 0.0653	1.7536± 0.1304

In addition, we conducted the above experiments with different values of d to determine the impact of d on the performance of the presented method.

Table IX: Performance of models with varying d

d	Datasets (two clusters)	Datasets (three clusters)	Datasets (four clusters)
5	1.0731±0.0075	0.8760±0.0157	1.7536±0.1304
10	1.0768±0.0124	0.8518±0.0180	1.9097±0.0295
15	1.0076±0.0023	0.9004±0.0380	0.6976±0.0034
20	0.9702±0.0117	0.9019±0.0055	1.3242±0.0072

The results in Table IX show that the performance of the constructed target model does not display an increasing trend as the value of d increases, which indicates that d does not play a critical role in the active learning-based domain adaptation method. Since IGMM identifies the data structure of the dataset, i.e., the number of clusters, a small number of instances could represent the information of one cluster, and the design of our algorithm satisfies the requirement of covering the information in all clusters. Thus, the results in Table IX are reasonable and acceptable. Further, these results are a promising signal for good transfer learning performance with little labeled target data. Therefore, we tried some different vales of d, and selected the smallest one as long as the performance was almost the same.

Also, some experiments are implemented to explore the impact of P on the performance of the proposed method. The results are shown in Table X.

Table X: Performance of models with varying P

Datasets	RMSE with varying P				
	3	4	5	6	7
Dataset 1 (2r)	1.08± 0.01	1.10± 0.01	1.09± 0.01	1.24± 0.00	1.15± 0.01
Dataset 2 (3r)	2.45± 0.01	2.46± 0.02	2.53± 0.05	2.41± 0.01	2.54± 0.07
Dataset 3 (4r)	1.91± 0.03	1.66± 0.10	1.62± 0.03	1.26± 0.05	1.35± 0.04

From the results in Table X, there is no obvious trend with the increase of P, and our method could achieve a good performance even the number of the sigmoid functions is small. Since the mappings based on sigmoid functions are constructed in a nonlinear way, a small number of sigmoid functions could constitute complex mappings for the transformation of input space.

C. Comparing with state-of-the-art transfer learning methods

We have compared our method with three state-of-the-art non fuzzy transfer learning approaches, including TCA, SA, and GFK, and a TSK model-based fuzzy transfer learning method. The comparison is shown as follows in Table XI.

The left column in Table XI indicates the source and target domains. For example, '2r to 4r' indicates that the source domain is 'dataset 2r', and the target domain is 'dataset 4r'. In the first three experiments, the datasets in the source and target domains have the same number of fuzzy rules, and in the last six experiments, the datasets in two domains are designed with different numbers of fuzzy rules. The second and third columns are the two baselines of the transfer learning problem: 1) the root mean square error (RMSE) of the source model on the unlabeled target data $\boldsymbol{H}_U$; 2) the RMSE of the model trained using only target data on $\boldsymbol{H}_U$. The fourth to the sixth columns show the RMSE's of three famous transfer learning approaches (TCA, SA, and GFK) on $\boldsymbol{H}_U$, respectively. The results in the seven column show the RMSE of a TSK-based fuzzy method on $\boldsymbol{H}_U$. And the final column shows the RMSE of our proposed method on $\boldsymbol{H}_U$.

The results in Table XI show that the mean value of RMSE for our method is smaller than that of the three non-fuzzy methods and the TSK-method. Therefore, we can conclude that the performance of our method is superior to the existing state-of-the-art non-fuzzy transfer learning approaches and the TSK-method.

V. Experiments on Real-world Datasets

Studies of regression problems in domain adaptation are scarce, so there are no public datasets available to verify the proposed method. Hence, we used real-world datasets from the UCI Machine Learning Repository and modified them to simulate various regression domain adaptation problems. A detailed description of these modifications follows.

The first dataset concerns "air quality". We selected two of the existing attributes, "temperature" and "relative humidity", as the input data and chose "absolute humidity" as the output. All the attributes were normalized, and the dataset was split into two domains based on "relative humidity". Data with a "relative humidity" greater than 0.5 were chosen as the source domain, and the remaining data were used to form the target domain. Further, the two attributes in the source data were all perturbed by random numbers following a normal distribution $N(0.1, 0.1)$, and the two attributes in the target data were perturbed by the normal random numbers following $N(7,1)$ and $N(5,1)$, respectively. There were 3600 labeled instances in the source domain and 1200 instances in the target data; 10 were labeled.

Although a target domain may only contain a small amount of labeled data, it can still be used to train a model. However, we assert that a model trained solely on a small amount of labeled data will not perform well. And, to support this assertion, we trained a target model with various levels of labeled target data and tested its performance denoted by "QT".

Table XI: Comparison results of different transfer learning methods

Datasets	RMSE of the models						
	Baseline 1	Baseline 2	TCA	SA	GFK	TSK	Our method
2r to 2r	1.71±0.00	1.81±0.01	4.44±0.00	4.42±0.00	4.46±0.00	1.71±0.05	1.40±0.03
3r to 3r	1.10±0.00	3.34±0.02	4.05±0.00	3.51±0.00	3.41±0.00	3.07±0.00	0.67±0.00
4r to 4r	1.96±0.00	3.53±0.04	10.34±0.00	10.54±0.00	10.54±0.01	2.42±0.18	1.56±0.00
2r to 3r	4.07±0.00	4.82±30.58	1.17±0.00	1.14±0.00	1.21±0.00	1.24±0.02	1.03±0.00
3r to 2r	3.81±0.00	3.45±54.25	3.28±0.00	3.23±0.00	3.19±0.00	0.98±0.05	0.50±0.04
2r to 4r	3.45±0.00	43.29±6502.94	4.45±0.00	4.23±0.00	4.72±0.00	1.08±0.04	0.73±0.00
4r to 2r	3.11±0.00	3.45±54.25	2.11±0.00	2.22±0.00	2.47±0.00	1.03±0.01	0.85±0.01
3r to 4r	0.94±0.00	30.12±2643.94	6.10±0.01	6.36±0.00	6.09±0.00	0.79±0.01	0.72±0.01
4r to 3r	0.97±0.00	17.90±828.50	2.19±0.00	2.37±0.00	2.61±0.00	0.70±0.00	0.65±0.00

We used the IGMM to identify the data structures in the "air quality" dataset and show the results in Fig. 9.

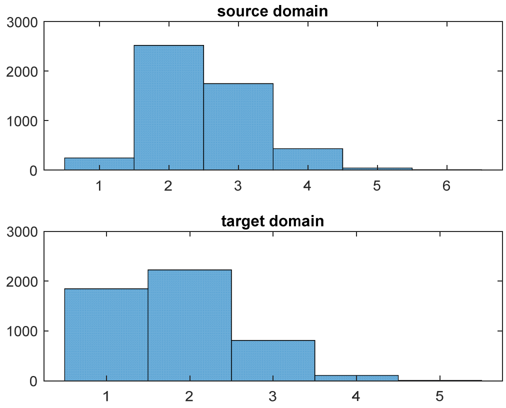


Fig. 9. IGMM's results with "air quality" dataset

From Fig. 9, we can see that the data in the source domain and target domain are both divided into two clusters with the greatest probability, so two is the optimal number of fuzzy rules to construct models and implement transfer learning. To verify this conclusion, we executed the proposed domain adaptation method with varying numbers of fuzzy rules, and compared the results, as shown in Table XII.

Table XII: Results with the "air quality" dataset

Clusters	Q	Q1	QT	Q2
2	0.1241± 0.0000	0.2575± 0.0000	0.4153± 0.0060	0.1075± 0.0000
3	0.1237± 0.0000	0.2568± 0.0000	0.3217± 0.0204	0.1097± 0.0000
4	0.1235± 0.0000	0.2618± 0.0000	0.1970± 0.0034	0.1076± 0.0000
5	0.1232± 0.0000	0.2616± 0.0000	0.2319± 0.0046	0.1085± 0.0000

In all experiments, the value for Q2 was smaller than for Q1 and QT, which indicates that the model built using our method is superior to both the existing source model and the model built using only labeled target data. Additionally, the small variances indicate that the models built using the proposed method have good generalizability. Comparing the values of Q2 with different clusters, we find that the transfer learning method has the best performance with two fuzzy rules. In addition, the number of labeled target data increased with an increase in the number of clusters due to the active learning technique. These results show that determining the appropriate number of fuzzy rules is significantly more important than accumulating more labeled data.

We conducted the same experiment on the "housing dataset", which aims to predict the "MEDV" (the median value of owner-occupied homes in US$1000's) using six input attributes. The data was normalized and split into two datasets using the attribute "TAX", which represents the full-value property tax rate per $10,000. Instances of "TAX" smaller than 0.5 were used to form the source dataset, and instances of "TAX" larger than 0.5 were used as the target dataset. The attributes "RM", "AGE", and "B" of the source data were perturbed by random numbers taken from $N(0.1, 0.1)$, while the same attributes in the target data were perturbed by normal random numbers using the distributions $N(7,1)$, $N(5,1)$ and $N(8,1)$, respectively. There were 360 labeled instances in the source domain and 130 instances in the target data; 10 were labeled.

Again, IGMM was used to identify the data structure. The results are shown in Fig. 10. Unlike the first dataset, where it was easy to determine the number of fuzzy rules, in this dataset, the probability distributions of the clusters in the source and target domains are quite different. Although the source data and target data were derived from the same domain, our modifications resulted in quite different data distributions in each domain. Based on our analysis of IGMM's results, we decided to try all the numbers of clusters as a cross-check of IGMM's performance. The results are shown in Table XIII.

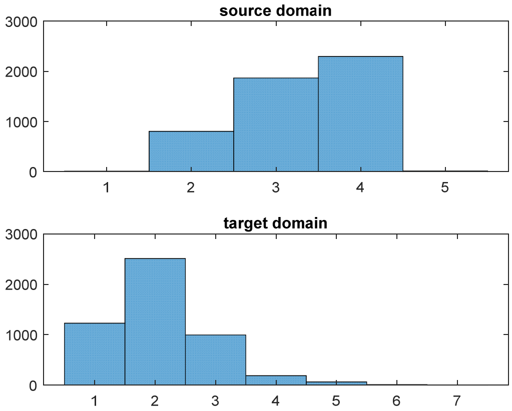


Fig. 10 IGMM's results with "housing" dataset

From Table XIII, we can see that the constructed target model performed best with three fuzzy rules. There was no obvious trend with a change in the number of fuzzy rules.

Therefore, traversing all the numbers of clusters and choosing the best one remains the best method for choosing the optimal number of fuzzy rules in cases where the source and target data greatly diverge.

Table XIII: Results with the “housing” dataset

Clusters	Q	Q1	QT	Q2
2	0.1098± 0.0003	0.2558± 0.0003	0.6306± 0.1175	0.1799± 0.0002
3	0.1006± 0.0001	0.2280± 0.0005	0.2931± 0.0179	0.1713± 0.0002
4	0.0913± 0.0002	0.1801± 0.0003	0.3670± 0.0093	0.2276± 0.0000
5	0.0902± 0.0001	0.1844± 0.0002	0.2297± 0.0040	0.1827± 0.0002
6	0.1044± 0.0017	0.2504± 0.0074	0.2053± 0.0007	0.2325± 0.0003
7	0.0920± 0.0001	0.3463± 0.0266	0.2850± 0.0088	0.1813± 0.0001

VI. Conclusion and Future Work

This work presents a method of discovering the structure of data and actively augmenting information in a target domain to improve the performance of fuzzy rule-based domain adaptation. IGMM is used to explore the relationship between the data structures in the source and target domains and provide guidance on a domain selection and transfer strategy. The idea of active learning is applied to increase the amount of labeled information in target domain by actively labeling the most informative data in the target domain. A set of experiments on synthetic datasets verifies both the positive effect of IGMM and the active learning technique on the transfer learning process. Additionally, promising results on real-world datasets validate the effectiveness of the proposed domain adaptation method in practical settings.

The method presented in this paper focuses on domain adaptation problems with homogeneous feature spaces. Future studies will explore the more challenge knowledge transfer problem of heterogeneous domain adaptation, where the feature spaces of the two domain are not identical.

References

[1] N. M. Nasrabadi, "Pattern recognition and machine learning," *Journal of Electronic Imaging,* vol. 16, no. 4, p. 049901, 2007.

[2] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks,* vol. 61, pp. 85-117, 2015.

[3] T. Hastie, R. Tibshirani, and J. Friedman, "Overview of supervised learning," in *The Elements of Statistical Learning*: Springer, 2009, pp. 9-41.

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

[5] M. Pratama, J. Lu, S. Anavatti, E. Lughofer, and C.-P. Lim, "An incremental meta-cognitive-based scaffolding fuzzy neural network," *Neurocomputing,* vol. 171, pp. 89-105, 2016.

[6] M. Pratama, J. Lu, E. Lughofer, G. Zhang, and S. Anavatti, "Scaffolding type-2 classifier for incremental learning under concept drifts," *Neurocomputing,* vol. 191, pp. 304-329, 2016.

[7] M. Xiao and Y. Guo, "Feature space independent semi-supervised domain adaptation via kernel matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 37, no. 1, pp. 54-66, 2015.

[8] S. J. Pan, V. W. Zheng, Q. Yang, and D. H. Hu, "Transfer learning for wifi-based indoor localization," in *Association for the Advancement of Artificial Intelligence (AAAI) Workshop*, 2008, p. 6.

[9] V. Behbood, J. Lu, and G. Zhang, "Long term bank failure prediction using fuzzy refinement-based transductive transfer learning," in *2011 IEEE International Conference on Fuzzy Systems (FUZZ)*, 2011, pp. 2676-2683: IEEE.

[10] H. Daumé III, A. Kumar, and A. Saha, "Frustratingly easy semi-supervised domain adaptation," in *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, 2010, pp. 53-59: Association for Computational Linguistics.

[11] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 668-675.

[12] J. Deng, Z. Zhang, F. Eyben, and B. Schuller, "Autoencoder-based unsupervised domain adaptation for speech emotion recognition," *IEEE Signal Processing Letters,* vol. 21, no. 9, pp. 1068-1072, 2014.

[13] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2066-2073: IEEE.

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

[15] J. Jiang and C. Zhai, "Instance weighting for domain adaptation in NLP," in *ACL*, 2007, vol. 7, pp. 264-271.

[16] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 36, no. 6, pp. 1134-1148, 2014.

[17] C. Wang and S. Mahadevan, "Heterogeneous domain adaptation using manifold alignment," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, 2011, vol. 22, no. 1, p. 1541.

[18] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, no. 5, pp. 770-787, 2010.

[19] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," *arXiv preprint arXiv:1412.4729,* 2014.

[20] J. R. Finkel and C. D. Manning, "Hierarchical bayesian domain adaptation," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2009, pp. 602-610: Association for Computational Linguistics.

[21] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, 2012, pp. 17-36.

[22] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1891-1898.

[23] S. Tan, K. C. Sim, and M. Gales, "Improving the interpretability of deep neural networks with stimulated learning," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 617-623: IEEE.

[24] V. Behbood, J. Lu, and G. Zhang, "Fuzzy refinement domain adaptation for long term prediction in banking ecosystem," *IEEE Transactions on Industrial Informatics,* vol. 10, no. 2, pp. 1637-1646, 2014.

[25] V. Behbood, J. Lu, G. Zhang, and W. Pedrycz, "Multistep fuzzy bridged refinement domain adaptation algorithm and its application to bank failure prediction," *IEEE Transactions on Fuzzy Systems,* vol. 23, no. 6, pp. 1917-1935, 2015.

[26] C. Yang, Z. Deng, K.-S. Choi, and S. Wang, "Takagi–Sugeno–Kang transfer learning fuzzy logic system for the adaptive recognition of epileptic electroencephalogram signals," *IEEE Transactions on Fuzzy Systems,* vol. 24, no. 5, pp. 1079-1094, 2016.

[27] Z. Deng, Y. Jiang, F.-L. Chung, H. Ishibuchi, and S. Wang, "Knowledge-leverage-based fuzzy system and its modeling," *IEEE Transactions on Fuzzy Systems,* vol. 21, no. 4, pp. 597-609, 2013.

[28] P. P. Angelov and D. P. Filev, "An approach to online identification of Takagi-Sugeno fuzzy models," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 34, no. 1, pp. 484-498, 2004.

[29] H.-J. Rong, N. Sundararajan, G.-B. Huang, and P. Saratchandran, "Sequential adaptive fuzzy inference system (SAFIS) for nonlinear

system identification and prediction," *Fuzzy Sets and Systems,* vol. 157, no. 9, pp. 1260-1275, 2006.

[30] M. Pratama, S. G. Anavatti, P. P. Angelov, and E. Lughofer, "PANFIS: A novel incremental learning machine," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 25, no. 1, pp. 55-68, 2014.

[31] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, "Fuzzy regression transfer learning in Takagi-Sugeno fuzzy models," *IEEE Transactions on Fuzzy Systems,* Vol. 25, No. 6, pp. 1795-1807, 2016.

[32] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, "Granular Fuzzy Regression Domain Adaptation in Takagi-Sugeno Fuzzy Models," *IEEE Transactions on Fuzzy Systems,* 2017.

[33] D. Weenink, "Canonical correlation analysis," in *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam*, 2003, vol. 25, pp. 81-99: University of Amsterdam.

[34] M. L. Hadjili and V. Wertz, "Takagi-Sugeno fuzzy modeling incorporating input variables selection," *IEEE Transactions on Fuzzy Systems,* vol. 10, no. 6, pp. 728-742, 2002.

[35] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Advances in Neural Information Processing Systems*, 2000, pp. 554-560.

[36] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison,* vol. 52, no. 55-66, p. 11, 2010.

[37] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proceedings of the Eleventh International Conference on Machine Learning*, 1994, pp. 148-156.

[38] R. Burbidge, J. J. Rowland, and R. D. King, "Active learning for regression based on query by committee," in *International Conference on Intelligent Data Engineering and Automated Learning*, 2007, pp. 209-218: Springer.

[39] B. Settles, M. Craven, and S. Ray, "Multiple-instance active learning," in *Advances in Neural Information Processing Systems*, 2008, pp. 1289-1296.

[40] Y. Guo and R. Greiner, "Optimistic Active-Learning Using Mutual Information," in *IJCAI*, 2007, vol. 7, pp. 823-829.

[41] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 1070-1079: Association for Computational Linguistics.

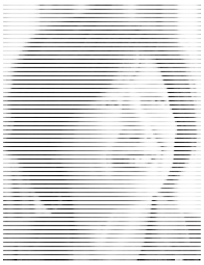
[42] M. Pratama, S. G. Anavatti, and J. Lu, "Recurrent classifier based on an incremental metacognitive-based scaffolding algorithm," *IEEE Transactions on Fuzzy Systems,* vol. 23, no. 6, pp. 2048-2066, 2015.

[43] E. Lughofer and M. Pratama, "Online Active Learning in Data Stream Regression Using Uncertainty Sampling Based on Evolving Generalized Fuzzy Models," *IEEE Transactions on Fuzzy Systems,* vol. 26, no. 1, pp. 292-309, 2018.

Hua Zuo is a postdoctoral research associate with the Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. She received the Ph.D. degree from the University of Technology Sydney, Sydney, Australia, in 2018.

Her research interests include transfer learning and fuzzy systems.

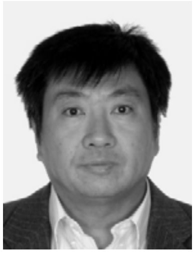
She is a Member of the Decision Systems and e-Service Intelligence (DeSI) Research Laboratory at the Centre for Artificial Intelligence, University of Technology Sydney.



Jie Lu (F'17) is a Distinguished Professor, the Director of the Centre for Artificial Intelligence, and the Associate Dean (Research Excellence) with the Faculty of Engineering and Information Technology at the University of Technology Sydney, Australia. She received her PhD in information systems from the Curtin University of Technology, Australia, in 2000.

Her research expertise spans fuzzy transfer learning, decision support systems, recommender systems, concept drift, and their applications in e-business. She has published 10 research books and over 400 papers in refereed journals and conference proceedings, with over 170 papers in IEEE Transactions and other international journals. She has been awarded eight Australian Research Council (ARC) Discovery Project grants and many other research grants. She is a member of the ARC College of Experts.

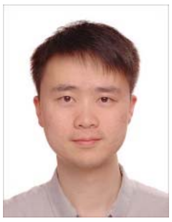
She serves as Editor-In-Chief for *Knowledge-Based Systems* (Elsevier), Editor-In-Chief for the *International Journal on Computational Intelligence Systems* (Atlantis), Associate Editor for *IEEE Transactions on Fuzzy Systems*, Editor for a book series on Intelligent Information Systems (World Scientific), and has served as a guest editor of 12 special issues, general/PC/organization chairs for ten international conferences as well as having delivered 16 keynote/plenary speeches at IEEE and other international conferences.



Guangquan Zhang is an Associate Professor and the Director of the Decision Systems and e-Service Intelligent (DeSI) Research Laboratory at the Center for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. He received his PhD in applied mathematics from Curtin University of Technology, Australia, in 2001.

His research interests include fuzzy sets and systems, fuzzy optimization, fuzzy transfer learning, and fuzzy modelling in machine learning and data analytics. He has authored four monographs, five textbooks, and 300 papers including 154 refereed international journal papers.

Dr. Zhang has won seven Australian Research Council (ARC) Discovery Projects grants and many other research grants. He was awarded an ARC QEII fellowship in 2005. He has served as a member of the editorial boards of several international journals, as a guest editor of eight special issues of IEEE Transactions and other international journals, and has co-chaired several international conferences and workshops in the area of fuzzy decision-making and knowledge engineering.



Feng Liu received the M.S. degree in probability and statistics and B.S. degree in mathematics from the School of mathematics and statistics, Lanzhou University, China, in 2015 and 2013, respectively. He is working toward the Ph.D. degree with the Faculty of Engineering and Information Technology, University of Technology Sydney, Australia.

His research interests include transfer learning and domain adaptation. He is a Member of the Decision Systems and e-Service Intelligence (DeSI) Research Laboratory, Center for Artificial Intelligence, University of Technology Sydney.